

# **GT 5.0.2 GSI C: Developer's Guide**

---

## **GT 5.0.2 GSI C: Developer's Guide**

### **Introduction**

This component provides an API for authentication and two APIs for authorization.

The authentication API is an implementation of the GSS-API (RFC 2743 and RFC 2744) extended with the functions described in the GSS-API Extensions document.

On the authorization front there is a coarse-grained API, which in addition to authorizing also provides a mapping function, and an API that allows finer grained authorization decisions to be made. The finer grained API follows the subject, object, action paradigm.

Both of the authorization APIs allow different back end implementations through the use of dynamic library loading.

---

---

# Table of Contents

1. Before you begin .....	1
1. Feature summary .....	1
2. Tested platforms .....	1
3. Backward compatibility summary .....	1
4. Technology dependencies .....	1
5. Security considerations for GSI C .....	2
2. Usage scenarios .....	3
3. Tutorials .....	4
4. Architecture and design overview .....	5
1. Authentication .....	5
2. Authorization .....	6
5. APIs .....	7
6. Protocol Specifications .....	9
1. GSI Message Specification .....	9
I. GSI Commands .....	10
globus-update-certificate-dir .....	11
grid-cert-diagnostics .....	12
grid-cert-info .....	14
grid-cert-request .....	16
grid-default-ca .....	20
grid-change-pass-phrase .....	22
grid-proxy-init .....	23
grid-proxy-destroy .....	26
grid-proxy-info .....	27
grid-mapfile-add-entry .....	29
grid-mapfile-check-consistency .....	31
grid-mapfile-delete-entry .....	33
7. Configuring Certificates .....	35
1. Configuring Globus to Trust a Particular Certificate Authority .....	35
2. Configuring Globus to Create Appropriate Certificate Requests .....	36
3. Requesting Service Certificates .....	38
4. Configuring Credential Mappings .....	38
5. GSI File Permissions Requirements .....	40
8. Environment variable interface .....	42
1. Environmental Variables for GSI C .....	42
9. Debugging .....	46
10. Troubleshooting .....	47
1. Credential Troubleshooting .....	47
2. Grid map Troubleshooting .....	50
11. Related Documentation .....	51
Glossary .....	52

---

## List of Tables

7.1. CA files .....	35
7.2. Certificate request configuration files .....	36
7.3. Certificate request files .....	38
7.4. Gridmap File Location Algorithm .....	39
7.5. Authorization Configuration File Locations .....	40
7.6. Authorization Configuration File Locations .....	40
10.1. Credential Errors .....	48
10.2. Gridmap Errors .....	50

---

# Chapter 1. Before you begin

## 1. Feature summary

Features new in GT 5.0.2

- Support for processing host certificates containing X.509 subjectAltName extensions with dNSName or iPAddress values.

Other Supported Features

- Authentication of user using standard X.509 End Entity and *Proxy Certificates*.
- Delegation using X.509 Proxy Certificates.
- Pluggable authorization based on the client's certificate chain for GridFTP and GRAM2.
- Pluggable authorization for GRAM2 based on the RSL of the job.

Deprecated Features

- None

## 2. Tested platforms

Tested platforms for GSI C:

- i386 Linux

## 3. Backward compatibility summary

Protocol changes in GSI C since GT 5.0.2

- None

API changes since GT 5.0.2

- None

Exception changes since GT 5.0.2

- Not applicable

Schema changes since GT 5.0.2

- Not applicable

## 4. Technology dependencies

The GSI C component depends on the following GT components:

- C Common Libraries

The GSI C component depends on the following 3rd party software:

- OpenSSL

## 5. Security considerations for GSI C

- During host authorization, the toolkit treats host names of the form "hostname-*ANYTHING*.edu" as equivalent to "hostname.edu". This means that if a service was set up to do host authorization and hence accept the certificate "hostname.edu", it would also accept certificates with DNs "hostname-*ANYTHING*.edu".

The feature is in place to allow a multi-homed host following a "hostname-interface" naming convention, to have a single host certificate. For example, host "grid.test.edu" would also accept the likes of "grid-1.test.edu" or "grid-foo.test.edu".



### Note

The string *ANYTHING* matches only the name of the host and not domain components. This means that "hostname.edu" will not match "hostname-foo.sub.edu", but will match "host-foo.edu".



### Note

If a host was set up to accept "hostname-1.edu", it will not accept "hostname-*ANYTHING*.edu" but will accept "hostname.edu". That is, only one of the names being compared may contain the hyphen character in the host name.

A [bug](#)<sup>1</sup> has been opened to see if this feature needs to be modified.

In GT 5.0.2, it is possible to disable this behavior, by setting the environment variable `GLOBUS_GSS-API_NAME_COMPATIBILITY` to `STRICT_RFC2818`.

---

<sup>1</sup> [http://bugzilla.globus.org/bugzilla/show\\_bug.cgi?id=2969](http://bugzilla.globus.org/bugzilla/show_bug.cgi?id=2969)

---

# Chapter 2. Usage scenarios

There is no content available at this time.

---

# Chapter 3. Tutorials

There are no tutorials available at this time

---

# Chapter 4. Architecture and design overview

## 1. Authentication

As mentioned in the introduction, the GSI C security framework uses the GSSAPI API and extensions to it to abstract security mechanism specific details. Below the GSSAPI layer there exist multiple APIs for dealing with credential management, X.509 certificates in general and *proxy certificates* in particular as well as security configuration. Each of these APIs is described in more detail below.

The general design principle guiding these APIs is data encapsulation. Data structures (handles and attributes) capture and encapsulate the state of the system. These data structures are then acted upon by various getters and setters, as well as other functions.

### 1.1. The GSS Assist API

The GSS Assist API provides helper functions wrapping the process of security (GSS) context establishment, support for gridmap authorization and various other helper functions that wrap GSSAPI functions and capture common usage.

### 1.2. GSSAPI

The GSSAPI implementation provided by the toolkit is based upon SSL/TLS with extensions to the standard path validation mechanism to handle proxy certificates. It relies upon the credential and certificate utility APIs for general certificate acquisition and inspection functionality.

### 1.3. The Callback API

This API provides a callback that can be plugged into the OpenSSL path validation framework. This callback provides the additions to path validation required for dealing with proxy certificates and X.509 extensions. Furthermore, it allows applications to inspect data, e.g. the validated certificate chain, after the validation is done.

### 1.4. The Certificate Utilities API

The Certificate Utilities API provides helper functions for dealing with X.509 certificates. This API does not use the "handle" concept mentioned in the introduction. Rather, it operates on datatypes provided by the OpenSSL APIs.

### 1.5. The Credential API

The Credential API deals with reading and writing certificates from and to the file system and the OpenSSL I/O abstraction layer. It also provides functions for inspecting and validating the read credentials.

### 1.6. The Proxy APIs

The Proxy APIs provide an implementation of the X.509 Proxy Certificate Extension ASN.1 structure as well as functions for creating new proxies.

## 1.7. The System Configuration API

This API serves as a abstraction layer for OS specific information needed by the security infrastructure. It provides OS specific functions for discovering certificates from a set of predefined standard locations as well as functions for doing the same for various configuration files.

## 2. Authorization

As described in the introduction the GSI C security framework essentially provides two authorization APIs, the generic Authorization API and the Gridmap API. These APIs differ in various ways:

The Authorization API provides a framework that allows callouts to 3rd party authorization solutions, does not provide a default authorization mechanism and is geared to authorizing the subject-action-object tuple.

The Gridmap API on the other hand, while allowing for custom callouts to be plugged in and override the default behavior, provides a default authorization and mapping mechanism based on the *grid map file*. Also, it only furnishes the callouts with information about the entity to be authorized, i.e. it does not provide information on the action and the object, so it is somewhat simpler in its approach. Finally, it provides the ability to map authorized entities to local system entities, e.g. UNIX user names. More information on the interface used for Gridmap callouts can be found [here](#)<sup>1</sup>.

---

<sup>1</sup> ../GSIAuthorizationCalloutSpecification-04.pdf

---

# Chapter 5. APIs

Documentation for the APIs in this component can be found here:

- [gaa\\_core](#)<sup>1</sup> [[no frames](#)<sup>2</sup>]
- [gaa\\_gss\\_generic](#)<sup>3</sup> [[no frames](#)<sup>4</sup>]
- [gaa\\_plugin](#)<sup>5</sup> [[no frames](#)<sup>6</sup>]
- [globus\\_authz](#)<sup>7</sup> [[no frames](#)<sup>8</sup>]
- [globus\\_authz\\_callout\\_error](#)<sup>9</sup> [[no frames](#)<sup>10</sup>]
- [globus\\_gridmap\\_callout\\_error](#)<sup>11</sup> [[no frames](#)<sup>12</sup>]
- [globus\\_gsi\\_callback](#)<sup>13</sup> [[no frames](#)<sup>14</sup>]
- [globus\\_gsi\\_cert\\_utils](#)<sup>15</sup> [[no frames](#)<sup>16</sup>]
- [globus\\_gsi\\_credential](#)<sup>17</sup> [[no frames](#)<sup>18</sup>]
- [globus\\_gsi\\_openssl\\_error](#)<sup>19</sup> [[no frames](#)<sup>20</sup>]
- [globus\\_gsi\\_proxy\\_core](#)<sup>21</sup> [[no frames](#)<sup>22</sup>]
- [globus\\_gsi\\_proxy\\_ssl](#)<sup>23</sup> [[no frames](#)<sup>24</sup>]
- [globus\\_gsi\\_sysconfig](#)<sup>25</sup> [[no frames](#)<sup>26</sup>]
- [globus\\_gss\\_assist](#)<sup>27</sup> [[no frames](#)<sup>28</sup>]

---

<sup>1</sup> [http://www.globus.org/api/c-globus-4.0/gaa\\_core/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/gaa_core/html/index.html#_top)

<sup>2</sup> [http://www.globus.org/api/c-globus-4.0/gaa\\_core/html/main.html#\\_top](http://www.globus.org/api/c-globus-4.0/gaa_core/html/main.html#_top)

<sup>3</sup> [http://www.globus.org/api/c-globus-4.0/gaa\\_gss\\_generic/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/gaa_gss_generic/html/index.html#_top)

<sup>4</sup> [http://www.globus.org/api/c-globus-4.0/gaa\\_gss\\_generic/html/main.html#\\_top](http://www.globus.org/api/c-globus-4.0/gaa_gss_generic/html/main.html#_top)

<sup>5</sup> [http://www.globus.org/api/c-globus-4.0/gaa\\_plugin/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/gaa_plugin/html/index.html#_top)

<sup>6</sup> [http://www.globus.org/api/c-globus-4.0/gaa\\_plugin/html/main.html#\\_top](http://www.globus.org/api/c-globus-4.0/gaa_plugin/html/main.html#_top)

