

Installing and Managing Certificates

Installing and Managing Certificates

After installing and configuring either the EGRID Live CD as a UI or EGRID Ready UI you need to install your certificates and create a proxy certificate to access any grid resources. The certificate should be issued by a Certification Authority (CA) trusted by the Grid system that you are accessing. CAs² are organised geographically and by research institute. Each CA has its own procedure to release certificates. The CERN web site [2.6]² maintains an updated list of recognised CAs², as well as detailed information on how to release and install certificates of a particular CA.

Typically, you will get the following two files from the CA:

```
userkey.pem
    contains the private key associated with the certificate
usercert.pem
    the certificate when sent by the CA
```

Installing the Certificate

It is necessary to create a directory named `.globus` inside user's home directory and place the user certificate and keys file there, named `usercert.pem` and `userkey.pem` respectively, with the right permissions.

Step 1

Make the `.globus` directory and copy certificates into it:

```
$ mkdir ~/.globus
$ cp usercert.pem ~/.globus
$ cp userkey.pem ~/.globus
```

Step 2

Then you need to set the proper access rights. The `userkey.pem` file should only be read by the user (keep it very private!), while `usercert.pem` file should be readable by everyone. None of the files need any executable or write privileges.

```
$ cd ~/.globus $ chmod 444 usercert.pem $ chmod 400 userkey.pem
```

After setting permissions, listing the `.globus` directory should result in output similar to this one:

```
$ ls -l
total 12
-r--r--r-- 1 doe doe 5432 2006-07-13 13:14 usercert.pem
-r----- 1 doe doe 1751 2006-07-13 13:14 userkey.pem
```

Note

Note:

The `userkey.pem` file *must* have permissions `=<600`, or else most grid software will not work and the error message is not likely to direct you to the actual problem.

Checking a Certificate

After installing the certificate you can verify that the certificate is not corrupted and information is readable.

The Globus command `grid-cert-info` can be used to list information about the certificate:

```
$ grid-cert-info
```

If the certificate is properly formed, the output will be something like:

Certificate:

Data:

```
Version: 3 (0x2)
Serial Number: 48 (0x30)
Signature Algorithm: sha1WithRSAEncryption
Issuer: O=GRID@Trieste, DC=it, DC=egrid, CN=GRID@Trieste CA 1
Validity
  Not Before: May 26 16:56:04 2006 GMT
  Not After : May 26 16:56:04 2007 GMT
Subject: O=GRID@Trieste, DC=it, DC=sisso, DC=grid, OU=people, CN=Moses Sokunbi
Subject Public Key Info:
  Public Key Algorithm: rsaEncryption
  RSA Public Key: (2048 bit)
    Modulus (2048 bit):
      00:bd:64:fe:fd:db:80:b0:5e:66:7d:81:ff:98:e5:
      59:a8:f2:91:4a:1b:08:36:ce:07:23:ec:60:87:a3:
      28:bd:24:52:53:00:13:67:69:1b:21:62:46:60:a5:
      ba:07:84:be:69:44:63:d0:1c:2d:15:15:87:3d:70:
      b4:4b:77:8b:23:ec:43:87:97:cd:4c:8b:65:4b:bd:
      49:17:cb:cc:11:85:56:ff:e4:f7:c1:47:ad:33:5a:
      59:0b:71:49:61:11:c7:e8:05:33:7e:0f:4f:b9:c6:
      b9:ce:aa:5f:af:87:a7:ab:3d:0b:c8:73:57:ee:ce:
      e9:f2:05:22:78:66:50:d2:cc:1a:c0:90:c1:7c:44:
      13:8a:55:9a:dc:56:c4:4c:c3:a1:1b:30:79:1b:4f:
      9b:dd:2e:54:4a:6d:5a:d8:66:67:69:bf:50:0d:31:
      c3:26:7e:b6:26:58:81:71:f1:7f:00:5d:b6:99:86:
      c9:b7:af:f3:6b:02:44:98:e1:05:ad:bb:a6:55:9e:
      6e:a4:f2:39:21:72:5a:44:51:85:72:c2:e0:6b:28:
      25:f4:c4:10:4a:37:bd:fe:08:36:b7:d1:75:d5:54:
      86:17:17:8b:43:ff:26:1c:67:d5:ba:25:38:76:f7:
      9f:58:2f:f5:31:86:5a:0a:91:ba:98:00:3a:6a:d4:
      d9:6f
    Exponent: 65537 (0x10001)
X509v3 extensions:
  X509v3 Basic Constraints: critical
    CA:FALSE
  X509v3 Key Usage:
    Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment, Key Agreement
  X509v3 Extended Key Usage:
    TLS Web Client Authentication, E-mail Protection, Time Stamping
  X509v3 Issuer Alternative Name:
    email:gridats-ca@egrid.it, URI:http://www.egrid.it/gridats/ca/
  X509v3 Subject Key Identifier:
    92:A0:6C:B6:7E:76:72:A7:4E:33:F9:F2:A5:A2:4C:BB:20:69:FA:41
  X509v3 Authority Key Identifier:
    keyid:D9:52:B7:93:AB:CA:95:15:03:AE:B0:69:61:AB:57:36:EA:CF:58:29
    DirName:/O=GRID@Trieste/DC=it/DC=egrid/CN=GRID@Trieste CA 1
    serial:A3:1F:65:F8:F4:AB:59:69

  X509v3 Subject Alternative Name:
    email:moses@democritos.it
Signature Algorithm: sha1WithRSAEncryption
```

```

92:a5:82:eb:56:bf:86:3b:39:a3:d6:f3:52:5d:d5:be:fe:d5:
a1:0f:78:bf:0a:ad:22:15:c5:24:a2:8c:2a:bb:22:1c:e5:6f:
25:ba:e0:d2:9e:0e:e5:04:9b:8e:e5:d7:ed:ba:70:aa:64:86:
cc:52:15:9a:db:ec:2b:dd:2e:7e:6a:08:e7:4b:f1:58:67:19:
cd:e5:ef:83:05:99:c2:44:b2:f8:07:bc:c3:cd:03:cd:0b:6a:
0f:9c:d5:f2:38:93:58:79:53:2b:61:ba:a3:65:ad:7b:f2:10:
f1:f7:09:07:4d:ae:d0:2c:17:7c:4f:02:d2:a0:df:66:ee:ea:
8a:12:e8:dd:29:ed:13:1a:b3:26:28:a6:de:cd:10:15:8b:53:
03:08:28:57:c4:9f:40:c9:85:16:19:94:94:16:a2:7a:e4:54:
48:6f:84:f2:f0:a2:b3:90:e1:5d:f1:24:b8:8b:12:75:76:95:
14:a9:9c:7d:10:7b:b7:bd:e3:36:9d:ac:ae:2d:38:d7:0e:42:
d5:51:ef:ac:0d:cd:4b:04:36:b7:b7:39:79:7b:43:d7:51:b6:
60:5d:4c:dd:a7:98:d5:9f:0a:ef:cf:2b:f8:13:f4:6c:c8:d4:
1f:28:2c:4f:71:a6:37:b3:81:07:a6:a5:e5:3d:7e:ea:e5:8c:
98:6f:63:98

```

The `grid-cert-info` command takes many options. The `-help` option lists all the options supported by the command:

```
$ grid-cert-info -help
```

```
grid-cert-info [-help] [-file certfile] [-all] [-subject] [...]
```

Displays certificate information. Unless the optional `-file` argument is given, the default location of the file containing the certificate is assumed:

```
-- The location pointed to by the .
-- If X509_USER_CERT not set, /home/rmurri/.globus/usercert.pem.
```

