Hands-on BEgrid

Bruno Goossens

Rosette Vandenbroucke

Document version: v8

Introduction

This document leads you through a number of exercises covering the major aspects of job submission and data management.
It is assumed that you are familiar with the basic Linux/UNIX user environment (bash, shell etc.)

This document is designed to be accompanied by a series of presentations providing a general overview of Grid and the gLite tools.

Please read carefully the full exercise before starting it.

Glossary

CA	Certificate Authority
CE	Computing Element
CPU	Central Processing Unit
EDG	European DataGrid
GridFTP	Grid version of the File Transport Protocol
GUID	Globally Unique Identifier
JDL	Job Description Language
JSS	Job Submission System
LCG	LHC Computing Grid
LFC	LHC File Catalog
LFN	Logical File Names
LHC	the Large Hadron Collider
LRMS	Local Resource Management System
PBS	Portable Batch System
PFN	Physical File Name
RA	Registration Authority
RB	Resource Broker
RLS	Replica Location Service
RM	Replica Manager
RMC	Replica Metadata Catalogue
RSL	Resource Specification Language
SE	Storage Element
SFN	Site File Name
SRM	Storage Resource Manager
UI	User Interface
VO	Virtual Organization
VOMS	Virtual Organization Management System
WMS	Workload Management System
WN	Worker Node

Authenticating

Exercise 1: Obtaining a BEgrid certificate

Goal: Connect to the “Certificate Authority” (CA) BELNET web site and request a new certificate.

Requisites:

- The user’s email address is mandatory to request a certificate.
- The user must have access to his/her mail.

Important: *Don’t ask for a new certificate if you already own one delivered by BELNET CA.*

When you want to log on a machine, usually you have to contact the site administrator and ask for a username and a password. Users are authenticated and gain authorization to use well identified systems. But the user has to remember as many passwords as there are systems. This cumbersome way of working is not suitable for the Grid, where you will be accessing numerous systems on sites without even noticing it. It is also unmanageable to install username/password on all the systems belonging to the grid. But even these arguments are not the only reason to chose for another system for identification to use the grid, namely security. There is a need for very strong authentication on the grid to reduce hacking risks.

Using Certificates is the mean used within the Grid to ensure secure access. The certificate binds together your identity (name, affiliation, etc.) and a unique piece of digital data called a public key. The use of a public key for authentication is based on a special mathematical trick, called asymmetric cryptography. It consists of two large (prime) numbers which are multiplied, making it virtually impossible to factorize the product into the two numbers again. The individual prime numbers are used to generate an encryption and a decryption function and the product of the two, and then the two numbers are destroyed. If you only have the encryption function, it is impossible to derive the decryption functions from it (and vice versa). So, if you distribute widely the encryption function called public key (e.g. on the web) and keep the decryption function private, everyone can send you encrypted messages that can only be read by you. This system can be used to gain access to a remote system. If that system knows your public key it can encrypt a challenge (e.g. a random number) using this key and ask you to decrypt it. If you can, you obviously own the private key and therefore you are who you say to be. Still the remote site has to know the public keys of all its customers. However authentication of the user has to be done in a trusted way and public keys have to be made available in a manageable and accessible way.

A PKI (Public Key Infrastructure) solves the above mentioned challenges: authentication and public key archiving. The PKI is based on the notion of a trusted third party, a human that can

authenticate people in person. In a PKI there is the notion of a CA (Certification Authority) and a RA (Registration Authority). The CA will sign the certificate while the RA will do the authentication. Like a CA, a RA is a real person, maybe the head of your personnel department or your team leader. The RAs do not sign certificates themselves, but tell a CA that a particular person belongs to a particular certificate and that they should sign the request. The task of an RA is simple, and many RAs can be appointed for one CA.
The RA operator confirms to the Certification Authority (CA) that you are who you claim to be. The CA will then take your public key and your identifier and put those together in a certificate. As a proof of authentication, the CA will then calculate a digest (hash) of the combination of the two and encrypt it with the private key of the CA. Everyone can recalculate the digest, decrypt the signature using the public key of the CA and verify that these two are the same. If you show up at a remote site that only knows your name (identifier) and trust the CA that you got your certificate from, the site knows that whoever can decrypt the challenge sent corresponds to the name they have in their list of allowed users.

