Installing GT 6.0

Installing GT 6.0

Table of Contents

List of Tables

This guide is the starting point for everyone who wants to install Globus Toolkit 6.0. It will take you through a basic installation that installs the following basic services: a security infrastructure (GSI), GridFTP, and Execution Services (GRAM5). This guide is also available as a PDF[1]. However, each component includes online reference material, which this guide sometimes links to.

[1] installingGT.pdf

Chapter 1. Before you begin

Before you start installing the Globus Toolkit 6.0, there are a few things you should consider. The toolkit contains several components, and you may only be interested in some of them.

The Globus Toolkit version 6.0 includes:

- GSI[1]: security
- GridFTP[2]: file transfer
- GRAM[3]: job execution/resource management
- MyProxy[4]: credential repository/certificate authority
- GSI-OpenSSH[5]: GSI secure single sign-on remote shell

If you are new to the toolkit and want to experiment with the components, you may want to use a supported RedHat based or Debian based Linux system. With the new supported native packaging installs, they are the simplest platforms on which to install GT services.

For the purposes of this documentation, Globus is being installed on a machine called **`elephant`**.

1. Typographical Conventions

Where there is a command to be typed, it will be preceded by one of the following prompts:

`root#` , **`root@donkey#`**

Run this command as the **`root`** super-user, on the **`elephant`** or **`donkey`** hosts respectively. You might have to use a command like **`su`**(8) or **`sudo`**(8) to start a root shell before executing the command.

`myproxy%`

Run this command as the **`myproxy`** user, on the **`elephant`** host. This user is created automatically when the **`myproxy-server`** package is installed.

`quser%` , **`quser@donkey%`**

Run this command as the normal user account you are intending to interact with your Globus sevices, on the **`elephant`** or **`donkey`** hosts. In this document, we use the **`quser`** accout for this, but if you have another user, you can use it for that purpose.

Commands themselves will be typeset as **`run-this-command -with-arguments`**, and responses to the commands like this `Some Response Text`. If there is some portion of a command which should be replaced by value, such as a version number, it will be typeset like this: *REPLACEME*.

Finally, in some cases you will be prompted for a passphrase. When that occurs, the entry of the passphrase will be indicated by `******`, even though nothing will be printed to the screen.

[1] ../../gsic/index.html
[2] ../../gridftp/index.html
[3] ../../gram5/index.html
[4] ../../myproxy/index.html
[5] ../../gsiopenssh/index.html

Chapter 2. Installing GT 6.0

1. Installing Binary Packages

1.1. Prerequisites

We distribute the Globus Toolkit 6 as a set of RPM and Debian packages for Linux systems, as an installable package for Mac OS X, as a .zip file for Windows and Cygwin, as well as a source installer which can be used on other operating systems. In this quickstart, we will be installing RPM packages. Thus, it is a prerequisite for following this quickstart that you are running a distribution for which we provide RPMs. If you are running a supported Debian or Ubuntu system, the process is very similar, but you'll need to use the **`apt-get`** or similar tools to install the packages. For the source installer, there is more work involved, and you'll need to consult the full installation guide.

First, we will to set up our system to use the Globus package repository. This repository contains the Globus software packages, signed by our build manager. We provide RPM and Debian packages that contain a source configuration file and the public key which can be used to verify the packages. If your distribution has Globus 6.0 packages within its repository, you can skip to the next section.

The globus toolkit package repo RPM can be downloaded from the repo RPM package on globus.org[1].

To install binary RPMs, download the globus-toolkit-repo package from the link above and install it with the command:

```
root# rpm -hUv globus-toolkit-repo-latest.noarch.rpm
```

The globus toolkit package repo Debian file can be downloaded from the repo Debian package on globus.org[2].

To install Debian or Ubuntu package, download the globus-toolkit-repo package from the link above and install it with the command:

```
root# dpkg -i globus-toolkit-repo_latest_all.deb
```

Once you've installed the Globus repository package, you can use your operating system's packaging tools: **`yum`** or **`apt-get`**, to install the Globus components.

Important

For operating systems based on RHEL (such as Red Hat Enterprise Linux, CentOS, and Scientific Linux), the compatible EPEL repository must be enabled before installing myproxy. For OS versions 5.x, install the EPEL 5 package[3], and for OS version 6.x, use 6 package[4]. For information about installing these, see the EPEL FAQ[5]. This step is not needed for Fedora, Debian, or Ubuntu systems.