<sup>7</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_authz/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/globus_authz/html/index.html#_top)

<sup>8</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_authz/html/main.html](http://www.globus.org/api/c-globus-4.0/globus_authz/html/main.html)

<sup>9</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_authz\\_callout\\_error/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/globus_authz_callout_error/html/index.html#_top)

<sup>10</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_authz\\_callout\\_error/html/main.html](http://www.globus.org/api/c-globus-4.0/globus_authz_callout_error/html/main.html)

<sup>11</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gridmap\\_callout\\_error/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/globus_gridmap_callout_error/html/index.html#_top)

<sup>12</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gridmap\\_callout\\_error/html/main.html](http://www.globus.org/api/c-globus-4.0/globus_gridmap_callout_error/html/main.html)

<sup>13</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gsi\\_callback/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/globus_gsi_callback/html/index.html#_top)

<sup>14</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gsi\\_callback/html/main.html](http://www.globus.org/api/c-globus-4.0/globus_gsi_callback/html/main.html)

<sup>15</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gsi\\_cert\\_utils/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/globus_gsi_cert_utils/html/index.html#_top)

<sup>16</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gsi\\_cert\\_utils/html/main.html](http://www.globus.org/api/c-globus-4.0/globus_gsi_cert_utils/html/main.html)

<sup>17</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gsi\\_credential/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/globus_gsi_credential/html/index.html#_top)

<sup>18</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gsi\\_credential/html/main.html](http://www.globus.org/api/c-globus-4.0/globus_gsi_credential/html/main.html)

<sup>19</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gsi\\_openssl\\_error/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/globus_gsi_openssl_error/html/index.html#_top)

<sup>20</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gsi\\_openssl\\_error/html/main.html](http://www.globus.org/api/c-globus-4.0/globus_gsi_openssl_error/html/main.html)

<sup>21</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gsi\\_proxy\\_core/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/globus_gsi_proxy_core/html/index.html#_top)

<sup>22</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gsi\\_proxy\\_core/html/main.html](http://www.globus.org/api/c-globus-4.0/globus_gsi_proxy_core/html/main.html)

<sup>23</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gsi\\_proxy\\_ssl/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/globus_gsi_proxy_ssl/html/index.html#_top)

<sup>24</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gsi\\_proxy\\_ssl/html/main.html](http://www.globus.org/api/c-globus-4.0/globus_gsi_proxy_ssl/html/main.html)

<sup>25</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gsi\\_sysconfig/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/globus_gsi_sysconfig/html/index.html#_top)

<sup>26</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gsi\\_sysconfig/html/main.html](http://www.globus.org/api/c-globus-4.0/globus_gsi_sysconfig/html/main.html)

<sup>27</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gss\\_assist/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/globus_gss_assist/html/index.html#_top)

<sup>28</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gss\\_assist/html/main.html](http://www.globus.org/api/c-globus-4.0/globus_gss_assist/html/main.html)

- [globus\\_gssapi\\_gsi](#)<sup>29</sup> [[no frames](#)<sup>30</sup>]
- [globus\\_openssl\\_module](#)<sup>31</sup> [[no frames](#)<sup>32</sup>]
- [gssapi\\_error](#)<sup>33</sup> [[no frames](#)<sup>34</sup>]

For information on the internationalization API, see the [CCommon Libraries Public Interface](#).

---

<sup>29</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gssapi\\_gsi/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/globus_gssapi_gsi/html/index.html#_top)

<sup>30</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_gssapi\\_gsi/html/main.html](http://www.globus.org/api/c-globus-4.0/globus_gssapi_gsi/html/main.html)

<sup>31</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_openssl\\_module/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/globus_openssl_module/html/index.html#_top)

<sup>32</sup> [http://www.globus.org/api/c-globus-4.0/globus\\_openssl\\_module/html/main.html](http://www.globus.org/api/c-globus-4.0/globus_openssl_module/html/main.html)

<sup>33</sup> [http://www.globus.org/api/c-globus-4.0/gssapi\\_error/html/index.html#\\_top](http://www.globus.org/api/c-globus-4.0/gssapi_error/html/index.html#_top)

<sup>34</sup> [http://www.globus.org/api/c-globus-4.0/gssapi\\_error/html/main.html](http://www.globus.org/api/c-globus-4.0/gssapi_error/html/main.html)

---

# Chapter 6. Protocol Specifications

## 1. GSI Message Specification

The GSSAPI implementation contained in this component produces security tokens that follow an extended version of the SSL/TLS protocol. More information about the protocol can be found [here](#)<sup>1</sup>.

---

<sup>1</sup> ../GSI-message-specification-02.doc

---

# GSI Commands

---

# Name

globus-update-certificate-dir -- Update symlinks in the trusted CA directory

globus-update-certificate-dir [-help] [-d *DIRECTORY*]

# Description

The **globus-update-certificate-dir** program creates symlinks between files (CA certificates, certificate revocation lists, signing policy, and certificate request configuration files) using the certificate hash the installed version of OpenSSL uses. OpenSSL 1.0.0 uses a different name hashing algorithm than previous versions, so CA distributions created with older versions of OpenSSL might not be able to locate trusted CAs and related files. Running **globus-update-certificate-dir** against a trusted CA directory will add symlinks to the files to the hash if needed.



## Note

To run globus-update-certificate-dir on Linux, modify that script so that the first line is

```
#!/usr/bin/env perl
```

instead of

```
#!/usr/bin/env perl -w
```

The full set of command-line options to **globus-update-certificate-dir** consists of:

**-help**            Display a help message to standard output and exit

**-d *DIRECTORY***    Create links in the trusted CA directory *DIRECTORY* instead of using the default search path.

# Environment

If the following variables affect the execution of **globus-update-certificate-dir**

**X509\_CERT\_DIR**    Default trusted certificate directory.

**HOME**            Path to the current user's home directory.

**GLOBUS\_LOCATION** Path to the Globus installation.

---

# Name

grid-cert-diagnostics -- Print diagnostic information about certificates and keys

grid-cert-diagnostics [-h] | [-help] [-p]

# Description

The **grid-cert-diagnostics** program displays information about the current user's security environment, including information about security-related environment variables, security directory search path, personal key and certificates, and trusted certificates. It is intended to provide information to help diagnose problems using GSIC.

By default, **grid-cert-diagnostics** prints out information regarding the environment and trusted certificate directory. If the `-p` command-line option is used, then additional information about the current user's default certificate and key will be printed.

The full set of command-line options to **grid-cert-diagnostics** consists of:

`-h,`      Display a help message and exit.

`-help`

`-p`      Display information about the personal certificate and key that is the current user's default credential.

# Examples

In this example, we see the default mode of checking the default security environment for the system, without processing the user's key and certificate. Note the user receives a warning about a `cog.properties` and about an expired CA certificate.

```
% grid-cert-diagnostics
```

```
Checking Environment Variables
```

```
=====
```

```
Checking if X509_CERT_DIR is set... no
Checking if X509_USER_CERT is set... no
Checking if X509_USER_KEY is set... no
Checking if X509_USER_PROXY is set... no
```

```
Checking Security Directories
```

```
=====
```

```
Determining trusted cert path... /etc/grid-security/certificates
Checking for cog.properties... found
    WARNING: If the cog.properties file contains security properties,
             Java apps will ignore the security paths described in the GSI
             documentation
```

```
Checking trusted certificates...
```

```
=====
```

```
Getting trusted certificate list...
Checking CA file /etc/grid-security/certificates/1c4f4c48.0... ok
Verifying certificate chain for "/etc/grid-security/certificates/1c3f2ca8.0"... ok
Checking CA file /etc/grid-security/certificates/9d8788eb.0... ok
```

```
Verifying certificate chain for "/etc/grid-security/certificates/9d8753eb.0"... failed
  globus_credential: Error verifying credential: Failed to verify credential
  globus_gsi_callback_module: Could not verify credential
  globus_gsi_callback_module: The certificate has expired:
  Credential with subject: /DC=org/DC=example/OU=grid/CN=CA has expired.
```

In this example, we show a user with a mismatched private key and certificate:

```
% grid-cert-diagnostics -p
```

```
Checking Environment Variables
```

```
=====
```

```
Checking if X509_CERT_DIR is set... no
Checking if X509_USER_CERT is set... no
Checking if X509_USER_KEY is set... no
Checking if X509_USER_PROXY is set... no
```

```
Checking Security Directories
```

```
=====
```

```
Determining trusted cert path... /etc/grid-security/certificates
Checking for cog.properties... not found
```

```
Checking Default Credentials
```

```
=====
```

```
Determining certificate and key file names... ok
Certificate Path: "/home/juser/.globus/usercert.pem"
Key Path: "/home/juser/.globus/userkey.pem"
Reading certificate... ok
Reading private key...
ok
Checking Certificate Subject...
"/O=Grid/OU=Example/OU=User/CN=Joe User"
Checking cert... ok
Checking key... ok
Checking that certificate contains an RSA key... ok
Checking that private key is an RSA key... ok
Checking that public and private keys have the same modulus... failed
Private key modulus: D294849E37F048C3B5ACEEF2CCDF97D88B679C361E29D5CB5
219C3E948F3E530CFC609489759E1D751F0ACFF0515A614276A0F4C11A57D92D7165B8
FA64E3140155DE448D45C182F4657DA13EDA288423F5B9D169DFF3822EFD81EB2E6403
CE3CB4CCF96B65284D92592BB1673A18354DA241B9AFD7F494E54F63A93E15DCAE2
Public key modulus : C002C7B329B13BFA87BAF214EACE3DC3D490165ACEB791790
600708C544175D9193C9BAC5AED03B7CB49BB6AE6D29B7E635FAC751E9A6D1CEA98022
6F1B63002902D6623A319E4682E7BFB0968DCE962CF218AAD95FAAD6A0BA5C42AA9AAF
7FDD32B37C6E2B2FF0E311310AA55FFB9EAFDF5B995C7D9EEAD8D5D81F3531E0AE5
Certificate and and private key don't match
```

---

# Name

grid-cert-info -- Display information about a certificate

```
grid-cert-info [-help] [-usage] [-version] [-versions]
grid-cert-info [-file CERTIFICATE-FILE] [-rfc2253] [-all]
[-subject] | [-s]
[-issuer] | [-i]
[-issuerhash] | [-ih]
[-startdate] | [-sd]
[-enddate] | [-ed]
```

# Description

The **grid-cert-info** program displays information contained within a certificate file. By default it shows a text representation of the entire certificate. Specific facts about the certificate can be shown instead by using command-line options. If any of those options are used, then the default display is suppressed. This can be added to the output by using the `-all` command-line option.

If multiple display options are included on the command-line, the facts related to those will be displayed on separate lines in the order that they occur. If an option is specified multiple time, that fact will be displayed multiple times.