BELNET is the CA for Begrid. Several RAs have been accredited. For the moment KULeuven, UA, UGent, ULB, VUB, VITO and VLIZ do directly the authentication of users belonging to their institute. For other users, BELNET acts as the RA.

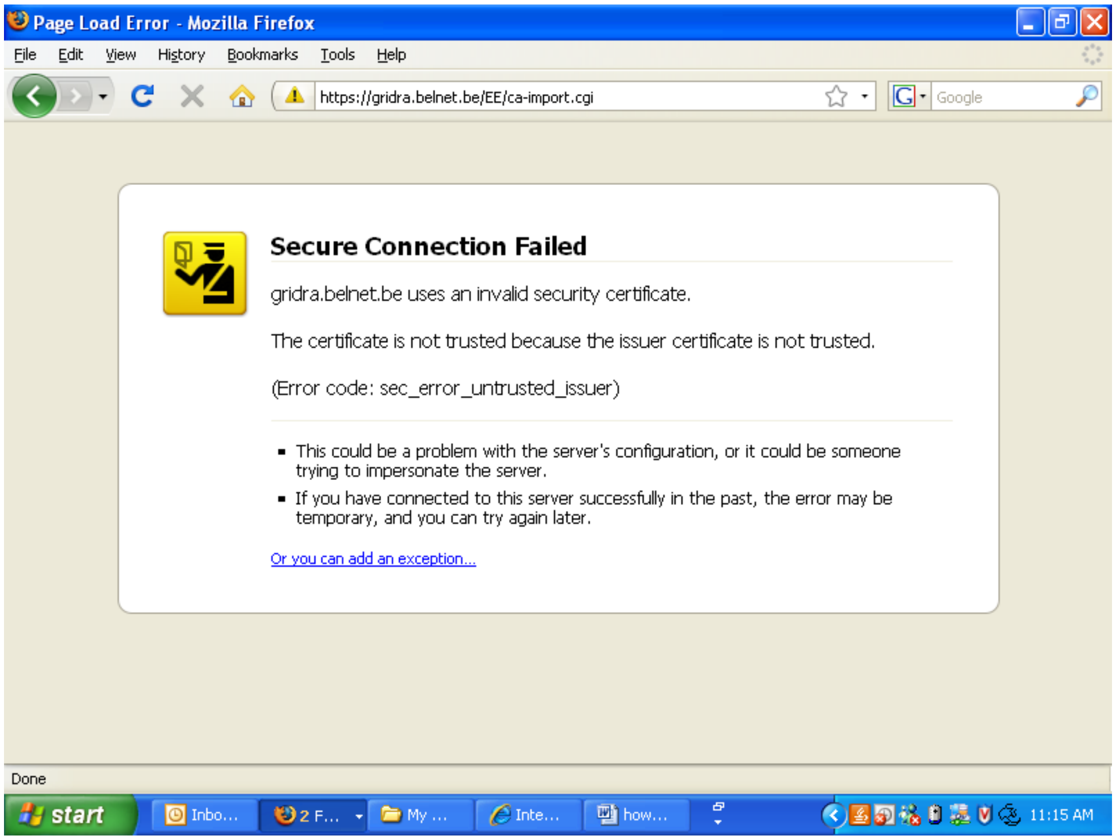
Comments:
1: This exercise of getting a BEGrid certificate is made using Firefox (3.0.4). However the steps are almost the same for all browsers.
2. Requesting and retrieving a certificate must be done on the same PC with the same browser. This is because during the request a key-pair is generated and saved by the browser. Only when you retrieve the certificate with that browser, the key-pair will be available in your certificate.

Practice:

1.1 Get and install the CA certificates with Firefox

1.1.1 Go to the webpage https://gridra.belnet.be/EE/

As a first time user, you receive the following warning because your browser does not contain any (Begrid recognized) certificate:


Page Load Error - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/ca-import.cgi
Google
Done
start
11:15 AM

Secure Connection Failed

gridra.belnet.be uses an invalid security certificate.