Several options can be given: The output of
`"grid-cert-info -subject -issuer"`
is equivalent to that of
`"grid-cert-info -subject ; grid-cert-info -issuer"`

Options

```
-help, -usage           Display usage
-version                Display version
-file certfile          | -f      Use 'certfile' at non-default location
```

Options determining what to print from certificate

```
-all                    Whole certificate
-subject                | -s      Subject string of the cert
-issuer                 | -i      Issuer
-startdate              | -sd     Validity of cert: start date
-enddate                | -ed     Validity of cert: end date
```

In particular, the `-subject` option returns the subject of the certificate:

```
$ grid-cert-info -subject
/O=GRID@Trieste/DC=it/DC=sissa/DC=grid/OU=people/CN=Moses Sokunbi
```

The `openssl x509` command can be used instead to verify the validity of a certificate with respect to the certificate of the CA that issued it. To verify a user certificate, just issue the following command from the UI:

```
$ openssl verify \
  -CApath /etc/grid-security/certificates \
  ~/.globus/usercert.pem
```

If the certificate is valid, the output will be like this:

```
/home/moses/.globus/usercert.pem: OK
```

If the certificate of the CA that issued the user certificate is not found in `-CApath2`, an error message like the following will appear:

```
usercert.pem: /O=GRID/O=GRID@Trieste/OU=people/CN=Moses Sokunbi  
error 20 at 0 depth lookup:unable to get local issuer certificate
```

Proxy Certificates

A proxy certificate is a delegated user credential that authenticates the user in every secure interaction, and has a limited lifetime. It prevents having to use user's own certificate, which could compromise its safety. When `userkey.pem` and `usercert.pem` files are copied to the `~/ .globus` directory and access privileges are set the user is in a position to generate a proxy certificate.

The `grid-proxy-init` command is used to generate proxy certificates. However in order to generate a proxy certificate, the user must provide the password to get the private key out of the `userkey.pem` file.

Creating a Proxy Certificate

To create a proxy certificate, issue the command:

```
$ grid-proxy-init
```

```
Your identity: /O=GRID@Trieste/DC=it/DC=sisssa/DC=grid/OU=people/CN=Moses Sukumbi  
Enter GRID pass phrase for this identity:
```

```
Creating proxy.....Done  
Your proxy is valid until: Fri Sep 1 23:01:57 2006
```

If the user gives a wrong pass phrase, the output will be:

```
ERROR: Couldn't read user key. This is likely caused by either  
giving the wrong passphrase or bad file permissions
```

```
key file location:  
    /home/moses/.globus/userkey.pem
```

Use `-debug` for further information.

The proxy certificate will be saved in the file `/tmp/x509up_u<uid>`, where `<uid>` is the Unix UID of the user. If the environment variable `X509_USER_PROXY` is defined (e.g. `X509_USER_PROXY=$HOME/.globus/proxy`), a proxy with that file name will be created, if possible.

If the proxy certificate file cannot be created, the output will be:

```
ERROR: The proxy credential could not be written to the output file.
```

Use `-debug` for further information.

If the user certificate files are missing, or the permissions of `userkey.pem` are not correct, the output will be:

```
ERROR: Couldn't find valid credentials to generate a proxy.
```

Use `-debug` for further information.

By default, the proxy has a lifetime of 12 hours. To specify a different lifetime, the `-valid H:M` option can be used (the proxy is valid for H hours and M minutes). When a proxy certificate has expired, it becomes useless and a new one has to be created with `grid-proxy-init` command. It is not advisable to create longer lifetime proxies because they imply bigger security risks.