The full set of command-line options to **grid-cert-info** are:

<code>-help, -usage</code>	Display the command-line options to <b>grid-cert-info</b> and exit.
<code>-version, -versions</code>	Display the version number of the <b>grid-cert-info</b> command. The second form includes more details.
<code>-file <i>CERTIFICATE-FILE</i></code>	Display information about the first certificate contained in the file named by <i>CERTIFICATE-FILE</i> instead of the default user certificate.
<code>-rfc2253</code>	Display X.509 distinguished names using the string representation defined in RFC 2253 instead of the default OpenSSL oneline format.
<code>-all</code>	Display the text representation of the entire certificate in addition to any other facts requested by command-line options. This is the default if no fact-specific command-line options are used.
<code>-subject, -s</code>	Display the subject name of the X.509 certificate.
<code>-issuer, -i</code>	Display the issuer name of the X.509 certificate.
<code>-issuerhash, -ih</code>	Display the default hash of the issuer name of the X.509 certificate. This can be used to locate which CA certificate in the trusted certificate directory issued the certificate being inspected.
<code>-startdate, -sd</code>	Display a string representation of the date and time when the certificate is valid from. This is displayed in the format used by the OpenSSL <b>x509</b> command.
<code>-enddate, -ed</code>	Display a string representation of the date and time when the certificate is valid until. This is displayed in the format used by the OpenSSL <b>x509</b> command.

## Examples

Display the validity times for the default certificate

```
% grid-cert-info -sd -ed
Aug 31 12:33:47 2009 GMT
Aug 31 12:33:47 2010 GMT
```

Display the same information about a different certificate specified on the command-line

```
% grid-cert-info -sd -ed -f /etc/grid-security/hostcert.pem
Jan 21 12:24:48 2003 GMT
Jul 15 11:30:57 2020 GMT
```

Display the subject of a certificate in both the default and the RFC 2253 forms.

```
% grid-cert-info -subject
/DC=org/DC=example/DC=grid/CN=Joe User
% grid-cert-info -subject -rfc2253
CN=Joe User,DC=grid,DC=example,DC=org
```

## Environment Variables

The following environment variables affect the execution of **grid-cert-info**:

**X509\_USER\_CERT** Path to the default certificate file to inspect.

---

# Name

grid-cert-request -- Generate a X.509 certificate request and corresponding private key

```
grid-cert-request [-help] [-h] [-?] [-usage]
[-version] [-versions]
grid-cert-request [ -cn NAME | -commonname NAME ]
[-dir DIRECTORY] [-prefix PREFIX]
[ -nopw | -nodes | -nopassphrase ]
[ -nopw | -nodes | -nopassphrase ]
[-ca [HASH]] [-verbose] [ -interactive | -int ] [-force]
grid-cert-request -host FQDN [-service SERVICE] [-dns FQDN...] [-ip IP-ADDRESS...]
[-dir DIRECTORY] [-prefix PREFIX]
[-ca [HASH]] [-verbose] [ -interactive | -int ] [-force]
```

# Description

The **grid-cert-request** program generates an X.509 Certificate Request and corresponding private key for the specified name, host, or service. It is intended to be used with a CA implemented using the `globus_simple_ca` package.

The default behavior of **grid-cert-request** is to generate a certificate request and private key for the user running the command. The subject name is derived from the `gecos` information in the local system's password database, unless the `-commonname`, `-cn`, or `-host` command-line options are used.

By default, **grid-cert-request** writes user certificate requests and keys to the `$HOME/.globus` directory, and host and service certificate requests and keys to `/etc/grid-security`. This can be overridden by using the `-dir` command-line option.

The full set of command-line options to **grid-cert-request** are:

- `-help, -h, -?, -usage`    Display the command-line options to **grid-cert-request** and exit.
- `-version, -versions`    Display the version number of the **grid-cert-request** command. The second form includes more details.
- `-cn NAME, -common-name NAME`    Create a certificate request with the common name component of the subject set to *NAME*. This is used to create user identity certificates.
- `-dir DIRECTORY`    Write the certificate request and key to files in the directory specified by *DIRECTORY*.
- `-prefix PREFIX`    Use the string *PREFIX* as the base name of the certificate, `certificate_request`, and key files instead of the default. For a user certificate request, this would mean creating files `$HOME/.globus/PREFIXcert_request.pem`, `$HOME/.globus/PREFIXcert.pem`, and `$HOME/.globus/PREFIXkey.pem`.
- `-ca CA-HASH`    Use the certificate request configuration for the CA with the name hash *CA-HASH* instead of the default CA chosen by running **grid-default-ca**.
- `-verbose`    Keep the output from the OpenSSL certificate request command visible after it completes, instead of clearing the screen..
- `-interactive, -int`    Prompt for each component of the subject name of the request, instead of generating the common name from other command-line options. Note that CAs may not sign certificates for subject names that don't match their signing policies.

- `-force` Overwrite any existing certificate request and private key with a new one.
- `-nopw, -nodes, -no-passphrase` Create an unencrypted private key for the certificate instead of prompting for a passphrase. This is the default behavior for host or service certificates, but not recommended for user certificates.
- `-host FQDN` Create a certificate request for use on a particular host. This option also causes the private key associated with the certificate request to be unencrypted. The *FQDN* argument to this option should be the fully qualified domain name of the host that will use this certificate. The subject name of the certificate will be derived from the *FQDN* and the service option if specified by the `-service` command-line option. If the host for the certificate has multiple names, then use either the `-dns` or `-ip` command-line options to add alternate names or addresses to the certificates.
- `-service SERVICE` Create a certificate request for a particular service on a host. The subject name of the certificate will be derived from the *FQDN* passed as the argument to the `-host` command-line option and the *SERVICE* string.
- `-dns FQDN,...` Create a certificate request containing a `subjectAltName` extension containing one or more host names. This is used when a certificate may be used by multiple virtual servers or if a host has different names when contacted within or outside a private network. Multiple DNS names can be included in the extension by separating them with a comma.
- `-ip IP-ADDRESS,...` Create a certificate request containing a `subjectAltName` extension containing the IP addresses named by the *IP-ADDRESS* strings. This is used when a certificate may be used by services listening on multiple networks. Multiple IP addresses can be included in the extension by separating them with a comma.

## Examples

Create a user certificate request:

```
% grid-cert-request
A certificate request and private key is being created.
You will be asked to enter a PEM pass phrase.
This pass phrase is akin to your account password,
and is used to protect your key file.
If you forget your pass phrase, you will need to
obtain a new certificate.
A private key and a certificate request has been generated with the subject:

/O=org/OU=example/OU=grid/CN=Joe User
```

If the `CN=Joe User` is not appropriate, rerun this script with the `-force -cn "Common Name"` options.

Your private key is stored in `/home/juser/.globus/userkey.pem`  
Your request is stored in `/home/juser/.globus/usercert_request.pem`

Please e-mail the request to the Example CA `ca@grid.example.org`  
You may use a command similar to the following:

```
cat /home/juser/.globus/usercert_request.pem | mail ca@grid.example.org
```

Only use the above if this machine can send AND receive e-mail. if not, please mail using some other method.

Your certificate will be mailed to you within two working days.  
If you receive no response, contact Example CA at ca@grid.example.org

Create a host certificate for a host with two names.

```
% grid-cert-request -host grid.example.org -dns grid.example.org,grid-internal.example.org
```

A private host key and a certificate request has been generated with the subject:

```
/O=org/OU=example/OU=grid/CN=host/grid.example.org
```

-----  
The private key is stored in /etc/grid-security/hostkey.pem  
The request is stored in /etc/grid-security/hostcert\_request.pem

Please e-mail the request to the Example CA ca@grid.example.org  
You may use a command similar to the following:

```
cat /etc/grid-security/hostcert_request.pem | mail ca@grid.example.org
```

Only use the above if this machine can send AND receive e-mail. if not, please mail using some other method.

Your certificate will be mailed to you within two working days.  
If you receive no response, contact Example CA at ca@grid.example.org

## Environment Variables

The following environment variables affect the execution of **grid-cert-request**:

X509_CERT_DIR	Path to the directory containing SSL configuration files for generating certificate requests.
GRID_SECURITY_DIR	Path to the directory containing SSL configuration files for generating certificate requests. This value is used if X509_CERT_DIR is not set.
GLOBUS_LOCATION	Path to the directory containing the Globus Toolkit. This is searched if neither the X509_CERT_DIR nor the GRID_SECURITY_DIR environment variables are set.

## Files

\$HOME/.globus/usercert\_request.pem Default path to write a user certificate request.

\$HOME/.globus/usercert.pem Default path to write a user certificate.

\$HOME/.globus/userkey.pem Default path to write a user private key.

<code>/etc/grid-security/host-cert_request.pem</code>	Default path to write a host certificate request.
<code>/etc/grid-security/host-cert.pem</code>	Default path to write a host certificate.
<code>/etc/grid-security/hostkey.pem</code>	Default path to write a host private key.
<code>TRUSTED-CERT-DIR/globus-user-ssl.conf</code> , <code>TRUSTED-CERT-DIR/globus-user-ssl.conf.CA-HASH</code>	SSL configuration file for requesting a user certificate. The first form is the default location, the second form is used when the <code>-ca</code> command-line option is specified.
<code>TRUSTED-CERT-DIR/globus-host-ssl.conf</code> , <code>TRUSTED-CERT-DIR/globus-host-ssl.conf.CA-HASH</code>	SSL configuration file for requesting a host or service certificate. The first form is the default location, the second form is used when the <code>-ca</code> command-line option is specified.

---

# Name

grid-default-ca -- Select default CA for certificate requests

```
grid-default-ca [-help] [-h] [-usage] [-u] [-version] [-versions]
grid-default-ca -list [-dir CA-DIRECTORY]
grid-default-ca [-ca CA-HASH] [-dir CA-DIRECTORY]
```

# Description

The **grid-default-ca** program sets the default certificate authority to use when the **grid-cert-request** script is run. The CA's certificate, configuration, and signing policy must be installed in the trusted certificate directory to be able to request certificates from that CA. Note that some CAs have different policies and use other tools to handle certificate requests. Please consult your CA's support staff if you are unsure. The **grid-default-ca** is designed to work with CAs implemented using the `globus_simple_ca` package.

By default, the **grid-default-ca** program displays a list of installed CA certificates and prompts the user for which one to set as the default. If invoked with the `-list` command-line option, **grid-default-ca** will print the list and not prompt nor set the default CA. If invoked with the `-ca` option, it will not list or prompt, but set the default CA to the one with the hash that matches the `CA-HASH` argument to that option. If **grid-default-ca** is used to set the default CA, the caller of this program must have write permissions to the trusted certificate directory.

The **grid-default-ca** program sets the CA in one of the grid security directories. It looks in the directory named by the `GRID_SECURITY_DIR` environment, the `X509_CERT_DIR`, `/etc/grid-security`, and `$GLOBUS_LOCATION/share/certificates`.