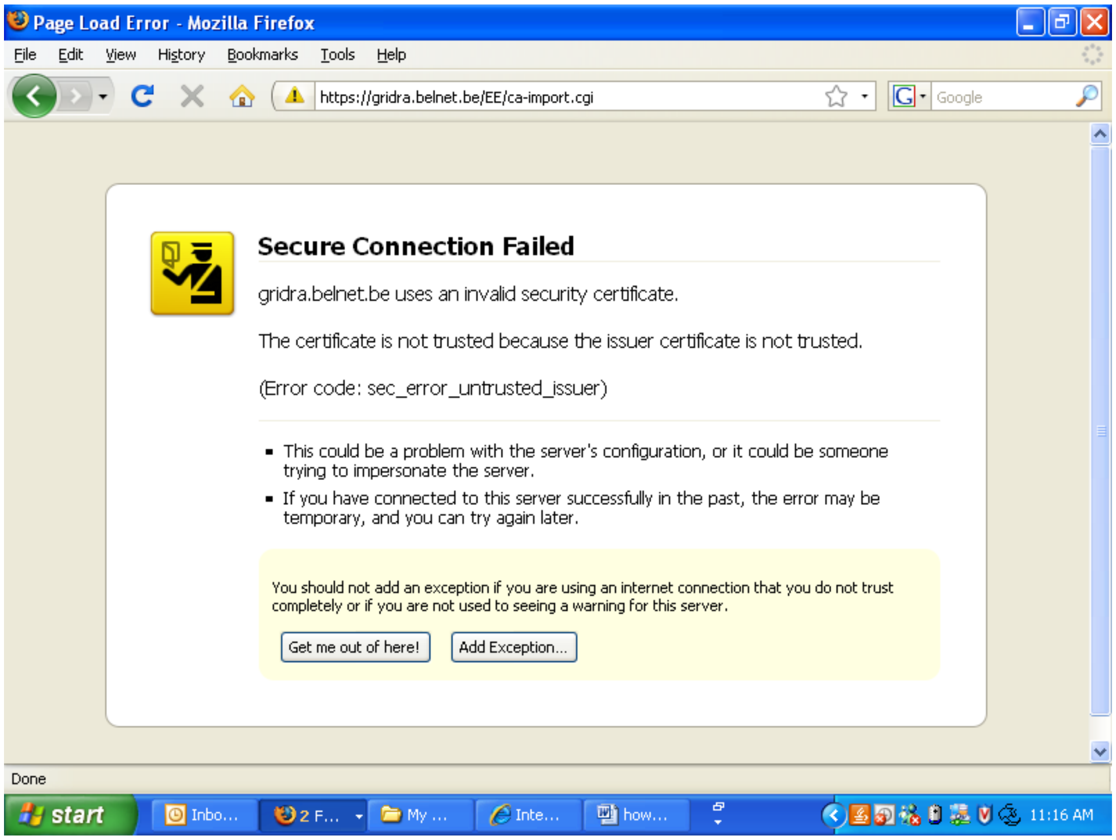
The certificate is not trusted because the issuer certificate is not trusted.

(Error code: sec_error_untrusted_issuer)

- This could be a problem with the server's configuration, or it could be someone trying to impersonate the server.
- If you have connected to this server successfully in the past, the error may be temporary, and you can try again later.

Or you can add an exception…

Click on “Or you can add an exception” and continue:


Page Load Error - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/ca-import.cgi
Google

Secure Connection Failed

gridra.belnet.be uses an invalid security certificate.