Important

For SUSE Linux Enterprise Server systems which will be using globus-connect-server, a newer version of apache2 must be installed in order for myproxy-oauth to work. This is available by adding the Apache2 and Apache2 Modules for SLES11 repositories from opensuse.org. These can be installed by running these commands:

[1] http://www.globus.org/ftppub/gt6/installers/repo/globus-toolkit-repo-latest.noarch.rpm
[2] http://www.globus.org/ftppub/gt6/installers/repo/globus-toolkit-repo%5flatest%5fall.deb
[3] http://download.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm
[4] http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-7.noarch.rpm
[5] http://fedoraproject.org/wiki/EPEL/FAQ#How%5fcan%5fI%5finstall%5fthe%5fpackages%5ffrom%5fthe%5fEPEL%5fsoftware%5frepository.3F

```
    root# zypper ar http://download.opensuse.org/repositories/Apache/SLE_11_SP3/Apache.re
    root# zypper ar http://download.opensuse.org/repositories/Apache:/Modules/Apache_SLE_
    root# rpm --import http://download.opensuse.org/repositories/Apache/SLE_11_SP3/repoda
    root# rpm --import http://download.opensuse.org/repositories/Apache:/Modules/Apache_S
```

1.2. Installing the Toolkit on Linux

The components of the toolkit can be installed separately, or all at once. This section will show how to install various components, on both RPM based and Debian based Linux systems.

For Fedora or Red Hat-based systems, used the **yum** command to install the Globus components and their dependencies. For SUSE Linux Enterprise Server systems, use **zypper**. For Debian-based systems, used the **apt-get** or **aptitude** commands.

For example, to install the GridFTP client tools, do the following for RPM-based systems:

```
root# yum install globus-data-management-client
```

Do the following for Debian-based systems:

```
root# apt-get install globus-data-management-client
```

1.2.1. Package Groups

The Globus Toolkit distribution includes several high-level package groups that can be used to install multiple packages to enable full client or server functionality of some Globus Toolkit component.

These packages are:

globus-gridftp
GridFTP client and server tools

globus-gram5
GRAM5 client and server tools

globus-gsi
Globus Security Infrastructure tools for managing certificates and proxies

globus-data-management-server
Server tools for deploying a GridFTP server.

globus-data-management-client
Client Tools for data management, including the GridFTP client programs and globus-url-copy

globus-data-management-sdk
Development headers and documentation for writing applications using the GridFTP APIs.

globus-resource-management-server
Server tools for deploying a GRAM5 resource manager

globus-resource-management-client
Client tools for resource management, including the globusrun tool, and the globus-job-* tools.

globus-resource-management-sdk
Development headers and documentation for writing applications using the GRAM5 APIs.

1.2.2. Updating a Globus Installation

In GT 6, there are three Globus Toolkit package repositories: **Stable**, **Testing**, and **Unstable**. The **Stable** repository is enabled by default, and is updated to to include fixes for major bugs and security issues. These can be easily installed via **`yum`** or **`apt-get`**. These updates will be published in the GT[1]. Also, this means that when the next point release is made, collecting other minor bug fixes, the upgrade can be done via **`yum`** or **`apt-get`** without installing a new repository definition package.

In addition, users may enable the **Testing** or **Unstable** package repositories. These have different levels of documentation and testing done to them.

The **Testing** repository contains packages which have passed our automated test suite and are made available to people who are interested in the latest bug fixes. These packages will likely be migrated to the **Stable** repository once the package has been verified to fix a bug or issue and the documentation has been updated to include informtion about the issue.

The **Unstable** repository contains packages which have compiled successfully, but may not have completed all tests or are experimental in some way. Packages from the **Unstable** will potentially make it to the **Testing** repository once they seem to be functional.

1.3. Installing the Toolkit on Mac OS X

Download the Mac OS X Globus Toolkit Installation Package from the Globus Toolkit web site. Click on `globus_toolkit-6.0.pkg`, and follow the installation instructions. If you select the "Install for me only" option, your , and follow the installation instructions. If you select the "Install for me only" option, your `$HOME/.profile` is modified to add the Globus Toolkit components to your path. If you are using a different shell, you may need to incorporate those changes into your shell initialization file. If you install for all users, the global path will be updated. is modified to add the Globus Toolkit components to your path. If you are using a different shell, you may need to incorporate those changes into your shell initialization file. If you install for all users, the global path will be updated.