The full set of command-line options to **grid-default-ca** are:

- `-help, -h, -usage, -u` Display the command-line options to **grid-default-ca** and exit.
- `-version, -versions` Display the version number of the **grid-default-ca** command. The second form includes more details.
- `-dir CA-DIRECTORY` Use the trusted certificate directory named by `CA-DIRECTORY` instead of the default.
- `-list` Instead of changing the default CA, print out a list of all available CA certificates in the trusted certificate directory
- `-ca CA-HASH` Set the default CA without displaying the list of choices or prompting. The CA file named by `CA-HASH` must exist.

# Examples

List the contents of the trusted certificate directory that contain the string Example:

```
% grid-default-ca | grep Example
15) cd1186ff - /DC=org/DC=Example/DC=Grid/CN=Example CA
```

Choose that CA as the default:

```
% grid-default-ca -ca cd1186ff
```

```
setting the default CA to: /DC=org/DC=Example/DC=Grid/CN=Example CA
```

```
linking /etc/grid-security/certificates/grid-security.conf.cd1186ff to
/etc/grid-security/certificates/grid-security.conf

linking /etc/grid-security/certificates/grid-host-ssl.conf.cd1186ff to
/etc/grid-security/certificates/grid-host-ssl.conf

linking /etc/grid-security/certificates/grid-user-ssl.conf.cd1186ff to
/etc/grid-security/certificates/grid-user-ssl.conf

...done.
```

## Environment Variables

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

GRID_SECURITY_DIRECTORY	Path to the default trusted certificate directory.
X509_CERT_DIR	Path to the default trusted certificate directory.
GLOBUS_LOCATION	Path to the Globus Toolkit installation directory.

## Bugs

The **grid-default-ca** program displays CAs from all of the directories in its search list; however, **grid-cert-request** only uses the first which contains a grid security configuration.

The **grid-default-ca** program may display the same CA multiple times if it is located in multiple directories in its search path. However, it does not provide any information about which one would actually be used by the **grid-cert-request** command.

## See Also

grid-cert-request(1)

---

# Name

`grid-change-pass-phrase --` Change the passphrase of a private key

```
grid-change-pass-phrase [-help] [-usage] [-version] [-versions]
grid-change-pass-phrase [-file PRIVATE-KEY]
```

# Description

The **grid-change-pass-phrase** program changes the passphrase protecting a private key or PKCS12 bundle containing a private key and certificate. By default, **grid-change-pass-phrase** uses the `X509_USER_KEY` environment variable to locate the private key. If that is not set, then it looks for `$HOME/.globus/userkey.pem` and `$HOME/.globus/usercred.p12` in succession. The path to a key can be specified by using the `-file` command-line option.

The full set of command-line options to **grid-change-pass-phrase** are:

- `-help, -usage`            Display the command-line options to **grid-change-pass-phrase** and exit.
- `-version, -versions`    Display the version number of the **grid-change-pass-phrase** command. The second form includes more details.
- `-file PRIVATE-KEY`      Change the passphrase of the private key named by `PRIVATE-KEY` instead of the default.

# Examples

Change the passphrase of the default private key:

```
% grid-change-pass-phrase
```

```
Enter pass phrase for /home/juser/.globus/userkey.pem:
writing RSA key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

# Environment Variables

The following environment variables affect the execution of **grid-change-pass-phrase**:

`X509_USER_KEY`    Path to the default private key file.

---

# Name

grid-proxy-init -- Generate a new proxy certificate

```
grid-proxy-init [-help] [-usage] [-version]
grid-proxy-init [-debug] [-q] [-verify]
[[-valid HOURS:MINUTES] | [-hours HOURS]] [-cert CERTFILE] [-key KEYFILE] [-certdir CERTDIR] [-out
PROXYPATH] [-bits BITS]
[-policy POLICYFILE]
[[-pl POLICY-OID] | [-policy-language POLICY-OID]] [-path-length MAXIMUM] [-pwstdin] [-limited] [-independent]
[[-draft] | [-old] | [-rfc]]
```

# Description

The **grid-proxy-init** program generates X.509 proxy certificates derived from the currently available certificate files. By default, this command generates a [RFC 3820](http://www.ietf.org/rfc/rfc3820.txt)<sup>1</sup> Proxy Certificate with a 512 bit key valid for 12 hours in a file named `/tmp/x509up_#UID`. Command-line options and variables can modify the format, strength, lifetime, and location of the generated proxy certificate.

X.509 proxy certificates are short-lived certificates, signed usually by a user's identity certificate or another proxy certificate. The key associated with a proxy certificate is unencrypted, so applications can authenticate using a proxy identity without providing a passphrase.

Proxy certificates provide a convenient alternative to constantly entering passwords, but are also less secure than the user's normal security credential. Therefore, they should always be user-readable only (this is enforced by the GSI libraries), and should be deleted after they are no longer needed.

This version of **grid-proxy-init** supports three different proxy formats: the old proxy format used in early releases of the Globus Toolkit up to version 2.4.x, an IETF draft version of X.509 Proxy Certificate profile used in Globus Toolkit 3.0.x and 3.2.x, and the RFC 3820 profile used in Globus Toolkit Version 4.0.x and 4.2.x. By default, this version of **grid-proxy-init** creates an RFC 3820 compliant proxy. To create a proxy compatible with older versions of the Globus Toolkit, use the `-old` or `-draft` command-line options.

The full set of command-line options to **grid-proxy-init** are:

<code>-help, -usage</code>	Display the command-line options to <b>grid-proxy-init</b> .
<code>-version</code>	Display the version number of the <b>grid-proxy-init</b> command
<code>-debug</code>	Display information about the path to the certificate and key used to generate the proxy certificate, the path to the trusted certificate directory, and verbose error messages
<code>-q</code>	Suppress all output from <b>grid-proxy-init</b> except for passphrase prompts.
<code>-verify</code>	Perform certificate chain validity checks on the generated proxy.
<code>-valid HOURS:MINUTES,</code> <code>-hours HOURS</code>	Create a certificate that is valid for <i>HOURS</i> hours and <i>MINUTES</i> minutes. If not specified, the default of twelve hours and no minutes is used.

---

<sup>1</sup> <http://www.ietf.org/rfc/rfc3820.txt>

---

<code>-cert CERTFILE, -key KEY-FILE</code>	Create a proxy certificate signed by the certificate located in <code>CERTFILE</code> using the key located in <code>KEYFILE</code> . If not specified the default certificate and key will be used. This overrides the values of environment variables described below.
<code>-certdir CERTDIR</code>	Search <code>CERTDIR</code> for trusted certificates if verifying the proxy certificate. If not specified, the default trusted certificate search path is used. This overrides the value of the <code>X509_CERT_DIR</code> environment variable
<code>-out PROXYPATH</code>	Write the generated proxy certificate file to <code>PROXYPATH</code> instead of the default path of <code>/tmp/x509up_uUID</code> .
<code>-bits BITS</code>	When creating the proxy certificate, use a <code>BITS</code> bit key instead of the default 512 bit keys.
<code>-policy POLICYFILE</code>	Add the certificate policy data described in <code>POLICYFILE</code> as the ProxyCertInfo X.509 extension to the generated proxy certificate.
<code>-pl POLICY-OID, -policy-language POLICY-OID</code>	Set the policy language identifier of the policy data specified by the <code>-policy</code> command-line option to the oid specified by the <code>POLICY-OID</code> string.
<code>-path-length MAXIMUM</code>	Set the maximum length of the chain of proxies that can be created by the generated proxy to <code>MAXIMUM</code> . If not set, the default of an unlimited proxy chain length is used.
<code>-pwstdin</code>	Read the private key's passphrase from stdin instead of reading input from the controlling tty. This is useful when scripting <b>grid-proxy-init</b> .
<code>-limited</code>	Create a limited proxy. Limited proxies are generally refused by process-creating services, but may be used to authorize with other services.
<code>-independent</code>	Create an independent proxy. An independent proxy is not treated as an impersonation proxy but as a separate identity for authorization purposes.
<code>-draft</code>	Create a IETF draft proxy instead of the default RFC 3280-compliant proxy. This type of proxy uses a non-standard proxy policy identifier. This might be useful for authenticating with older versions of the Globus Toolkit.
<code>-old</code>	Create a legacy proxy instead of the default RFC 3280-compliant proxy. This type of proxy uses a non-standard method of indicating that the certificate is a proxy and whether it is limited. This might be useful for authenticating with older versions of the Globus Toolkit.
<code>-rfc</code>	Create an RFC 3820-compliant proxy certificate. This is the default for this version of <b>grid-proxy-init</b> .

## Examples

To create a proxy with the default lifetime and format, run the **grid-proxy-init** program with no arguments. For example:

```
% grid-proxy-init
Your identity: /DC=org/DC=example/CN=Joe User
Enter GRID pass phrase for this identity:
Creating proxy ..... Done
Your proxy is valid until: Thu Mar 18 03:48:05 2010
```

To create a stronger proxy that lasts for only 8 hours, use the `-hours` and `-bits` command-line options to **grid-proxy-init**. For example:

```
% grid-proxy-init -hours 8 -bits 1024
Your identity: /DC=org/DC=example/CN=Joe User
Enter GRID pass phrase for this identity:
Creating proxy ..... Done
Your proxy is valid until: Thu Mar 17 23:48:05 2010
```

## Environment Variables

The following environment variables affect the execution of **grid-proxy-init**:

`X509_USER_CERT` Path to the certificate to use as issuer of the new proxy.

`X509_USER_KEY` Path to the key to use to sign the new proxy.

`X509_CERT_DIR` Path to the directory containing trusted certificate certificates and signing policies.

## Files

The following files affect the execution of **grid-proxy-init**:

`$HOME/.globus/user-cert.pem` Default path to the certificate to use as issuer of the new proxy.

`$HOME/.globus/userkey.pem` Default path to the key to use to sign the new proxy.

## Compatibility

For more information about proxy certificate types and their compatibility in GT, see <http://dev.globus.org/wiki/Security/ProxyCertTypes>.

## See Also

`grid-proxy-destroy(1)`, `grid-proxy-info(1)`

---

# Name

`grid-proxy-destroy --` Destroy the default proxy certificate