The `-help` option provides a full list of options supported by the command:

```
$ grid-proxy-init -help
```

```
Syntax: grid-proxy-init-bin [-help][--pwstdin][--limited][--valid H:M] ...
```

Options	
<code>-help, -usage</code>	Displays usage
<code>-version</code>	Displays version
<code>-debug</code>	Enables extra debug output
<code>-q</code>	Quiet mode, minimal output
<code>-verify</code>	Verifies certificate to make proxy for
<code>--pwstdin</code>	Allows passphrase from stdin
<code>--limited</code>	Creates a limited globus proxy
<code>--independent</code>	Creates a independent globus proxy
<code>--old</code>	Creates a legacy globus proxy
<code>--valid <h:m></code>	Proxy is valid for h hours and m minutes (default:12:00)
<code>--hours <hours></code>	Deprecated support of hours option
<code>--bits <bits></code>	Number of bits in key {512 1024 2048 4096}
<code>--policy <policyfile></code>	File containing policy to store in the ProxyCertInfo extension
<code>--pl <oid>, --policy-language <oid></code>	OID string for the policy language used in the policy file
<code>--path-length <l></code>	Allow a chain of at most l proxies to be generated from this one
<code>--cert <certfile></code>	Non-standard location of user certificate
<code>--key <keyfile></code>	Non-standard location of user key
<code>--certdir <certdir></code>	Non-standard location of trusted cert dir
<code>--out <proxyfile></code>	Non-standard location of new proxy cert

After creating a certificate it is also possible to print information about it.

Getting information about the proxy certificate

To print information about a proxy certificate, for example, the subject or the time left before expiration, use the command:

```
$ grid-proxy-info
```

The output, if a valid proxy exists, will be similar to:

```
subject   : /O=GRID@Trieste/DC=it/DC=sissa/DC=grid/OU=people/CN=Moses Sokunbi/CN=proxy
issuer    : /O=GRID@Trieste/DC=it/DC=sissa/DC=grid/OU=people/CN=Moses Sokunbi
identity  : /O=GRID@Trieste/DC=it/DC=sissa/DC=grid/OU=people/CN=Moses Sokunbi
type      : full legacy globus proxy
strength  : 512 bits
path      : /tmp/x509up_u1000
timeleft  : 11:49:12
```

If a proxy certificate does not exist, the output is:

```
ERROR: Couldn't find a valid proxy.
Use -debug for further information.
```

If the certificate exists but is no longer valid `timeleft` : will be indicated as `00:00:00`.

Destroying a Proxy Certificate

It is also possible to destroy a proxy certificate before its expiration. The `grid-proxy-destroy` command is used to destroy an existing proxy certificate, though proxies already sent with jobs will still be valid when the local copy is destroyed.

Use the following command to destroy the proxy certificate:

```
$ grid-proxy-destroy
```

No output will be shown if the proxy file was successfully removed. If no proxy certificate exists, the result will be:

```
ERROR: Proxy file doesn't exist or has bad permissions
Use -debug for further information.
```

Long-term proxy creation and management

A proxy certificate created as described in the previous section can cause problems, if the job does not finish before the proxy expires, then the job will be aborted. This is clearly a problem if, for example, the user must submit a number of jobs that take a lot of time to finish: he/she should create a proxy certificate with a very long lifetime, which could increase the security risks.

To overcome this limit, a proxy credential repository system is used, which allows the user to create and store a long-term proxy certificate on a dedicated server (called `MyProxy? Server`). The WMS (Workload Management System) will then be able to use this long-term proxy to periodically renew the proxy for a submitted job before it expires and until the job ends (or the long-term proxy expires).

As the renewal process should start some time before the initial proxy expires, it is necessary to generate an initial proxy that is long enough. If the renewal is triggered bit too late job will fail by giving the following error:

```
Status Reason: Got a job held event, reason: Globus error 131: the
user proxy expired (job is still running)
```

The minimum recommended time for the initial proxy is 30 minutes, and the `edg-job-*` commands will not even be accepted if the lifetime of the proxy credentials in the UI is lower than 10 minutes. An error message like the following will be produced:

```
**** Error: UI_PROXY_DURATION ****
Proxy certificate will expire within less then 00:10 hours.
```

The advanced proxy management can be performed using the `myproxy` command suite. The user must know the host name of a `MyProxy? Server`.