To uninstall the toolkit, run the **`globus-uninstall`** script which will remove the toolkit and revert the PATH changes.

1.4. Installing the Toolkit on Windows

There are four options when installing the Globus Toolkit on Windows: either using cygwin (32- and 64- bit builds) or MingW (32- and 64- bit builds).

The Cygwin installation requires the cygwin runtime (either 32-bit or 64-bit) to be installed: see cygwin.com[7] for details. To use the Globus Toolkit on cygwin, download the globus_toolkit-6.0-x86_64-pc-cygwin.zip or To use the Globus Toolkit on cygwin, download and unzip the globus_toolkit-6.0-i386-pc-cygwin.zip file and in the cygwin root directory. This will create files in /opt/globus

The mingw installtion does not require a special runtime, but some parts of the toolkit do not work with it: (LIST PENDING). To install the MingW packages, download the globus_toolkit-6.0-x86_64-w64-mingw32.zip or To use the Globus Toolkit on cygwin, download and unzip the globus_toolkit-6.0-i386-w64-mingw32.zip file. Add the unzipped directory’s Globus\bin and Globus\sbin paths to your PATH environment to be able to use the Globus Toolkit.

2. Installation from Source Installer

Note

Installing using the Source Installer is only recommended on platforms for which native packages are not available. If you are installing onto a RedHat or Debian based Linux system, please see the section above.

[1] http://www.globus.org/toolkit/rss/advisories/6.rss

[7] http://www.cygwin.com

Note

Make you sure you check out Platform Notes[8] for specific installation information related to your platform.

2.1. Software Prerequisites

2.1.1. Required software

To build the Globus Toolkit from the source installer, first download the source from download page[1], and be sure you have all of the following prerequisites installed.

This table shows specific package names (where available) for systems supported by GT 6.0:

Prerequisite	Reason	RedHat-based Systems	Debian-based Systems	Solaris 11	Mac OS X
C Compiler	Most of the toolkit is written in C, using C99 and POSIX.1 features and libraries.	gcc	gcc	pkg:/developer/gcc-45 or Solaris Studio[10] 12.3	XCode[11]
GNU or BSD sed	Standard sed does not support long enough lines to process autoconf-generated scripts and Makefiles	sed	sed	pkg:/text/gnu-sed	(included in OS)
GNU Make	Standard make does not support long enough lines to process autoconf-generated makefiles	make	make	pkg:/developer/build/gnu-make	(included in XCode)
OpenSSL 0.9.8 or higher	GSI security uses OpenSSL's implementation of the SSL protocol and X.509 certificates.	openssl-devel	libssl-dev	pkg:/library/security/openssl	(included in base OS)
Perl 5.10 or higher	Parts of GRAM5 are written in Perl, as are many test scripts	perl	perl	pkg:/runtime/perl-512	(included in base OS)

[8] #gtadmin-platform
[1] http://www.globus.org/toolkit/downloads/6.0
[11] https://developer.apple.com/xcode/
[10] http://www.oracle.com/technetwork/server-storage/solarisstudio/downloads/index.html

Prerequisite	Reason	RedHat-based Systems	Debian-based Systems	Solaris 11	Mac OS X
pkg-config	Parts of GRAM5 are written in Perl	pkgconfig	pkg-config	pkg:/developer/gnome/gettext	Download and install from freedesktop.org source packages[12]

Note

In order to use the GNU versions of sed, tar, and make on Solaris, put `/usr/gnu/bin` at the head of your path. Also, to use all of the perl executables, add at the head of your path. Also, to use all of the perl executables, add `/usr/perl5/bin` to your path. to your path.

2.2. Installing from Source Installer

1. Create a user named **`globus`**. This non-privileged user will be used to perform administrative tasks, deploying services, etc. Pick an installation directory, and make sure this account has read and write permissions in the installation directory.

