GT 5.2.2 SimpleCA: Admin Guide

GT 5.2.2 SimpleCA: Admin Guide

Introduction

This guide contains advanced configuration information for system administrators working with SimpleCA. It provides references to information on procedures typically performed by system administrators, including installation, configuring, deploying, and testing the installation.

Important

This information is in addition to the basic Globus Toolkit prerequisite, overview, installation, security configuration instructions in the Installing GT 5.2.2. Read through this guide before continuing!

The following are instructions for how to use SimpleCA to request and sign a *certificate* for a GT 5.2.2 installation.

SimpleCA provides an easy way to create and package a *Certificate Authority (CA)*, as well as tools for requesting and signing user and host certificates. It is similar to OpenSSL's **CA.sh** command but includes support for packaging the CA certificate, creating a signing policy file, and generating information needed by clients to request certificates. You can find other CA options in Obtaining host certificates.

Table of Contents

List of Tables

List of Examples

Chapter 1. Building and Installing

1. Create users

Make sure you have the following users on your machine:

- Your *user* account, which will be used to run the client programs.
- A generic *globus* account, which will be used to perform administrative tasks. This user will also be in charge of managing the SimpleCA.

2. Install SimpleCA

SimpleCA can be installed in three ways, from a debian package, from an RPM package, and from the source installer. These installation methods are described in Installing GT 5.2.2

To install SimpleCA from binary packages, install the packages globus-simple-ca and globus-gsi-cert-utils-progs and their dependencies. On Debian based systems, use the command **apt-get install globus-simple-ca globus-gsi-cert-utils-progs**. On RPM-based systems, use the command **yum install globus-simple-ca globus-gsi-cert-utils-progs**.

To install SimpleCA from the source installer, build the `globus_simple_ca` and `globus_gsi_cert_utils` installer targets with the command **make globus_simple_ca globus_gsi_cert_utils**.

Chapter 2. Creating a SimpleCA

To create a CA and certificate, as the *globus* user, run the **grid-ca-create** command. This will prompt for information needed to name the certificate, how to contact the CA administrator, lifetime of the CA certificate, and passphrase, and will then generate the new CA certificate and private key. Command-line options described in grid-ca-create can be used to avoid some of these prompts.

1. Invoking grid-ca-create

As the *globus* user, when you run the command, you'll see output like this:

```
    C e r t i f i c a t e    A u t h o r i t y    S e t u p

This script will setup a Certificate Authority for signing Globus
users certificates.  It will also generate a simple CA package
that can be distributed to the users of the CA.

The CA information about the certificates it distributes will
be kept in:

/home/globus/.globus/simpleCA
```

This intro screen shows the path that the CA will be created into (in this example, `/home/globus/.globus/simpleCA`). The other commands needed by SimpleCA will automatically look in that path by default when invoked by the *globus* user.

2. Configure the subject name

The **grid-ca-create** program next prompts you for information about the name of CA you wish to create:

```
The unique subject name for this CA is:
cn=Globus Simple CA, ou=simpleCA-grid.example.org, ou=GlobusTest, o=Grid

Do you want to keep this as the CA subject (y/n) [y]:
```

To accept the default name, enter **y**. To choose a different name, type **n**, after which you will be prompted by

```
Enter a unique subject name for this CA:
```

The subject name is an X.509 distinguished name. The name component type abbreviations have the following meanings:

Table 2.1. CA Name components

cn	Represents "common name". It identifies this particular certificate as the *CA Certificate* within the "GlobusTest/simpleCA-grid.example.org" domain, which in this case is Globus Simple CA.
ou	Represents "organizational unit". It identifies this CA from other CAs created by SimpleCA by oth-

	er people. The second "ou" is specific to your hostname (in this case GlobusTest).
o	Represents "organization". It identifies the Grid.

3. Configure the CA's email

The next prompt looks like this:

```
Enter the email of the CA (this is the email where certificate
requests will be sent to be signed by the CA) [globus@grid.example.org]:
```

Enter the email address where you intend to receive certificate requests. It should be your real email address that you check, not the address of the globus user. When users request certificates with **grid-cert-request**, they will be instructed to send the request to this address.