```
grid-proxy-destroy [-help] [-usage] [-version]
grid-proxy-destroy [-debug] [-dryrun] [-default] [-all] [--] [FILENAME...]
```

# Description

The **grid-proxy-destroy** program removes X.509 proxy files from the local filesystem. It overwrites the data in the files and removes the files from the filesystem. By default, it removes the current user's default proxy (either `/tmp/x509up_uid` where `UID` is the current POSIX user id, or the file pointed to by the `X509_USER_PROXY` environment variable) unless a list of proxy file paths are included as part of the command line.

Use the `--` command-line option to separate a list of proxy paths from command line options if the proxy file begins with the `-` character.

The full list of command-line options to **grid-proxy-destroy** are:

- `-help,`     Display the command-line options to **grid-proxy-destroy**.
- `-usage`
  
- `-version`   Display the version number of the **grid-proxy-destroy** command
  
- `-debug`     Display verbose error messages.
  
- `-dryrun`    Do not remove the proxy, but display the path of the files that would have been removed, or the directory where they would have been removed from if the `-all` command-line option is used.
  
- `-default`   Remove the default proxy in addition to the files included on the command-line. Only needed if other paths are included on the command-line.
  
- `-all`        Remove the default proxy and all delegated proxies in the temporary file directory.

# Environment Variables

The following environment variables affect the execution of **grid-proxy-destroy**:

`X509_USER_PROXY`   Path to the default user proxy.

# See Also

`grid-proxy-init(1)`, `grid-proxy-info(1)`

---

# Name

grid-proxy-info -- Display information about a proxy certificate

```
grid-proxy-info [-help] [-usage] [-version]
grid-proxy-info [[-subject] | [-s]]
[[-issuer] | [-i]]
[-identity] [-type] [-timeleft] [-strength] [-all] [-text] [-path] [-rfc2253]
[[-exists] | -e ]
[[-valid HOURS:MINUTES] | [-v HOURS:MINUTES]]
[[-hours HOURS] | [-h HOURS]]
[[-bits BITS] | [-b BITS]]
```

# Description

The **grid-proxy-info** program extracts information from an X.509 proxy certificates, and optionally displays or returns an exit code based on that information.

The default mode of operation is to print the following facts about the current user's default proxy: subject, issuer, identity, type, strength, path, and time left. If the command-line option `-exists` or `-e` is included in the command-line, nothing is printed unless one of the print options is specified. Instead, **grid-proxy-info** determines if a valid proxy exists and, if so, exits with the exit code 0; if a proxy does not exist or is not valid, **grid-proxy-info** exits with the exit code 1. Additional validity criteria can be added by using the `-valid`, `-v`, `-hours`, `-h`, `-bits`, or `-b` command-line options. If used, these options must occur *after* the `-e` or `-exists` command-line options. Those options are only valid if one of the `-e` or `-exists` command-line options is used.

The complete set of command-line options to **grid-proxy-info** are:

<code>-help, -usage</code>	Display the command-line options to <b>grid-proxy-info</b> .
<code>-version</code>	Display the version number of the <b>grid-proxy-info</b> command
<code>-debug</code>	Display verbose error messages.
<code>-file PROXYFILE, -f PROXYFILE</code>	Read the proxy located in the file <i>PROXYFILE</i> instead of using the default proxy.
<code>-subject, -s</code>	Display the proxy certificate's subject distinguished name.
<code>-issuer, -i</code>	Display the proxy certificate issuer's distinguished name.
<code>-identity</code>	Display the proxy certificate's identity. For non-independent proxies, the identity is the subject of the certificate which issued the first proxy in the proxy chain.
<code>-type</code>	Display the type of proxy certificate. The type string includes the format ("legacy", "draft", or RFC 3280 compliant), identity type ("impersonation" or "independent"), and policy ("limited" or "full"). See <code>grid-proxy-init(1)</code> for information about how to create different types of proxies.
<code>-timeleft</code>	Display the number of seconds remaining until the proxy certificate expires.
<code>-strength</code>	Display the strength (in bits) of the key associated with the proxy certificate.
<code>-all</code>	Display the default information for the proxy when also using the <code>-e</code> or <code>-exists</code> command-line option.

- `-text` Display the proxy certificate contents to standard output, including policy information, issuer, public key, and modulus.
- `-path` Display the path to the file containing the default proxy certificate.
- `-rfc2253` Display distinguished names for the subject, issuer, and identity using the string representation described in RFC 2253, instead of the legacy format.
- `-exists, -e` Perform an existence and validity check for the proxy. If a valid proxy exists and matches the criteria described by other command-line options (if any), exit with 0; otherwise, exit with 1. This option must be before other validity check predicate in the command-line options. If this option is specified, the output of the default facts about the proxy is disabled. Use the `-all` option to have the information displayed as well as the exit code set.
- `-valid HOURS:MINUTES,` Check that the proxy certificate is valid for at least *HOURS* hours and *MINUTES*  
`-v HOURS:MINUTES,` minutes. If it is not, **grid-proxy-info** will exit with exit code 1.  
`-hours HOURS, -h HOURS`
- `-bits BITS, -b BITS` Check that the proxy certificate key strength is at least *BITS* bits.

## Environment Variables

The following environment variables affect the execution of **grid-proxy-info**:

`X509_USER_PROXY` Path to the default user proxy.

## See Also

`grid-proxy-init(1)`, `grid-proxy-destroy(1)`

---

# Name

grid-mapfile-add-entry -- Add an entry to a gridmap file

```
grid-mapfile-add-entry [-help] [-usage] [-version] [-versions]
grid-mapfile-add-entry {-dn DISTINGUISHED-NAME} {-ln LOCAL-NAME... }
[[-d] | [-dryrun]]
[[-mapfile MAPFILE] | [-f MAPFILE]]
```

## Description

The **grid-mapfile-add-entry** program adds a new mapping from an X.509 distinguished name to a local POSIX user name to a gridmap file. Gridmap files are used as a simple authorization method for services such as GRAM5 or GridFTP.

The **grid-mapfile-add-entry** program verifies that the *LOCAL-NAME* is a valid user name on the system on which it was run, and that the mapping between *DISTINGUISHED-NAME* and *LOCAL-NAME* does not already exist in the gridmap file.

By default, **grid-mapfile-add-entry** will modify the gridmap file named by the GRIDMAP environment variable if present, or the file `/etc/grid-security/grid-mapfile` if not. This can be changed by the use of the `-mapfile` or `-f` command-line options.

If the gridmap file does not exist, **grid-mapfile-add-entry** will create it. If it already exists, **grid-mapfile-add-entry** will save the current contents of the file to a new file with the string `.old` appended to the file name.

The full set of command-line options to **grid-mapfile-add-entry** are:

<code>-help, -usage</code>	Display the command-line options to <b>grid-mapfile-add-entry</b> .
<code>-version, -versions</code>	Display the version number of the <b>grid-mapfile-add-entry</b> command. The second form includes more details.
<code>-dn <i>DISTINGUISHED-NAME</i></code>	The X.509 distinguished name to add a mapping for. The name should be in OpenSSL's oneline format.
<code>-ln <i>LOCAL-NAME...</i></code>	The POSIX user name to map the distinguished name to. This name must be a valid username. Add multiple <i>LOCAL-NAME</i> strings after the <code>-ln</code> command-line option. If any of the local names are invalid, no changes will be made to the gridmap file. Note that if multiple occurrences of the <code>-ln</code> command-line option are present, only the the last one will be added.
<code>-d, -dryrun</code>	Verify local names and display diagnostics about what would be added to the gridmap file, but don't actually modify the file.
<code>-mapfile <i>MAPFILE</i>, -f <i>MAPFILE</i></code>	Modify the gridmap file named by <i>MAPFILE</i> instead of the default.

## Examples

Add a mapping between the current user's certificate to the current user id to a gridmap file in `$HOME/.gridmap`:

```
% grid-mapfile-add-entry -f $HOME/.gridmap -dn "`grid-cert-info -subject`" -ln "`id -un`"
Modifying /home/juser/.gridmap ...
```

```
/home/juser/.gridmap does not exist... Attempting to create /home/juser/.gridmap
New entry:
"/DC=org/DC=example/DC=grid/CN=Joe User" juser
(1) entry added
```

Add a mapping between the a distinguished name and multiple local names:

```
% grid-mapfile-add-entry -dn "/DC=org/DC=example/DC=grid/CN=Joe User" juser" local1 local2
Modifying /home/juser/.gridmap ...
/home/juser/.gridmap does not exist... Attempting to create /home/juser/.gridmap
New entry:
"/DC=org/DC=example/DC=grid/CN=Joe User" local1,local2
(1) entry added
```

## Environment Variables

The following environment variables affect the execution of **grid-mapfile-add-entry**:

**GRIDMAP** Path to the default gridmap to modify.

## Files

The following files affect the execution of **grid-mapfile-add-entry**:

<code>/etc/grid-security/grid-</code>	Path to the default gridmap to modify if <b>GRIDMAP</b> environment variable is not
<code>mapfile</code>	set.

## See Also

[grid-mapfile-check-consistency\(8\)](#), [grid-mapfile-delete-entry\(8\)](#)

---

# Name

grid-mapfile-check-consistency -- Add an entry to a grid map file

```
grid-mapfile-check-consistency [-h] [-help] [-usage] [-version]
grid-mapfile-check-consistency [-mapfile MAPFILE] | [-f MAPFILE]
```

# Description

The **grid-mapfile-check-consistency** program performs basic checks for validity of a gridmap file. These checks include checks for existence, duplication of entries, and valid local user names. If the gridmap file is valid, **grid-mapfile-check-consistency** exits with a zero exit code, otherwise it exits with a non-zero exit code. In either case, it displays information about its progress as it parses and validates the gridmap file.

By default, **grid-mapfile-check-consistency** will check the gridmap file named by the GRIDMAP environment variable if present. If that variable is not set, it will check the file `$HOME/.gridmap` for non-root users if present. If that doesn't exist or **grid-mapfile-check-consistency** is run as root, it will then check `/etc/grid-security/grid-mapfile`. This can be changed by the use of the `-mapfile` or `-f` command-line options.

The full set of command-line options to **grid-mapfile-check-consistency** are:

```
-help, -h, -usage    Display the command-line options to grid-mapfile-check-consistency.
-version            Display the version number of the grid-mapfile-check-consistency command.
-mapfile MAPFILE,  Check the gridmap file named by MAPFILE instead of the default.
-f MAPFILE
```

# Examples

Check that the gridmap file in `/etc/grid-security` is valid:

```
% grid-mapfile-check-consistency -f /etc/grid-security/grid-mapfile
Checking /etc/grid-security/grid-mapfile
Verifying grid mapfile existence...OK
Checking for duplicate entries...OK
Checking for valid user names...OK
```

Check a gridmap file that has an invalid local user name:

```
% grid-mapfile-check-consistency -f /etc/grid-security/grid-mapfile
Checking /etc/grid-security/grid-mapfile
Verifying grid mapfile existence...OK
Checking for duplicate entries...OK
ERROR: baduser is not a valid local username
ERROR: Found 1 invalid username(s)
```

# Environment Variables

The following environment variables affect the execution of **grid-mapfile-check-consistency**:

GRIDMAP Path to the default gridmap to check.

## Files

The following files affect the execution of **grid-mapfile-check-consistency**:

<code>\$HOME/.gridmap</code>	Path to the default gridmap to check if the GRIDMAP environment variable is not set for non-root users.
<code>/etc/grid-security/grid-mapfile</code>	Path to the default gridmap to check if GRIDMAP environment variable is not set and the above file does not exist.

## See Also

`grid-mapfile-add-entry(8)`, `grid-mapfile-delete-entry(8)`

---

# Name

grid-mapfile-delete-entry -- Remove entries from a gridmap file