For the WMS to know what `MyProxy? Server` to use in the proxy certificate renewal process, the name of the server must be included in an attribute of the job's JDL file. If the user does not add it manually, then the name of the default `MyProxy? Server` is added automatically when the job is submitted. This default

MyProxy? Server node is site- and VO- dependent and is usually defined in the UI VO's configuration file, stored at \$EDG_WL_LOCATION/etc/<vo>/edg_wl_ui.conf.

Creating a long-term proxy

To create and store a long-term proxy certificate, the user must use the `myproxy-init` command which has the following format:

```
myproxy-init -s "host_name" -d -n
```

where `-s "host_name"` specifies the hostname of the machine where a MyProxy? Server runs, the `-d` option instructs the server to use the subject of the certificate as the default username, and the `-n` option avoids the use of a passphrase to access to the long-term proxy, so that the WMS can perform the renewals automatically.

Use the following command:

```
$ myproxy-init -s gridts04.grid.elettra.trieste.it -d -n
```

The output will be similar to:

```
Your identity: /O=GRID@Trieste/DC=it/DC=sissa/DC=grid/OU=people/CN=Moses Sokunbi
Enter GRID pass phrase for this identity:
Creating proxy ..... Done
Proxy Verify OK
Your proxy is valid until: Fri Sep 8 11:15:37 2006
A proxy valid for 168 hours (7.0 days) for user
/O=GRID@Trieste/DC=it/DC=sissa/DC=grid/OU=people/CN=Moses Sokunbi now
exists on gridts04.grid.elettra.trieste.it
```

By default, the long-term proxy lasts for one week and the proxy certificates created from it last 12 hours. These lifetimes can be changed using the `-c` and `-t` option, respectively.

If the `-s "host_name"` option is missing, the command will try to use the `$MYPROXY_SERVER` environment variable to determine the MyProxy? Server.

If the hostname of MyProxy? Server is wrong, or the service is unavailable, the output will be similar to:

```
Your identity: /O=GRID@Trieste/DC=it/DC=sissa/DC=grid/OU=people/CN=Moses Sokunbi
Enter GRID pass phrase for this identity:
Creating proxy ..... Done
Proxy Verify OK
Unknown host "gridts04.elettra.trieste.it"
```

where only the last line reveals that an error occurred.

Retrieving information about a long-term Proxy

After creating a long term proxy, information about it can be retrieved. To get information about a long-term proxy stored in MyProxy? Server, the `myproxy-info` command which has the following format can be used:

```
myproxy-info -s "host_name" -d
```

where the `-s` and `-d` options have the same meaning as in the `myproxy-init` command.

Use the following command to get information about the long term proxy that was just created:

```
$ myproxy-info -s gridts04.grid.elettra.trieste.it -d
```

The output will be similar to the following:

```
username: /O=GRID@Trieste/DC=it/DC=sissa/DC=grid/OU=people/CN=Moses Sokunbi
owner: /O=GRID@Trieste/DC=it/DC=sissa/DC=grid/OU=people/CN=Moses Sokunbi
timeleft: 166:19:21 (6.9 days)
```

Note

Note:

The user must have a valid proxy certificate on the UI, created with `grid-proxy-init` command, to successfully interact with his long-term certificate on MyProxy² server.

Deleting a long-term proxy

When ever the long term proxy is not necessary it can be deleted using the `myproxy-destroy` command which has the following format:

```
myproxy-destroy -s "host_name" -d
```

To delete a long term proxy certificate that was just created se the following command:

```
$ myproxy-destroy -s gridts04.grid.elettra.trieste.it -d
```

The output will be similar to the following:

```
Default MyProxy credential for user /O=GRID@Trieste/DC=it/DC=sissa/DC=grid/OU=people/CN=Moses Sokunbi
successfully removed.
```