4. Configure the expiration date

Then you'll see:

```
The CA certificate has an expiration date. Keep in mind that
once the CA certificate has expired, all the certificates
signed by that CA become invalid.  A CA should regenerate
the CA certificate and start re-issuing ca-setup packages
before the actual CA certificate expires.  This can be done
by re-running this setup script.  Enter the number of DAYS
the CA certificate should last before it expires.
[default: 5 years 1825 days]:
```

This is the number of days for which the CA certificate is valid. Once this time expires, the CA certificate will have to be recreated.

To accept the default, hit **enter**, or otherwise, enter a value in days.

5. Create a Passphrase to Encrypt the CA's Private Key

The next prompt will be for the passphrase for the CA's private key. It will be used to decrypt the CA's private key when signing certificates. It should be hard to guess, as its compromise might compromise all the certificates signed by the CA. You will be prompted twice for the passphrase, to verify that you typed it correctly. Enter the passphrase at these prompts.

```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

6. SimpleCA Distribution Files

Finally **grid-ca-create** will create a tarball containing the public information about the CA, including its public certificate, signing policy, and supported X.509v3 extensions. This information is needed on machines that will be trusting the CA, and also on machines which will be used to request certificates from this CA.

Since we didn't run in this example as root, **grid-ca-create** will not be able to write the CA files to system paths, so it displays a warning message indicating that. We can use the tarball output here, or packages described below to install the CA support files on this and other machines.

The package output summary looks like this:

```
Insufficient permissions to install CA into the trusted certifiicate
directory (tried ${sysconfdir}/grid-security/certificates and
${datadir}/certificates)
Creating RPM source tarball... done
 globus_simple_ca_68ea3306
```

This information will be important for setting up other machines in your grid. The number *68ea3306* in the last line is known as your *CA hash*. It is an 8 digit hexadecimal string which is a hash of the subject name of the CA certificate.

The tarball contains Debian and RPM package metadata, so that it can be compiled to a binary package which can be easily installed on this and other systems on your Grid. It can also be packaged as a GPT setup package for compatibility with older versions of the Globus Toolkit.

7. Generating Binary CA Packages

The grid-ca-package command can be used to generate RPM, debian, or legacy GPT packages for a SimpleCA, or for any other CA which is installed on a host. These packages can make it easy to distribute the CA certificate and policy to other hosts with which you want to establish Grid trust relationships.

7.1. Generating RPM Packages

To generate an RPM package for the CA which we created, use the following command:

```
globus% grid-ca-package -r -cadir ~/.globus/simpleCA
Creating RPM source tarball... done
        globus_simple_ca_68ea3306.tar.gz
Creating RPM binary.../home/globus/globus-simple-ca-68ea3306
```

The resulting rpm package will be placed in the current directory. As root, you can install this via the **yum** or **rpm** tools. This package can then be installed on any RPM-based system.

7.2. Generating Debian Packages

To generate an Debian package for the CA which we created, use the following command:

```
globus% grid-ca-package -d -cadir ~/.globus/simpleCA
    Creating RPM source tarball... done
        globus_simple_ca_68ea3306.tar.gz
Creating debian binary...dpkg-buildpackage: export CFLAGS from dpkg-buildflags (origin: ve

...
Lots of dpkg-buildpackage output
```

The resulting debian package will be placed in the current directory. As root, you can install this via the **dpkg** tool.

7.3. Generating GPT Packages

The **grid-ca-package** command can also generate GPT packages in the form similar to previous versions of the Globus Toolkit. This is done with the `-g` and `-b` command-line options. See grid-ca-package for more details.

Chapter 3. Using a SimpleCA

As a CA, your main task will be to sign certificates. To sign a certificate request, use the tool **grid-ca-sign**. This tool reads a certificate request (such as those created by grid-cert-request) and creates a certificate signed by the CA certificate with the public key from in the certificate request. This indicates that you confirm that the identity of the certificate matches its name. You can use the **openssl** command to view the contents of the certificate request.

1. Examining a Certificate Request

To examine a certificate request, use the command **openssl req -text -in *REQNAME***, as shown in the following example.

Example 3.1. Examine a Certificate Request