```
grid-mapfile-delete-entry [-help] [-usage] [-version] [-versions]
grid-mapfile-delete-entry {-dn DISTINGUISHED-NAME} {-ln LOCAL-NAME... }
[[-d] | [-dryrun]]
[[-mapfile MAPFILE] | [-f MAPFILE]]
```

## Description

The **grid-mapfile-delete-entry** program deletes mappings from a gridmap file. If both the `-dn` and `-ln` options are specified, **grid-mapfile-delete-entry** removes entries which meet both criteria (remove entries mapping *DISTINGUISHED-NAME* to *LOCAL-NAME* for each *LOCAL-NAME* specified). If only `-dn` or `-ln` is specified *all* entries for that *DISTINGUISHED-NAME* or *LOCAL-NAME* are removed.

By default, **grid-mapfile-delete-entry** will modify the gridmap file named by the `GRIDMAP` environment variable if present, or the file `/etc/grid-security/grid-mapfile` if not. This can be changed by the use of the `-mapfile` or `-f` command-line options.

Prior to modifying a gridmap file, **grid-mapfile-delete-entry** saves its current contents to a file with the string `.old` appended to the original file name.

The full set of command-line options to **grid-mapfile-delete-entry** are:

<code>-help, -usage</code>	Display the command-line options to <b>grid-mapfile-delete-entry</b> .
<code>-version, -versions</code>	Display the version number of the <b>grid-mapfile-delete-entry</b> command. The second form includes more details.
<code>-dn <i>DISTINGUISHED-NAME</i></code>	The X.509 distinguished name to remove from the gridmap file. If the <code>-ln</code> option is not specified, remove all entries for this name; otherwise, remove entries that match both this name and the local name. The name should be in OpenSSL's oneline format.
<code>-ln <i>LOCAL-NAME...</i></code>	The POSIX user name to remove from the gridmap file. Include multiple <i>LOCAL-NAME</i> strings after the <code>-ln</code> command-line option to remove multiple names from the gridmap. If the <code>-dn</code> option is not specified, remove all entries for these names; otherwise, remove entries that match the <i>DISTINGUISHED-NAME</i> and any of the <i>LOCAL-NAME</i> values.
<code>-d, -dryrun</code>	Display diagnostics about what would be removed from the gridmap file, but don't actually modify the file.
<code>-mapfile <i>MAPFILE</i>, -f <i>MAPFILE</i></code>	Modify the gridmap file named by <i>MAPFILE</i> instead of the default.

## Examples

Remove all mappings for a distinguished name:

```
% grid-mapfile-delete-entry "/DC=org/DC=example/DC=grid/CN=Joe User"
Modifying /etc/grid-security/grid-mapfile ...
```

```
Deleting entry: "/DC=org/DC=example/DC=grid/CN=Joe User" juser,juser2
(1) entry deleted
```

Remove the mapping between a distinguished name and a single local username:

```
% grid-mapfile-delete-entry "/DC=org/DC=example/DC=grid/CN=Joe User" -ln juser2
Modifying /etc/grid-security/grid-mapfile ...
Current entry: "/DC=org/DC=example/DC=grid/CN=Joe User" juser
(1) mapping removed: (juser2), (0) not present and ignored
(0) entries deleted
```

## Environment Variables

The following environment variables affect the execution of **grid-mapfile-delete-entry**:

**GRIDMAP** Path to the default gridmap to modify.

## Files

The following files affect the execution of **grid-mapfile-delete-entry**:

`/etc/grid-security/grid-mapfile` Path to the default gridmap to modify if **GRIDMAP** environment variable is not set.

## See Also

[grid-mapfile-add-entry\(8\)](#), [grid-mapfile-check-consistency\(8\)](#)

---

# Chapter 7. Configuring Certificates

This section describes the configuration steps required to:

- determine whether or not to trust certificates issued by a particular *Certificate Authority (CA)*,
- provide appropriate default values for use by the **grid-cert-request** command, which is used to generate certificates,
- request *service certificates*, used by services to authenticate themselves to users, and
- specify identity mapping information.

In general, Globus tools will look for a configuration file in a user-specific location first, and in a system-wide location if no user-specific file was found. The configuration commands described here may be run by administrators to create system-wide defaults and by individuals to override those defaults.

## 1. Configuring Globus to Trust a Particular Certificate Authority

### 1.1. Trusted certificates directory

The Globus tools will trust certificates issued by a CA if (and only if) it can find information about the CA in the trusted certificates directory.

The trusted certificates directory is located as described below and exists either on a per-machine or on a per-installation basis.

X509\_CERT\_DIR is the environment variable used to specify the path to the trusted certificates directory. This directory contains information about which CAs are trusted (including the *CA certificates* themselves) and, in some cases, configuration information used by **grid-cert-request** to formulate certificate requests. The location of the trusted certificates directory is looked for in the following order:

1. value of the X509\_CERT\_DIR environment variable
2. \$HOME/.globus/certificates
3. /etc/grid-security/certificates exists
4. \$GLOBUS\_LOCATION/share/certificates

### 1.2. Trusted certificates files

The following two files must exist in the directory for each trusted CA:

**Table 7.1. CA files**

<code>cert_hash.0</code>	The trusted <i>CA Certificate</i> .
<code>cert_hash.signing_policy</code>	A configuration file defining the distinguished names of certificates signed by the CA.

Non-WS Globus components will honor a certificate only if:

- its CA certificate exists (with the appropriate name) in the *TRUSTED\_CA* directory, and
- the certificate's distinguished name matches the pattern described in the signing policy file.

### 1.3. Hash of the CA certificate

The *cert\_hash* that appears in the file names above is the hash of the CA certificate, which can be found by running the command:

```
$GLOBUS_LOCATION/bin/openssl x509 -hash -noout < ca_certificate
```

### 1.4. Creating a signing policy by hand

Some CAs provide tools to install their CA certificates and signing policy files into the trusted certificates directory. You can, however, create a signing policy file by hand; the signing policy file has the following format:

```
access_id_CA X509 'CA Distinguished Name'
pos_rights globus CA:sign
cond_subjects globus '"Distinguished Name Pattern"'
```

In the above, the *CA Distinguished Name* is the subject name of the CA certificate, and the *Distinguished Name Pattern* is a string used to match the distinguished names of certificates granted by the CA.

Some very simple wildcard matching is done: if the *Distinguished Name Pattern* ends with a '\*', then any distinguished name that matches the part of the CA subject name before the '\*' is considered a match.

Note: the *cond\_subjects* line may contain a space-separated list of distinguished name patterns.

### 1.5. Repository of CAs

A repository of CA certificates that are widely used in academic and research settings can be found [here](#)<sup>1</sup>.

## 2. Configuring Globus to Create Appropriate Certificate Requests

The **`grid-cert-request`** command, which is used to create certificates, uses the following configuration files:

**Table 7.2. Certificate request configuration files**

<code>globus-user-ssl.conf</code>	Defines the distinguished name to use for a user's certificate request. The format is described <a href="#">here</a> <sup>2</sup> .
<code>globus-host-ssl.conf</code>	Defines the distinguished name for a host (or service) certificate request. The format is described <a href="#">here</a> <sup>3</sup> .
<code>grid-security.conf</code>	A base configuration file that contains the name and email address for the CA.
<code>directions</code>	An optional file that may contain directions on using the CA.

<sup>1</sup> <https://www.tacar.org/certs.html>

<sup>2</sup> [http://www.openssl.org/docs/apps/req.html#CONFIGURATION\\_FILE\\_FORMAT](http://www.openssl.org/docs/apps/req.html#CONFIGURATION_FILE_FORMAT)

<sup>3</sup> [http://www.openssl.org/docs/apps/req.html#CONFIGURATION\\_FILE\\_FORMAT](http://www.openssl.org/docs/apps/req.html#CONFIGURATION_FILE_FORMAT)

Many CAs provide tools to install configuration files with the following names in the Trusted Certificates directory:

- `globus-user-ssl.conf.cert_hash`
- `globus-host-ssl.conf.cert_hash`
- `grid_security.conf.cert_hash`
- `directions.cert_hash`

## 2.1. Creating a certificate request for a specific CA

The command:

```
grid-cert-request -ca cert_hash
```

will create a certificate request based on the specified CA's configuration files.

## 2.2. Listing available CAs

The command:

```
grid-cert-request -ca
```

will list the available CAs and let the user choose which one to create a request for.

## 2.3. Specifying a default CA for certificate requests

The default CA is the CA that will be used for certificate requests if **`grid-cert-request`** is invoked without the `-ca` flag.

You can specify a default CA by invoking the **`grid-default-ca`** command (follow the link for examples of using the command).

## 2.4. directions file

The `directions` file may contain specific directions on how to use the CA. There are three types of printed messages:

- *REQUEST HEADER*, printed to a certificate request file,
- *USER INSTRUCTIONS*, printed on the screen when one requests a *user certificate*,
- *NONUSER INSTRUCTIONS*, printed on the screen when one requests a certificate for a service.

Each message is delimited from others with lines `----- BEGIN message type TEXT -----` and `----- END message type TEXT -----`. For example, the `directions` file would contain the following lines:

```
----- BEGIN REQUEST HEADER TEXT -----
```

```
This is a Certificate Request file
```

```
It should be mailed to ${GSI_CA_EMAIL_ADDR}
```

```
----- END REQUEST HEADER TEXT -----
```

If this file does not exist, the default messages are printed.

## 3. Requesting Service Certificates

Different CAs use different mechanisms for issuing end-user certificates; some use mechanisms that are entirely web-based, while others require you to generate a certificate request and send it to the CA. If you need to create a certificate request for a service certificate, you can do so by running:

```
grid-cert-request -host hostname -service service_name
```

where *hostname* is the fully-qualified name of the host on which the service will be running, and *service\_name* is the name of the service. This will create the following three files:

**Table 7.3. Certificate request files**

<code>GRID_SECURITY/service_name/service_namecert.pem</code>	An empty file. When you receive your actual service certificate from your CA, you should place it in this file.
<code>GRID_SECURITY/service_name/service_namecert_request.pem</code>	The certificate request, which you should send to your CA.
<code>GRID_SECURITY/service_name/service_namekey.pem</code>	The <i>private key</i> associated with your certificate request, encrypted with the pass phrase that you entered when prompted by <b>grid-cert-request</b> .

The **grid-cert-request** command recognizes several other useful options; you can list these with:

```
grid-cert-request -help
```

## 4. Configuring Credential Mappings

Several Globus services map certificates to local unix usernames to be used with unix services. The default implementation uses a *gridmap* file to map the distinguished name of the identity of the client's certificate to a local login name. Administrators can modify the contents of the gridmap file to control what certificate identities are allowed to access Globus services, as well as configure, via an environment variable, what gridmap file a particular service uses.

In addition to the identity-based mapping done via the gridmap file, administrators can configure Globus services to use arbitrary mapping functions. These may use other criteria, such as SAML assertions, to map a certificate to a local account, or may map certificates to temporary accounts. Administrators can install different mapping implementations and configure services to use them by creating appropriate configuration files and setting environment variables.

### 4.1. Configuring Identity Mappings Using *gridmap* Files

Gridmap files contain a database of entries mapping distinguished names to local user names. These may be manipulated by using the following tools.

#### 4.1.1. Adding an entry to a gridmap file

To add an entry to the gridmap file, run:

```
$GLOBUS_LOCATION/sbin/grid-mapfile-add-entry \
    -dn "Distinguished Name" \
    -ln local_name
```

### 4.1.2. Deleting an entry from a gridmap file

To delete an entry from the gridmap file, run:

```
$GLOBUS_LOCATION/sbin/grid-mapfile-delete-entry \
    -dn "Distinguished Name" \
    -ln local_name
```

### 4.1.3. Checking consistency of a gridmap file

To check the consistency of the gridmap file, run

```
$GLOBUS_LOCATION/sbin/grid-mapfile-check-consistency
```

### 4.1.4. Configuring per-service gridmap files

To configure a service to use a particular gridmap file, set the GRIDMAP variable in the service's environment to the path of the gridmap file. In this way, you can grant different access rights to different certificate identities on a per-service basis by setting the GRIDMAP variable in different service environments.

You can use tools described above to operate on different gridmap files by either setting the GRIDMAP environment variable prior to invoking them, or by using the `-mapfile` command-line option.

For reference, the GSI C code looks for the gridmap in these locations:

**Table 7.4. Gridmap File Location Algorithm**

Location	notes
GRIDMAP environment variable	
<code>/etc/grid-security/grid-mapfile</code>	Only for services running as root.
<code>HOME.gridmap</code>	Only for services not running as root.

### 4.1.5. Gridmap formats

A gridmap line of the form:

```
"Distinguished Name" local_name
```

maps the distinguished name *Distinguished Name* to the local name *local\_name*.

A gridmap line of the form:

```
"Distinguished Name" local_name1,local_name2
```

maps *Distinguished Name* to both *local\_name1* and *local\_name2*; any number of local user names may occur in the comma-separated local name list.

For more detailed information about the gridmap file see the [file description and grammars](https://dev.globus.org/wiki/Gridmap)<sup>4</sup> on dev.globus.org.

---

<sup>4</sup> <https://dev.globus.org/wiki/Gridmap>

## 4.2. Configuring Alternate Credential Mappings

To use an alternative credential mapping, you create a `gsi-authz.conf` file containing information about how the mapping functions are called from the authorization library.

To configure a per-service authorization configuration file, set the `GSI_AUTHZ_CONF` variable to the path to the configuration file in the environment of the service.

For reference, the GSI C code looks for the authorization configuration file in these locations (in the given order):

**Table 7.5. Authorization Configuration File Locations**

Location
<code>GSI_AUTHZ_CONF</code> environment variable
<code>/etc/grid-security/gsi-authz.conf</code>
<code>GLOBUS_LOCATION/etc/gsi-authz.conf</code>
<code>HOME/.gsi-authz.conf</code>

### 4.2.1. Callout File Format

The authorization file defines a set of callouts, one per line. Each callout is defined by an *abstract type*, *library*, and *symbol* separated by whitespace. Comments begin with the `#` character and continue to the end of line.

**Table 7.6. Authorization Configuration File Locations**

Field	Meaning
<i>abstract type</i>	Type of the callout: <i>globus_mapping</i> is used for credential mapping callouts
<i>library</i>	Path to the shared object containing the callout implementation. The library name may be a literal filename, or a partial filename to which the compilation flavor of the service is appended to the filename before its extension.
<i>symbol</i>	The exported symbol containing the entry point to the callout implementation.

Here is a sample `gsi-authz.conf` file that configures a *globus\_mapping* callout to use the *globus\_gridmap\_callout* function in the `/usr/local/globus/lib/libglobus_gridmap_callout_gcc32dbg` shared object:

```
# abstract-type      library                                     symbol
globus_mapping      /opt/globus/lib/libglobus_gridmap_callout_gcc32dbg globus_gridmap_call
```

## 5. GSI File Permissions Requirements

- End Entity Certificate (User, Host and Service) Certificates and the GSI Authorization Callout Configuration File:
  - May not be executable
  - May not be writable by group and other
  - Must be either regular files or soft links
- Private Keys and Proxy Credentials:

- Must be owned by the current (effective) user
- May not be executable
- May not be readable by group and other
- May not be writable by group and other
- Must be either regular files or soft links
- CA Certificates, CA Signing Policy Files, the Grid Map File and the GAA Configuration File:
  - Must be either regular files or soft links
- GSI Authorization callout configuration files
  - Must exist
  - Should be world readable
  - Should not be writable by group and other
  - Should be either a regular file or a soft link
- GSI GAA configuration files
  - Must exist
  - Should be world readable
  - Should not be writable by group and other
  - Should be either a regular file or a soft link

---

# Chapter 8. Environment variable interface

## 1. Environmental Variables for GSI C

### 1.1. Credentials

Credentials are looked for in the following order:

1. service credential
2. host credential
3. proxy credential
4. user credential

X509\_USER\_PROXY specifies the path to the *proxy credential*. If X509\_USER\_PROXY is not set, the proxy credential is created (by **grid-proxy-init**) and searched for (by client programs) in an operating-system-dependent local temporary file.

X509\_USER\_CERT and X509\_USER\_KEY specify the path to the end entity (user, service, or host) certificate and corresponding *private key*. The paths to the certificate and key files are determined as follows:

For *service credentials*:

1. If X509\_USER\_CERT and X509\_USER\_KEY exist and contain a valid certificate and key, those files are used.
2. Otherwise, if the files `/etc/grid-security/service/servicecert` and `/etc/grid-security/service/servicekey` exist and contain a valid certificate and key, those files are used.
3. Otherwise, if the files `$GLOBUS_LOCATION/etc/grid-security/service/servicecert` and `$GLOBUS_LOCATION/etc/grid-security/service/servicekey` exist and contain a valid certificate and key, those files are used.
4. Otherwise, if the files `service/servicecert` and `service/servicekey` in the user's `.globus` directory exist and contain a valid certificate and key, those files are used.

For *host credentials*:

1. If X509\_USER\_CERT and X509\_USER\_KEY exist and contain a valid certificate and key, those files are used.
2. Otherwise, if the files `/etc/grid-security/hostcert.pem` and `/etc/grid-security/hostkey.pem` exist and contain a valid certificate and key, those files are used.
3. Otherwise, if the files `$GLOBUS_LOCATION/etc/hostcert.pem` and `$GLOBUS_LOCATION/etc/hostkey.pem` exist and contain a valid certificate and key, those files are used.
4. Otherwise, if the files `hostcert.pem` and `hostkey.pem` in the user's `.globus` directory, exist and contain a valid certificate and key, those files are used.

For *user credentials*:

1. If `X509_USER_CERT` and `X509_USER_KEY` exist and contain a valid certificate and key, those files are used.
2. Otherwise, if the files `usercert.pem` and `userkey.pem` exist in the user's `.globus` directory, those files are used.
3. Otherwise, if a PKCS-12 file called `usercred.p12` exists in the user's `.globus` directory, the certificate and key are read from that file.

## 1.2. Gridmap file

GRIDMAP specifies the path to the *grid map file*, which is used to map distinguished names (found in certificates) to local names (such as login accounts). The location of the grid map file is determined as follows:

1. If the GRIDMAP environment variable is set, the grid map file location is the value of that environment variable.
2. Otherwise:
  - If the user is root (uid 0), then the grid map file is `/etc/grid-security/grid-mapfile`.
  - Otherwise, the grid map file is `$HOME/.gridmap`.

## 1.3. Trusted CAs directory

`X509_CERT_DIR` is used to specify the path to the trusted certificates directory. This directory contains information about which CAs are trusted (including the *CA certificates* themselves) and, in some cases, configuration information used by **grid-cert-request** to formulate certificate requests. The location of the trusted certificates directory is determined as follows:

1. If the `X509_CERT_DIR` environment variable is set, the trusted certificates directory is the value of that environment variable.
2. Otherwise, if `$HOME/.globus/certificates` exists, that directory is the trusted certificates directory.
3. Otherwise, if `/etc/grid-security/certificates` exists, that directory is the trusted certificates directory.
4. Finally, if `$GLOBUS_LOCATION/share/certificates` exists, then it is the trusted certificates directory.

## 1.4. GSI authorization callout configuration file

`GSI_AUTHZ_CONF` is used to specify the path to the *GSI authorization callout configuration file*. This file is used to configure authorization callouts used by both the gridmap and the authorization API. The location of the GSI authorization callout configuration file is determined as follows:

1. If the `GSI_AUTHZ_CONF` environment variable is set, the authorization callout configuration file location is the value of this environment variable.
2. Otherwise, if `/etc/grid-security/gsi-authz.conf` exists, then this file is used.
3. Otherwise, if `$GLOBUS_LOCATION/etc/gsi-authz.conf` exists, then this file is used.
4. Finally, if `$HOME/.gsi-authz.conf` exists, then this file is used.

## 1.5. GAA (Generic Authorization and Access control) configuration file

GSI\_GAA\_CONF is used to specify the path to the GSI *GAA (Generic Authorization and Access control) configuration file*. This file is used to configure policy language specific plugins to the GAA-API. The location of the GSI GAA configuration file is determined as follows:

1. If the GSI\_GAA\_CONF environment variable is set, the GAA configuration file location is the value of this environment variable.
2. Otherwise, if `/etc/grid-security/gsi-gaa.conf` exists, then this file is used.
3. Otherwise, if `$GLOBUS_LOCATION/etc/gsi-gaa.conf` exists, then this file is used.
4. Finally, if `$HOME/.gsi-gaa.conf` exists, then this file is used.

## 1.6. Grid security directory

GRID\_SECURITY\_DIR specifies a path to a directory containing configuration files that specify default values to be placed in certificate requests. This environment variable is used only by the **grid-cert-request** and **grid-default-ca** commands.

The location of the *grid security directory* is determined as follows:

1. If the GRID\_SECURITY\_DIR environment variable is set, the grid security directory is the value of that environment variable.
2. If the configuration files exist in `/etc/grid-security`, the grid security directory is that directory.
3. If the configuration files exist in `$GLOBUS_LOCATION/etc`, the grid security directory is that directory.

## 1.7. Using TLS

GLOBUS\_GSSAPI\_FORCE\_TLS specifies whether to use TLS by default when establishing a security context. The default behavior if this is not set is to use SSLv3.

## 1.8. Name Comparisons

GLOBUS\_GSSAPI\_NAME\_COMPATIBILITY specifies what name matching algorithms are supported by GSSAPI for mutual authentication and `gss_compare_name`. This variable may be set to any of the following values:

STRICT_GT2	Strictly backward-compatible with GT 2.0 name matching. X.509 subjectAltName values are ignored. Names with hyphens are treated as wildcarded as described in the <a href="#">security considerations</a> documentation. Name matching will rely on canonical host name associated with connection IP addresses.
STRICT_RFC2818	Support <a href="#">RFC 2818</a> <sup>1</sup> server identity processing. Hyphen characters are treated as normal part of a host name. DNSName and IPAddress subjectAltName extensions are matched against the host and port passed to GSSAPI. If subjectAltName is present, X.509 SubjectName is ignored.

<sup>1</sup> <http://www.ietf.org/rfc/rfc2818.txt>

HYBRID	Support a hybrid of the two previous name matching algorithms, liberally matching both hyphen wildcards, canonical names associated with IP addresses, and subjectAlt-Name extensions.
--------	--

If this variable is not set, the HYBRID behavior is used.

---

# Chapter 9. Debugging

For information about system administrator logs, see [Chapter 4, Debugging](#) in the GSI C Admin Guide.

---

# Chapter 10. Troubleshooting

For a list of common errors in GT, see [Error Codes](#).

## 1. Credential Troubleshooting

### 1.1. Credential Errors

The following are some common problems that may cause clients or servers to report that credentials are invalid:

For a list of common errors in GT, see [Error Codes](#).

**Table 10.1. Credential Errors**

<b>Error Code</b>	<b>Definition</b>	<b>Possible Solutions</b>
Your proxy credential may have expired	Your proxy credential may have expired.	Use <code>grid-proxy-info</code> to check whether the proxy credential has actually expired. If it has, generate a new proxy with <code>grid-proxy-init</code> .
The system clock on either the local or remote system is wrong.	This may cause the server or client to conclude that a credential has expired.	Check the system clocks on the local and remote system.
Your end-user certificate may have expired	Your end-user certificate may have expired	Use <code>grid-cert-info</code> to check your certificate's expiration date. If it has expired, follow your CA's procedures to get a new one.
The permissions may be wrong on your proxy file	If the permissions on your proxy file are too lax (for example, if others can read your proxy file), Globus Toolkit clients will not use that file to authenticate.	You can "fix" this problem by changing the permissions on the file or by destroying it (with <code>grid-proxy-destroy</code> ) and creating a new one (with <code>grid-proxy-init</code> ).  <b>Important:</b> However, it is still possible that someone else has made a copy of that file during the time that the permissions were wrong. In that case, they will be able to impersonate you until the proxy file expires or your permissions or end-user certificate are revoked, whichever happens first.
The permissions may be wrong on your private key file	If the permissions on your end user certificate private key file are too lax (for example, if others can read the file), <code>grid-proxy-init</code> will refuse to create a proxy certificate.	You can "fix" this by changing the permissions on the private key file.  <b>Important:</b> However, you will still have a much more serious problem: it is possible that someone has made a copy of your private key file. Although this file is encrypted, it is possible that someone will be able to decrypt the private key, at which point they will be able to impersonate you as long as your end user certificate is valid. You should contact your CA to have your end-user certificate revoked and get a new one.
The remote system may not trust your CA	The remote system may not trust your CA	Verify that the remote system is configured to trust the CA that issued your end-entity certificate. See <a href="#">Installing GT 5.0.2</a> for details.
You may not trust the remote system's CA	You may not trust the remote system's CA	Verify that your system is configured to trust the remote CA (or that your environment is set up to trust the remote CA). See <a href="#">Installing GT 5.0.2</a> for details.
There may be something wrong with the remote service's credentials	There may be something wrong with the remote service's credentials	It is sometimes difficult to distinguish between errors reported by the remote service regarding your credentials and errors reported by the client interface regarding the remote service's credentials. If you cannot find anything wrong with your credentials, check for the same conditions on the remote system (or ask a remote administrator to do so).

## 1.2. Some tools to validate certificate setup

### 1.2.1. grid-cert-diagnostics

The **grid-cert-diagnostics** program checks prints diagnostics about the user's certificates, and host security environment.