   ### Tip

   You might need to create the target directory as **`root`**, then chown it to the **`globus`** user:

   ```
   root# mkdir
   root# chown globus:globus
   ```

   ### Important

   If for some reason you do **not** create a user named **`globus`**, be sure to run the installation as a **non-root** user. In that case, make sure to pick an install directory that your user account has write access to.

2. Download the required software noted in Software Prerequisites[13].

3. The Globus Toolkit Source Installer sets the installation directory by default to `/usr/local/globus-6`, but you may replace , but you may replace `/usr/local/globus-6` with whatever directory you wish to install to, by setting the prefix when you configure. with whatever directory you wish to install to, by setting the prefix when you configure.

   As the globus user, run:

   ```
   globus% ./configure --prefix=
   ```

   You can use command line arguments to ./configure for a more custom install.

   For a full list of options, see **`./configure --help`**.

4. The source installer will build all of the Globus Toolkit packages in the default make rule. The following Makefile targets can be used to build subsets of the Globus Toolkit:

ccommonlibs C Common Libraries

[12] http://pkgconfig.freedesktop.org/releases/

[13] #gtadmin-prereq

gridftp	GridFTP Client and Server
gsi	Security Libraries and Tools
gsi	Security Libraries and Tools
udt	Globus XIO UDT Driver
myproxy	MyProxy Client and Server
gsi-openssh	GSI OpenSSH Client and Server
gram5	GRAM5 Client and Libraries
gram5-server	GRAM5 Service
gram5-lsf	GRAM5 LSF Adapter
gram5-sge	GRAM5 SGE Adapter
gram5-slurm	GRAM5 SLURM Adapter
gram5-condor	GRAM5 Condor Adapter
gram5-pbs	GRAM5 PBS Adapter
gram5-auditing	GRAM5 Auditing Support

Run:

```
globus% make
```

Note that this command can take a while to complete. If you wish to have a log file of the build, use **`tee`**:

```
globus% make 2>&1 | tee build.log
```

The syntax above assumes a Bourne shell. If you are using another shell, redirect stderr to stdout and then pipe it to **`tee`**.

5. To test the toolkit, or particular packages within the toolkit, run:

```
globus% make check
```

or

```
globus% make COMPONENT-check
```

where *COMPONENT* is the name of the package to test. As an example, you could run

```
globus% make globus_gssapi_gsi-check
```

to run the GSSAPI test programs.

6. Finally, run:

```
globus% make install
```

This completes your installation. Now you may move on to the configuration sections of the following chapters.

We recommend that you install any security advisories available for your installation, which are available from the Advisories page[1]. You may also be interested in subscribing to some mailing lists[15] for general discussion and security-related announcements.

2.3. Updating an Installation

The updates available in the native packages described above are also published as source packages on the updates page[1]. To install update packages, follow their download link, untar them, and then configure them with the same prefix as your original installation.

[1] http://www.globus.org/toolkit/advisories.html?version=6

[15] http://dev.globus.org/wiki/Mailing%5fLists

[1] http://www.globus.org/toolkit/advisories.html?version=6

Chapter 3. Basic Security Configuration

1. Obtain host credentials

You must have X.509 certificates to use the GT 6.0 software securely (referred to in this documentation as **host certificates**). For an overview of certificates for GSI (security) see GSI Configuration Information[1] and GSI Environment Variables[2].

If you will need to be interoperable with other sites, you will need to obtain certs from a trusted Certificate Authority, such as those that are included in IGTF[3]. If you are simply testing the software on your own resources, SimpleCA offers an easy way to create your own certificates (see section below).

Host credentials must:

- consist of the following two files: `hostcert.pem` and and `hostkey.pem`
- be in the appropriate directory for secure services: `/etc/grid-security/`
- match the hostname for a the machine. If the machine is going to be accessed remotely, the name on the certificate must match the network-visible hostname.

You have the following options:

1.1. Request a certificate from an existing CA

Your best option is to use an already existing CA. You may have access to one from the company you work for or an organization you are affiliated with. Some universities provide certificates for their members and affiliates. Contact your support organization for details about how to acquire a certificate. You may find your CA listed in the TERENA Repository[4].

If you already have a CA, you will need to follow their configuration directions. If they include a CA setup package, follow the CAs instruction on how to install the setup package. If they do not, you will need to create an `/etc/grid-security/certificates` directory and include the CA cert and signing policy in that directory. See directory and include the CA cert and signing policy in that directory. See Configuring a Trusted CA[5] for more details.