```
globus% openssl req -noout -text -in certreq.pem
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: o=Grid, OU=GlobusTest, OU=simpleCA-grid.example.org, OU=local, CN=Joe Use
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    79:bd:a7:29:16:77:4c:e9:82:d3:73:a0:25:34:c7:
                    25:07:67:b3:2d:11:c1:e2:c9:b1:ec:41:20:a7:9a:
                    b7:2f:ee:d4:88:78:14:ff:d4:f2:f9:1b:d3:56:bc:
                    37:6f:f0:06:ea:b0:6f:70:12:a8:34:ac:8e:be:98:
                    00:b9:b8:ec:39:b5:6b:23:ad:1b:00:62:4b:cc:79:
                    97:cc:56:fb:54:7b:03:6d:a7:76:27:4e:ce:bd:94:
                    d0:eb:59:6b:25:c5:30:b0:47:15:bc:11:d5:7e:ff:
                    04:13:70:de:3b:8f:80:65:ae:63:82:61:38:f9:c6:
                    03:4a:92:b0:de:6f:bb:0a:bd
                Exponent: 65537 (0x10001)
        Attributes:
        Requested Extensions:
            Netscape Cert Type:
                SSL CA, S/MIME CA, Object Signing CA
    Signature Algorithm: sha1WithRSAEncryption
        85:70:a6:5d:de:be:61:45:83:48:43:8d:4b:4b:4a:79:79:98:
        0d:6c:d4:a9:96:26:41:a4:c2:94:10:92:ad:eb:ad:c5:3c:bf:
        d6:4e:84:0a:db:46:96:a9:52:5b:90:cc:6a:d1:57:73:27:98:
        9e:e2:8c:9a:7f:b4:ab:a8:28:2b:02:98:a2:d8:69:73:5e:12:
        ad:5b:de:0c:6e:60:e0:0f:2c:ad:8d:b9:59:3b:d3:49:19:52:
        e0:e1:8a:57:f2:c3:a6:4d:b9:2c:5c:58:ef:0e:59:84:55:8e:
        16:fc:f4:39:82:13:6f:28:a9:59:e3:c8:f1:4e:87:75:33:4f:
        ae:be
```

In this case, you see a certificate request with the subject distinguished name `o=Grid, OU=GlobusTest, OU=simpleCA-grid.example.org, OU=local, CN=Joe User`.

2. Signing a Certificate Request

If you are satisfied with the certificate request and are willing to sign it, use the **grid-ca-sign** command to do so. The command will store a copy of the newly signed certificate in the SimpleCA directory, so that you can keep track of what you've signed, and will also write it to the value of the `-out` parameter. Transmit this result file back to the user which requested the certificate.

Example 3.2. Sign with grid-ca-sign

```
globus% grid-ca-sign -in certreq.pem -out cert.pem


To sign the request
please enter the password for the CA key:

The new signed certificate is at: /home/globus/.globus/simpleCA/newcerts/01.pem
```

3. Revoking a Certificate

SimpleCA does not yet provide a convenient interface to revoke a signed certificate, but it can be done with the **openssl** command.

Example 3.3. Revoke a certificate

```
globus% openssl ca -config ~/.globus/simpleCA/grid-ca-ssl.conf -revoke ~/.globus/simpleCA/
Using configuration from /home/globus/.globus/simpleCA/grid-ca-ssl.conf
Enter pass phrase for /home/globus/.globus/simpleCA/private/cakey.pem:
Revoking Certificate 01.
Data Base Updated
```

Once a certificate is revoked, you can generate a Certificate Revocation List (CRL) for your CA, which will be a signed list of certificates which have been revoked. Sites which use your CA will need to keep the CRL up to date to be able to reject revoked certificates. This CRL can be generated with an **openssl** command. See ca(1) for details about how to control the CRL lifetime and other options.

Example 3.4. Create CRL