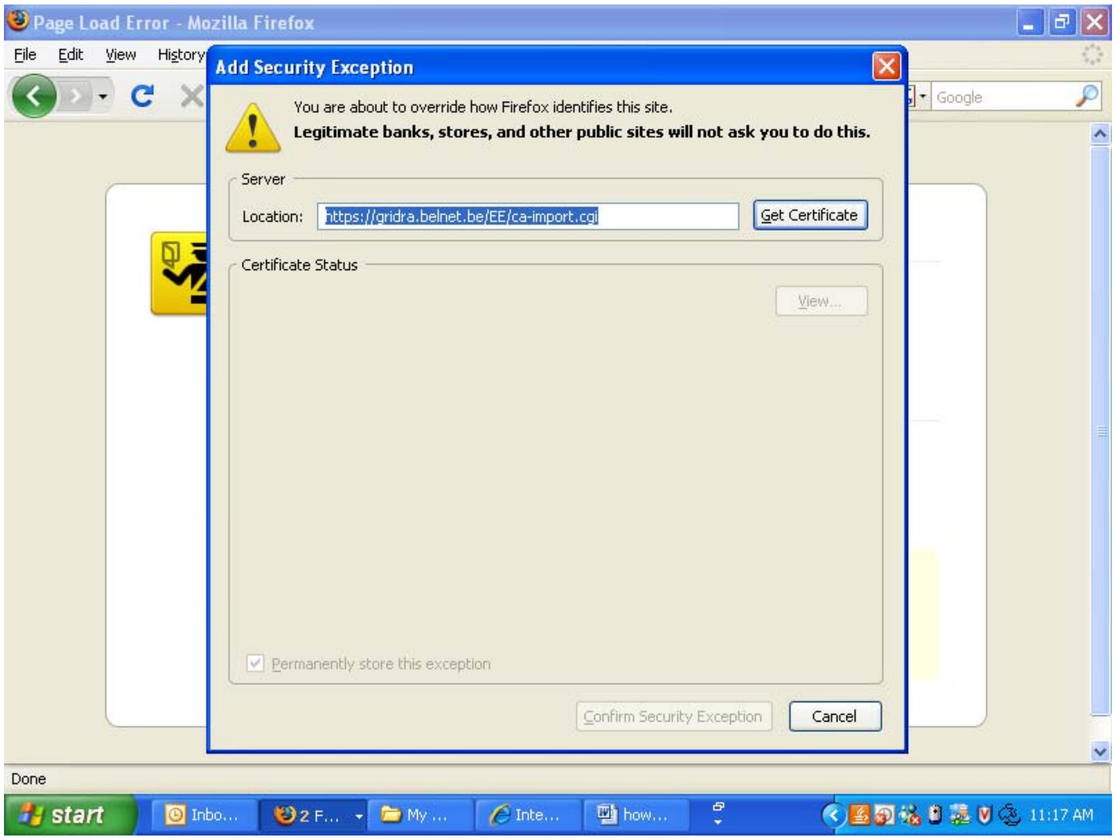
The certificate is not trusted because the issuer certificate is not trusted.

(Error code: sec_error_untrusted_issuer)

- This could be a problem with the server's configuration, or it could be someone trying to impersonate the server.
- If you have connected to this server successfully in the past, the error may be temporary, and you can try again later.

You should not add an exception if you are using an internet connection that you do not trust completely or if you are not used to seeing a warning for this server.

"Add the exception" and continue



Add Security Exception

You are about to override how Firefox identifies this site.
Legitimate banks, stores, and other public sites will not ask you to do this.