This type of certificate is best for service deployment and Grid inter-operation.

1.2. SimpleCA

SimpleCA provides a wrapper around the OpenSSL CA functionality and is sufficient for simple Grid services. Alternatively, you can use OpenSSL's **`CA.sh`** command on its own. Instructions on how to use the SimpleCA can be found in Installing SimpleCA[6].

SimpleCA is suitable for testing or when a certificate authority is not available.

[1] ../../gsic/admin/index.html#gsic-configuring
[2] ../../gsic/developer/index.html#gsic-developer-env
[3] http://www.igtf.net
[4] http://www.tacar.org/
[5] ../../gsic/admin/index.html#gsic-configuring
[6] ../../admin/install/appendix.html#gtadmin-simpleca

If you install the **globus-simpleca** native package, it will automatically create a CA and host certificate if you don’t have one configured yet. Otherwise, you’ll need to use **grid-ca-create** to create the CA and **grid-default-ca** to make that the default for requesting credentials.

To create user credentials, you can run the command **grid-cert-request** as a user that you want to create a credential for. You can then run the **grid-ca-sign** command as the **simpleca** user to sign the certificate.

2. Add authorization

Installing Globus services on your resources doesn’t automatically authorize users to use these services. Each user must have their own user certificate, and each user certificate must be mapped to a local account.

To add authorizations for users, you’ll need to update the `grid-mapfile` database to include the mapping between the credentials and the local user accounts. database to include the mapping between the credentials and the local user accounts.

You’ll need two pieces of information:

- the subject name of a user’s certificate
- the local account name that the certificate holder can access.

To start with, if you have created a user certificate, you can run the **grid-cert-info** command to get the certificate’s subject name, and **id -un** to get the account name:

```
globus% grid-cert-info -subject
/O=Grid/OU=GlobusTest/OU=simpleCA-elephant.globus.org/CN=Globus User
globus% id -un
globus
```

You may add the line by running the following command as root:

```
root# grid-mapfile-add-entry \
    -dn "/O=Grid/OU=GlobusTest/OU=simpleCA-elephant.globus.org/CN=Globus User" \
    -ln gtuser
Modifying /etc/grid-security/grid-mapfile ...
/etc/grid-security/grid-mapfile does not exist... Attempting to create /etc/grid-security/
New entry:
"/O=Grid/OU=GlobusTest/OU=simpleCA-elephant.globus.org/CN=Globus User" globus
(1) entry added
```

Important

The quotes around the subject name are **important**, because it contains spaces.

3. Verify Basic Security

Now that you have installed a trusted CA, acquired a hostcert and acquired a usercert, you may verify that your security setup is complete. As your user account, run the following command:

```
gtuser$ grid-proxy-init -verify -debug

User Cert File: /home/gtuser/.globus/usercert.pem
User Key File: /home/gtuser/.globus/userkey.pem
```

```
Trusted CA Cert Dir: /etc/grid-security/certificates

Output File: /tmp/x509up_u506
Your identity: /DC=org/DC=doegrids/OU=People/CN=GT User 332900
Enter GRID pass phrase for this identity:
Creating proxy ...++++++++++++
..................++++++++++++
 Done
Proxy Verify OK
Your proxy is valid until: Fri Jan 28 23:13:22 2005
```

There are a few things you can notice from this command. Your usercert and key are located in `$HOME/.globus/`. The proxy certificate is created in . The proxy certificate is created in `/tmp/`. The "up" stands for "user proxy", and the . The "up" stands for "user proxy", and the *_u506* will be your UNIX userid. It also prints out your distinguished name (DN), and the proxy is valid for 12 hours.

If this command succeeds, your single node is correctly configured.

If you get an error, or if you want to see more diagnostic information about your certificates, run the following:

```
gtuser$ grid-cert-diagnostics
```

For more troubleshooting information, see the GSI troubleshooting guide[7]

4. Firewall configuration

There are four possible firewall scenarios that might present themselves: restrictions on incoming and outgoing ports for both client and server scenarios.