```
globus% openssl ca -config ~/.globus/simpleCA/grid-ca-ssl.conf -gencrl > CAHASH.crl
Using configuration from /home/globus/.globus/simpleCA/grid-ca-ssl.conf
Enter pass phrase for /home/globus/.globus/simpleCA/private/cakey.pem:
```

The output file *CAHASH*`.crl` (based on the hash of your CA subject name) should be distributed to sites which trust your CA, so that they can install it into the trusted certificate directory.

4. Renewing a CA

The **openssl** command can be used to renew a CA certificate. This will generate a new CA certificate with the same subject name and private key as before, but valid for a different time interval. This new certificate packaged and distributed as before using grid-ca-package.

Example 3.5. Renew CA Certificate

```
globus% openssl req -key ~/.globus/simpleCA/private/cakey.pem -new -x509 -days 1825 -out n
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Level 0 Organization [Grid]:
Level 0 Organizational Unit [GlobusTest]:
Level 1 Organizational Unit [simpleCA-grid.example.org]:
Name (E.g., John M. Smith) []:Globus Simple CA
```

Important

The Subject Name of the new certificate must match *exactly* the previous certificate name, or clients will not recognize it as the correct certificate.

Simple CA Commands

Name

grid-ca-create — Create a CA to sign certificates for use on a grid

Synopsis

`grid-ca-create` [-help] [-h] [-usage] [-version] [-versions]

`grid-ca-create` [-force] [-noint] [-dir *`DIRECTORY`*]
[-subject *`SUBJECT`*] [-email *`ADDRESS`*] [-days *`DAYS`*] [-pass *`PASSWORD`*]
[-nobuild] [-g] [-b]
[-openssl-help] [*`OPENSSL-OPTIONS`*]

Description

The **grid-ca-create** program creates a self-signed CA certificate and related files needed to use the CA with other Globus tools. The **grid-ca-create** program prompts for information to use to generate the CA certificate, but the prompts may be avoided by using the command line options.

By default, the **grid-ca-create** program creates the self-signed CA certificate, installs it on the current machine in its trusted certificate directory, and creates a source tarball which can be used to generate an RPM package for the CA. If the RPM package is installed on a machine, users on that machine can create certificate requests for user, host, or service identity certificates to be signed by the CA certificate generated by running **grid-ca-create**.

If run as a privileged user, the **grid-ca-create** program creates the CA certificate and support files in `${local-statedir}/lib/globus/simple_ca` and the CA certificate and signing policy are installed in the `/etc/grid-security` directory. Otherwise, the files are created in the `${HOME}/.globus/simpleCA` directory.

The full set of command-line options to **grid-ca-create** follows. In addition to these, unknown options will be passed to the **openssl** command when creating the self-signed certificate.

`-help`, `-h`, `-usage` — Display the command-line options to **grid-ca-create** and exit.

`-version`, `-versions` — Display the version number of the **grid-ca-create** command. The second form includes more details.

`-force` — Overwite existing CA in the destination directory if one exists

`-noint` — Run in non-interactive mode. This will choose defaults for parameters or those specified on the command line without prompting. This option also implies `-force`.

`-dir` *`DIRECTORY`* — Create the CA in *`DIRECTORY`*. The *`DIRECTORY`* must not exist prior to running **grid-ca-create**.

`-subject` *`SUBJECT`* — Use *`SUBJECT`* as the subject name of the self-signed CA to create. If this is not specified on the command-line, **grid-ca-create** will default to using the subject name *cn=Globus Simple CA, ou=`$HOSTNAME`, ou=GlobusTest, o=Grid*.

`-email` *`ADDRESS`* — Use *`ADDRESS`* as the email address of the CA. The default instructions generated by **grid-ca-create** tell users to mail the certificate request to this address. If this is not specified on the command-line, **grid-ca-create** will default to the `$LOGNAME@$HOSTNAME`

`-days` *`DAYS`* — Set the default lifetime of the self-signed CA certificate to *`DAYS`*. If not set, the **grid-ca-create** program will default to `1825` days (5 years).

`-pass` *PASSWORD* Use the string *PASSWORD* to protect the CA's private key. This is useful for automating Simple CA, but may make it easier to compromise the CA if someone obtains a shell on the machine storing the CA's private key.

`-nobuild` Disable building a source tarball for distributing the CA's public information to other machines. The source tarball can be created later by using the **grid-ca-package** command.

`-g` Create a binary GPT package containing the new CA's public information. The package will be created in the current working directory. This package can be deployed by with the **gpt-install** tool.

`-b` Create a binary GPT package containing the new CA's public information that is backward-compatible with GPT 3.2. Packages created in this manner will work with Globus Toolkit 2.0.0-5.0.x.

Examples

Create a simple CA in `$HOME/SimpleCA`