Server

Location: https://gridra.belnet.be/EE/ca-import.cgi Get Certificate

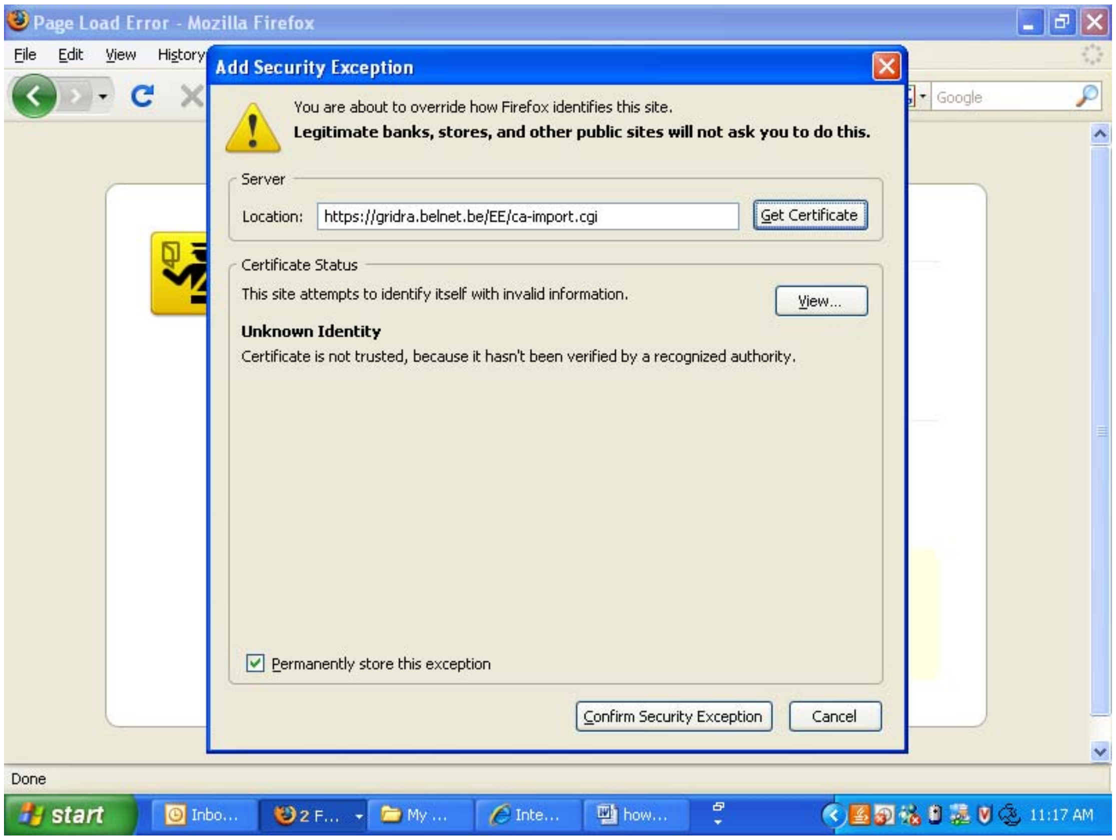
Certificate Status

View...

Permanently store this exception

Confirm Security Exception Cancel

Click on “Get Certificate”



You are about to override how Firefox identifies this site.
Legitimate banks, stores, and other public sites will not ask you to do this.

Server
Location: https://gridra.belnet.be/EE/ca-import.cgi

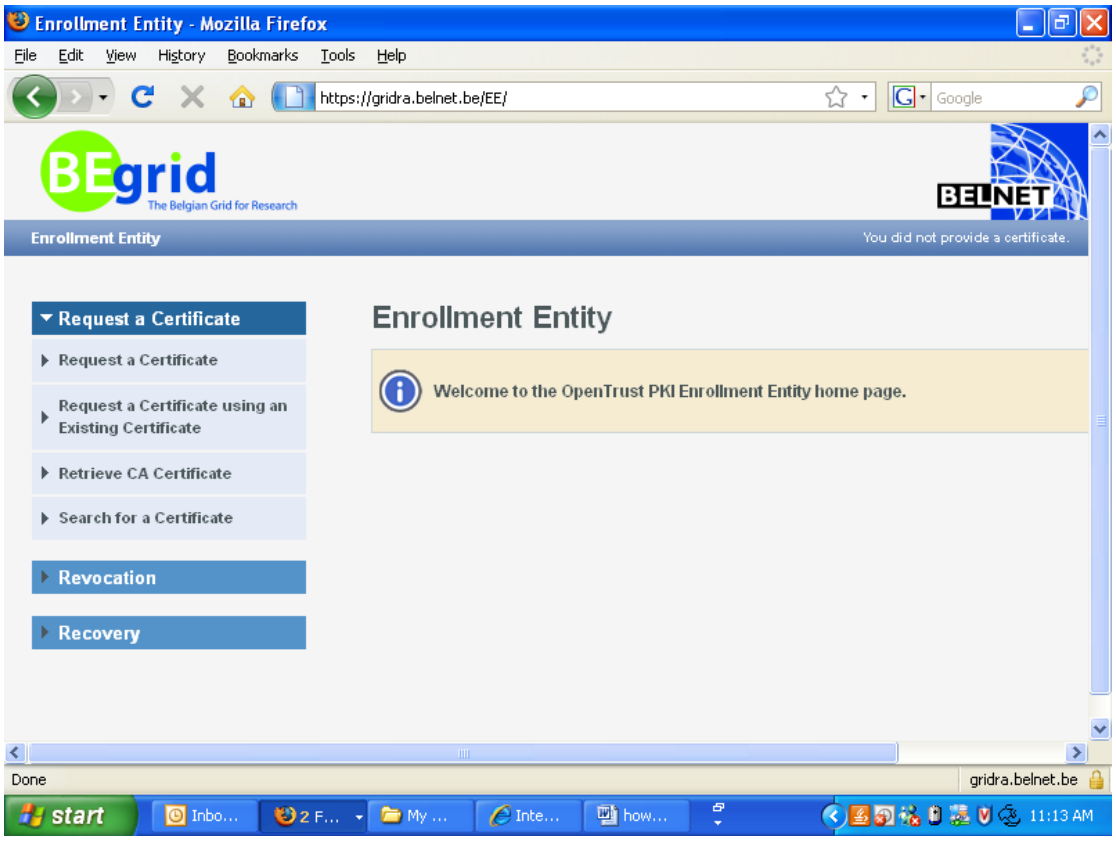
Certificate Status
This site attempts to identify itself with invalid information.