This section divides sites into two categories: client sites, which have users that are acting as clients to Grid services, and server sites, which are running Grid services. Server sites also often act as client sites either because they also have users on site or jobs submitted by users to the site act as clients to other sites by retrieving data from other sites or spawning sub-jobs.

4.1. Client Site Firewall Requirements

This section describes the requirements placed on firewalls at sites containing Globus Toolkit clients. Note that often jobs submitted to sites running Globus services will act as clients (e.g. retrieving files needed by the job, spawning subjobs), so server sites will also have client site requirements.

4.1.1. Allowed Outgoing Ports

Clients need to be able to make outgoing connections freely from ephemeral ports on hosts at the client site to all ports at server sites.

4.1.2. Allowed Incoming Ports

As described in Job State Callbacks and Polling[8], the Globus Toolkit GRAM service uses callbacks to communicate state changes to clients and, optionally, to stage files to/from the client. If connections are not allowed back to the Globus Toolkit clients, the following restrictions will be in effect:

[7] ../../gsic/admin/index.html#gsic-admin-troubleshooting

[8] ../../gram5/developer/index.html#gram5-developer-jobstatecallbacks

- You cannot do a job submission request and redirect the output back to the client. This means the globus-job-run command won't work. globus-job-submit will work, but you cannot use globus-job-get-output. globusrun with the -o option also will not work.
- Staging to or from the client will also not work, which precludes the -s and -w options.
- The client cannot be notified of state changes in the job, e.g. completion.

To allow these callbacks, client sites should allow incoming connection in the ephemeral port range. Client sites wishing to restrict incoming connections in the ephemeral port range should select a port range for their site. The size of this range should be approximately 10 ports per expected simultaneous user on a given host, though this may vary depending on the actual usage characteristics. Hosts on which clients run should have the GLOBUS_TCP_PORT_RANGE environment variable set for the users to reflect the site's chosen range.

4.1.3. Network Address Translation (NAT)

Clients behind NATs will be restricted as described in Allowed Incoming Ports[9] unless the firewall and site hosts are configured to allow incoming connections.

This configuration involves:

- Select a separate portion of the ephemeral port range for each host at the site on which clients will be running (e.g. 45000-45099 for host A, 45100-45199 for host B, etc.).
- Configure the NAT to direct incoming connections in the port range for each host back to the appropriate host (e.g., configure 45000-45099 on the NAT to forward to 45000-45099 on host A).
- Configure the Globus Toolkit clients on each site host to use the selected port range for the host using the techniques described in If client is behind a firewall[10].
- Configure Globus Toolkit clients to advertise the firewall as the hostname to use for callbacks from the server host. This is done using the GLOBUS_HOSTNAME environment variable. The client must also have the GLOBUS_HOSTNAME environment variable set to the hostname of the external side of the NAT firewall. This will cause the client software to advertise the firewall's hostname as the hostname to be used for callbacks causing connections from the server intended for it to go to the firewall (which redirects them to the client).

4.2. Server Site Firewall Requirements

This section describes firewall policy requirements at sites that host Grid services. Sites that host Grid services often host Grid clients, however the policy requirements described in this section are adequate for clients as well.

4.2.1. Allowed Incoming Ports

A server site should allow incoming connections to the well-known Grid Service Ports as well as ephemeral ports. These ports are 22/tcp (for gsi-enabled openssh), 2119/tcp (for GRAM) and 2811/tcp for GridFTP.

A server not allowing incoming connections in the ephemeral port range will have the following restrictions:

- If port 2119/tcp is open, GRAM will allow jobs to be submitted, but further management of the jobs will not be possible.
- While it will be possible to make GridFTP control connections if port 2811/tcp is open, it will not possible to actually get or put files.

[9] #gtadmin-client-incomingports
[10] ../../gridftp/user/index.html#gridftp-user-config-client-firewall

Server sites wishing to restrict incoming connections in the ephemeral port range should select a range of port numbers. The size of this range should be approximately 20 ports per expected simultaneous user on a given host, though this may vary depending on the actual usage characteristics. While it will take some operational experience to determine just how big this range needs to be, it is suggested that any major server site open a port range of at least a few hundred ports. Grid Services should configured as described in Section to reflect the site's chosen range.

4.2.2. Allowed Outgoing Ports

Server sites should allow outgoing connections freely from ephemeral ports at the server site to ephemeral ports at client sites as well as to Grid Service Ports at other sites.