```
% grid-ca-create -noint -dir $HOME/SimpleCA

    C e r t i f i c a t e   A u t h o r i t y   S e t u p

    This script will setup a Certificate Authority for signing Globus
    users certificates.  It will also generate a simple CA package
    that can be distributed to the users of the CA.

    The CA information about the certificates it distributes will
    be kept in:

    /home/juser/SimpleCA

    The unique subject name for this CA is:

    cn=Globus Simple CA, ou=simpleCA-grid.example.org, ou=GlobusTest, o=Grid

    Insufficient permissions to install CA into the trusted certifiicate
    directory (tried ${sysconfdir}/grid-security/certificates and
    ${datadir}/certificates)
    Creating RPM source tarball... done
      globus_simple_ca_0146c503.tar.gz
```

Environment Variables

The following environment variables affect the execution of **grid-ca-create**:

`GLOBUS_LOCATION` Non-standard installation path of the Globus toolkit.

See Also

grid-cert-request(1), grid-ca-sign(1), grid-default-ca(1), grid-ca-package(1)

Name

grid-ca-package — Prepare a CA certificate, configuration, and policy for distribution

Synopsis

`grid-ca-package` [-help] [-h] [-usage] [-version] [-versions]

`grid-ca-package` [[-ca *HASH*] | [-cadir *SIMPLECADIR*]] [-g] [-b] [-r] [-d]

Description

The **grid-ca-package** utility creates a tarball containing an RPM spec file and the files needed to use a CA with grid tools. It optionally will also create a GPT package for distributing a CA.

By default, the **grid-ca-package** utility displays a list of installed grid CA and prompts for which CA to package. It then creates a tarball containing the CA certificate, signing policy, CA configuration files, and an spec script to generate a binary RPM package containing the CA. If the CA hash is known prior to running **grid-ca-package**, it may provided as an argument to the `-ca` parameter to avoid prompting. **grid-ca-package** may also be used to package a SimpleCA directory, using the `-cadir` parameter.

In addition to generating a spec script and tarball, **grid-ca-package** creates a GPT package if either the `-g` or `-b` options are used on the command-line. These packages may be used to distribute a CA and configuration to systems which do not support RPM packages.

The **grid-ca-package** utility writes the package tarballs to the current working directory.

The full set of command-line options to **grid-ca-package** follows.

`-help`, `-h`, `-usage`	Display the command-line options to **grid-ca-package** and exit.
`-version`, `-versions`	Display the version number of the **grid-ca-package** command. The second form includes more details.
`-ca` *CA*	Use the CA whose name matches the hash string *CA*. When invoked with this option, **grid-ca-package** runs non-interactively.
`-cadir` *SIMPLECADIR*	Use the SimpleCA located in *SIMPLECADIR* When invoked with this option, **grid-ca-package** runs non-interactively.
`-g`	Create a GPT binary package in addition to the RPM script tarball. This package may be installed on other systems using the **gpt-install** program.
`-b`	Create a GPT binary package with GPT metadata located in the path expected by GPT 3.2 (used in Globus 2.0.0-5.0.x) instead of `${datadir}/globus/packages` as used in Globus 5.2.x. This option overrides the `-g` command-line option.
`-r`	Create a binary RPM package for the CA. This option currently only works on RPM-based distributions.
`-d`	Create a binary Debian package for the CA. This option currently only works on Debian-based distributions.

Examples

Package a Simple CA with hash `0146c503`