```
% grid-cert-diagnostics -p
```

### 1.2.2. Check that the user certificate is valid

```
openssl verify -CApath /etc/grid-security/certificates
-purpose sslclient ~/.globus/usercert.pem
```

### 1.2.3. Connect to the server using s\_client

```
openssl s_client -ssl3 -cert ~/.globus/usercert.pem -key
~/.globus/userkey.pem -CApath /etc/grid-security/certificates
-connect <host:port>
```

Here *<host:port>* denotes the server and port you connect to.

If it prints an error and puts you back at the command prompt, then it typically means that the *server* has closed the connection, i.e. that the server was not happy with the client's certificate and verification. Check the SSL log on the server.

If the command "hangs" then it has actually opened a telnet style (but secure) socket, and you can "talk" to the server.

You should be able to scroll up and see the subject names of the server's verification chain:

```
depth=2 /DC=net/DC=ES/O=ESnet/OU=Certificate Authorities/CN=ESnet Root CA 1
verify return:1
depth=1 /DC=org/DC=DOEGrids/OU=Certificate Authorities/CN=DOEGrids CA 1
verify return:1
depth=0 /DC=org/DC=doegrids/OU=Services/CN=wiggum.mcs.anl.gov
verify return:1
```

In this case, there were no errors. Errors would give you an extra line next to the subject name of the certificate that caused the error.

### 1.2.4. Check that the server certificate is valid

Requires root login on server:

```
openssl verify -CApath /etc/grid-security/certificates -purpose sslserver
/etc/grid-security/hostcert.pem
```

## 2. Grid map Troubleshooting

### 2.1. Grid map errors

The following are some common problems that may cause clients or servers to report that user are not authorized:

For a list of common errors in GT, see [Error Codes](#).

**Table 10.2. Gridmap Errors**

Error Code	Definition	Possible Solutions
The content of the grid map file does not conform to the expected format	The content of the grid map file does not conform to the expected format	Run <a href="#">grid-mapfile-check-consistency</a> to make sure that your gridmap file conforms to the expected format.
The grid map file does not contain a entry for your DN	The grid map file does not contain a entry for your DN	Use <a href="#">grid-mapfile-add-entry</a> to add the relevant entry.

---

# Chapter 11. Related Documentation

- [RFC 3820](#)<sup>1</sup> Proxy Certificates
- [RFC 2744](#)<sup>2</sup> GSSAPI: C-bindings
- [RFC 2743](#)<sup>3</sup> GSSAPI
- [GSSAPI Extensions](#)<sup>4</sup>
- [RFC 2246](#)<sup>5</sup> TLS
- [Grid Security Infrastructure Message Specification](#)<sup>6</sup>

---

<sup>1</sup> <http://www.faqs.org/rfcs/rfc3820.html>

<sup>2</sup> <http://www.faqs.org/rfcs/rfc2744.html>

<sup>3</sup> <http://www.faqs.org/rfcs/rfc2743.html>

<sup>4</sup> <http://www.ggf.org/documents/GWD-I-E/GFD-E.024.pdf>

<sup>5</sup> <http://www.faqs.org/rfcs/rfc2246.html>

<sup>6</sup> <http://www.globus.org/toolkit/docs/3.0/gsi/GSI-message-specification-02.doc>

---

# 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 <code>/etc/grid-security/certificates/&lt;hash&gt;.0</code> , where <code>&lt;hash&gt;</code> is the hash code of the CA identity.
CA Signing Policy	The CA signing policy is used to place constraints on the information you trust a given CA to bind to public keys. Specifically it constrains the identities a CA is trusted to assert in a certificate. In GSI the signing policy for a given CA can typically be found in <code>/etc/grid-security/certificates/&lt;hash&gt;.signing_policy</code> , where <code>&lt;hash&gt;</code> is the hash code of the CA identity.

## E

End Entity Certificate (EEC)	A certificate belonging to a non-CA entity, e.g. you, me or the computer on your desk.
------------------------------	--

## G

GAA configuration file	A file that configures the Generic Authorization and Access control GAA libraries. When using GSI, this file is typically found in <code>/etc/grid-security/gsi-gaa.conf</code> .
grid map file	A file containing entries mapping certificate subjects to local user names. This file can also serve as a access control list for GSI enabled services and is typically found in <code>/etc/grid-security/grid-mapfile</code> . For more information see the Gridmap section <a href="#">here</a> .
grid security directory	The directory containing GSI configuration files such as the GSI authorization callout configuration and GAA configuration files. Typically this directory is <code>/etc/grid-security</code> . For more information see <a href="#">this</a> .
GSI authorization callout configuration file	A file that configures authorization callouts to be used for mapping and authorization in GSI enabled services. When using GSI this file is typically found in <code>/etc/grid-security/gsi-authz.conf</code> .

## H

host certificate	An EEC belonging to a host. When using GSI this certificate is typically stored in <code>/etc/grid-security/hostcert.pem</code> . For more information on possible host certificate locations see the <a href="#">GSI C Developer's Guide</a> .
host credentials	The combination of a host certificate and its corresponding private key.

## P

**private key** The private part of a key pair. Depending on the type of certificate the key corresponds to it may typically be found in `$HOME/.globus/userkey.pem` (for user certificates), `/etc/grid-security/hostkey.pem` (for host certificates) or `/etc/grid-security/<service>/<service>key.pem` (for service certificates).

For more information on possible private key locations see [this](#).

**proxy certificate** A short lived certificate issued using a EEC. A proxy certificate typically has the same effective subject as the EEC that issued it and can thus be used in its place. GSI uses proxy certificates for single sign on and delegation of rights to other entities.

For more information about types of proxy certificates and their compatibility in different versions of GT, see <http://dev.globus.org/wiki/Security/ProxyCertTypes>.

**proxy credentials** The combination of a proxy certificate and its corresponding private key. GSI typically stores proxy credentials in `/tmp/x509up_u<uid>`, where `<uid>` is the user id of the proxy owner.

## S

**service certificate** A EEC for a specific service (e.g. FTP or LDAP). When using GSI this certificate is typically stored in `/etc/grid-security/<service>/<service>cert.pem`. For more information on possible service certificate locations, see [this](#).

**service credentials** The combination of a service certificate and its corresponding private key.

## U

**user certificate** A EEC belonging to a user. When using GSI, this certificate is typically stored in `$HOME/.globus/usercert.pem`. For more information on possible user certificate locations, see [this](#).

**user credentials** The combination of a user certificate and its corresponding private key.