4.2.3. Network Address Translation (NAT)

Grid services are not supported to work behind NAT firewalls because the security mechanisms employed by Globus require knowledge of the actual IP address of the host that is being connected to.

We do note there have been some successes in running GT services behind NAT firewalls.

4.3. Summary of Globus Toolkit Traffic

Table 3.1. Summary of Globus Toolkit Traffic

Application	Network Ports	Comments
GRAM Gatekeeper(to start jobs)	To 2119/tcp on server from controllable ephemeral port on client	Connections back to client (controllable ephemeral port to controllable ephemeral port) required if executable or data staged from client or output from job sent back to client. Port 2119/tcp defined by IANA
GRAM Job-Manager	From controllable ephemeral port on client to controllable ephemeral port on server.	Port on server selected when original connection made by the client to the Gatekeeper and returned to the client in a URL. May result in connection back to client from ephemeral port on server to controllable ephemeral port on client.
GridFTP	From controllable ephemeral port on client to port 2811/tcp on server for control channel.	Port 2811/tcp defined by IANA.
GSI-Enabled SSH	From ephemeral port on client to port 22/tcp on server.	Same as standard SSH. Port 22/tcp defined by IANA.
MyProxy	From ephemeral port on client to port 7512/tcp on server.	Default. Can be modified by site.

4.4. Controlling The Ephemeral Port Range

Controllable ephemeral ports in the Globus Toolkit can be restricted to a given range. setting the environment variable GLOBUS_TCP_PORT_RANGE can restrict ephemeral ports. The value of this variable should be formatted as min,max (a comma separated pair). This will cause the GT libraries (specifically GlobusIO) to select port numbers for controllable ports in that specified range.

```
% GLOBUS_TCP_PORT_RANGE=40000,40010
% export GLOBUS_TCP_PORT_RANGE
% globus-gass-server
https://globicus.lbl.gov:40000
^C
%
```

This environment variable is respected by both clients and servers that are started from within the environment in which it is set. There are better ways, however, to configure a globus-job-manager or a GridFTP server to restrict its port range.

- globus-job-manager has an option, -globus-tcp-port-range PORT_RANGE that acts in the same manner as the environment variable. It can be specified on the command line or in the configuration file. See the job manager documentation[11] for all of its options.
- See the GridFTP documentation[12] for information about using GridFTP with firewalls.

[11] ../../gram5/admin/index.html#gram5-cmd-globus-job-manager

[12] ../../gridftp/admin/index.html#gridftp-config-security-firewalls

Chapter 4. Basic Setup for GT 6.0

The Quickstart Guide[1] walks you through setting up basic services on multiple machines.

[1] ../../admin/quickstart/index.html

Chapter 5. Platform Notes

1. Platform Notes

1.1. Mac OS X 10.8+ (Mountain Lion, Mavericks, Yosemite)

The GNU autotools and libtool is no longer distributed with OS X 10.8+. If you are building from git repository, you'll need to install the latest versions of those tools. If you are building from the source installer, these do not need to be installed.

- GNU Autoconf[1]
- GNU Automake[2]
- GNU Libtool[3]

Configure libtool with the configuration option **`--program-prefix=g`** to cause the libtool script to be named **`glibtool`** to avoid conflicts with the OS X libtool program which provides different functionality than GNU libtool. Install libtool (and the other tools) into the a common directory. If you do so, you'll need to set the `LIBTOOLIZE` environment variable to the path to the **`glibtoolize`** program. You'll need to include the autotools in your path to regenerate the configurable scripts and Makefile.in files for the toolkit.

The Globus Toolki build requires the **`pkg-config`** package to be installed. It is available from freedesktop.org[4]. Additionally, you'll need to set the environment variable `PKG_CONFIG_PATH` to `/usr/lib/pkgconfig` prior to running the configure script. prior to running the configure script.

[1] http://ftpmirror.gnu.org/autoconf/
[2] http://ftpmirror.gnu.org/automake/
[3] http://ftpmirror.gnu.org/libtool/
[4] http://pkgconfig.freedesktop.org/releases/

Chapter 6. Appendix

The Install Guide appendix can be found here.[1]

[1] ../../admin/install/appendix.html