```
% grid-ca-package -ca 0146c503
Creating RPM source tarball... done
  globus_simple_ca_0146c503.tar.gz
```

Environment Variables

The following environment variables affect the execution of **grid-ca-package**:

`GLOBUS_LOCATION` Non-standard installation path of the Globus toolkit.

See Also

grid-cert-request(1), grid-ca-sign(1), grid-default-ca(1), grid-ca-create(1)

Name

grid-ca-sign — Sign a certificate with a SimpleCA for use on a grid

Synopsis

`grid-ca-sign` [-help] [-h] [-usage] [-version] [-versions]

`grid-ca-sign` -in *`REQUEST`* -out *`CERTIFICATE`*
[-force] [-dir *`DIRECTORY`*]
[-openssl-help] [*`OPENSSL-OPTIONS`*]

Description

The **grid-ca-sign** program signs a certificate based on a request file with a CA certificate created by **grid-ca-create**. The new certificate is written to a file. If the CA has already signed a certificate with the same subject name as contained in the certificate request, it will refuse to sign the new request unless the `-force` option is provided on the command-line.

If run as a privileged user, **grid-ca-sign** uses the CA certificate and configuration located in `${localstate-dir}/lib/globus/simple_ca` to sign the certificate. For a non-privileged user, **grid-ca-sign** uses the CA certificate and configuration located in `$HOME/.globus/simpleCA`. The **grid-ca-sign** program an use a different CA configuration and certificate by using the `-dir` option.

The full set of command-line options to **grid-ca-sign** follows. In addition to these, unknown options will be passed to the **openssl** command when creating the self-signed certificate.

`-help`, `-h`, `-usage`	Display the command-line options to **grid-ca-sign** and exit.
`-version`, `-versions`	Display the version number of the **grid-ca-sign** command. The second form includes details about the package containing **grid-ca-sign**.
`-in` *`REQUEST`*	Sign the request contained in the *`REQUEST`* file.
`-out` *`CERTIFICATE`*	Write the signed request to the *`CERTIFICATE`* file.
`-force`	Revoke any previously issued certificate with the same subject name as in the certificate request and issue a new certificate. Otherwise, **grid-ca-sign** will refuse to sign the request.
`-dir` *`DIRECTORY`*	Sign the certificate using the Simple CA certificate and configuration located in *`DIRECTORY`* instead of the default.
`-openssl-help`	Print the command-line options available for the **openssl ca** command.

Examples

Sign a certificate request using the simple CA in `$HOME/SimpleCA`

```
% grid-ca-sign -in usercert_request.pem -out usercert.pem -dir $HOME/SimpleCA

To sign the request
please enter the password for the CA key:

The new signed certificate is at: /home/juser/.globus/simpleCA/newcerts/01.pem
```

Environment Variables

The following environment variables affect the execution of **grid-ca-sign**:

`GLOBUS_LOCATION` Non-standard installation path of the Globus toolkit.

See Also

grid-cert-request(1), grid-ca-create(1), grid-default-ca(1), grid-ca-package(1)

Chapter 4. Security Considerations

1. Security considerations for SimpleCA

The operator of a CA must protect the private key of the CA. It should not be stored unencrypted or on a network filesystem.

Simple CA enforces the subject name policies in the simple CA's configuration files. If modified, the signing_policy file distributed to clients of the CA must also be modified.

Glossary

C

Certificate Authority (CA)	An entity that issues certificates.
CA Certificate	The CA's certificate. This certificate is used to verify signature on certificates issued by the CA. GSI typically stores a given CA certificate in `/etc/grid-security/certificates/<hash>.0`, where <hash> is the hash code of the CA identity.
certificate	A public key plus information about the certificate owner bound together by the digital signature of a CA. In the case of a CA certificate, the certificate is self signed, i.e. it was signed using its own private key.