Unknown Identity
Certificate is not trusted, because it hasn't been verified by a recognized authority.

☑ Permanently store this exception

And finally, "Confirm security exception"

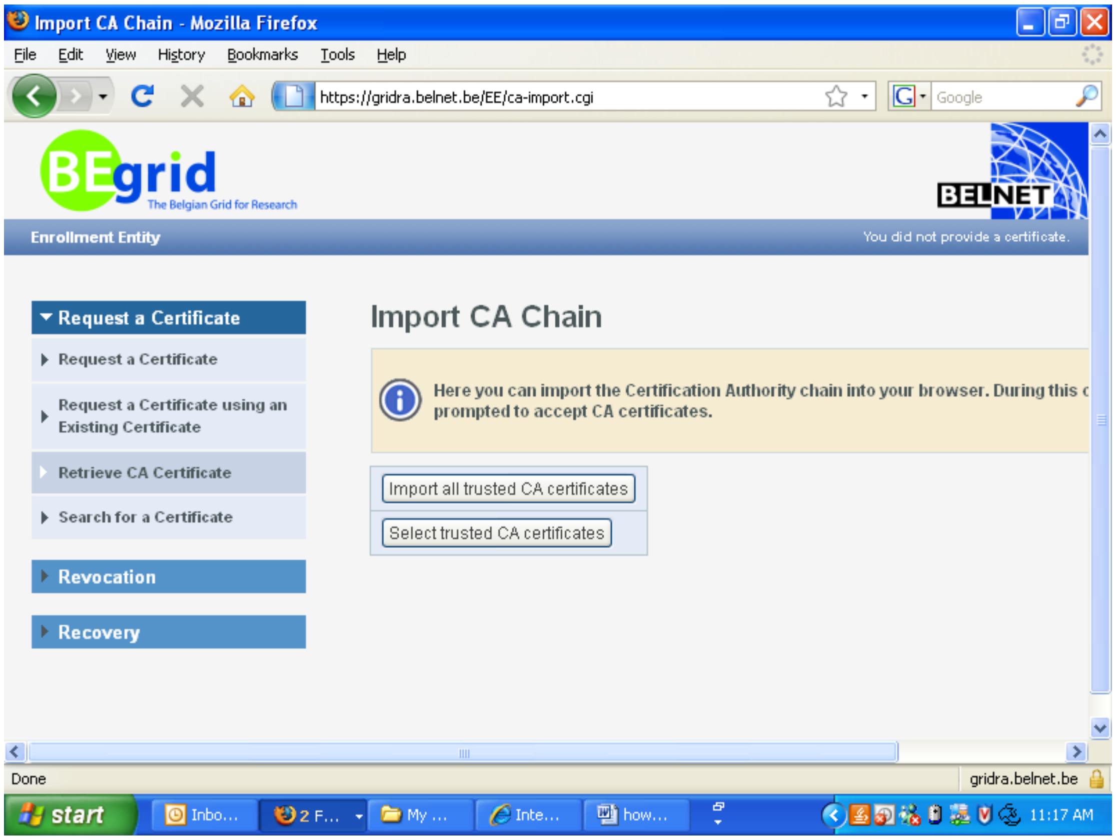
You will get the main page of the Enrollment Entity:


Enrollment Entity - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/
BEgrid The Belgian Grid for Research
BELNET
Enrollment Entity
You did not provide a certificate.
Request a Certificate
Request a Certificate
Request a Certificate using an Existing Certificate
Retrieve CA Certificate
Search for a Certificate
Revocation
Recovery
Enrollment Entity
Welcome to the OpenTrust PKI Enrollment Entity home page.
Done
gridra.belnet.be

1.1.2 Getting the CA certificate in your browser

The BELNET Public Key Infrastructure provides: a Root CA and two specific CAs, one for BEgrid, the BEgrid CA, and the other for other activities, the Technical CA.
For BEgrid, you MUST download the Root CA and the BEgrid CA.

Click on "Retrieve CA certificate"

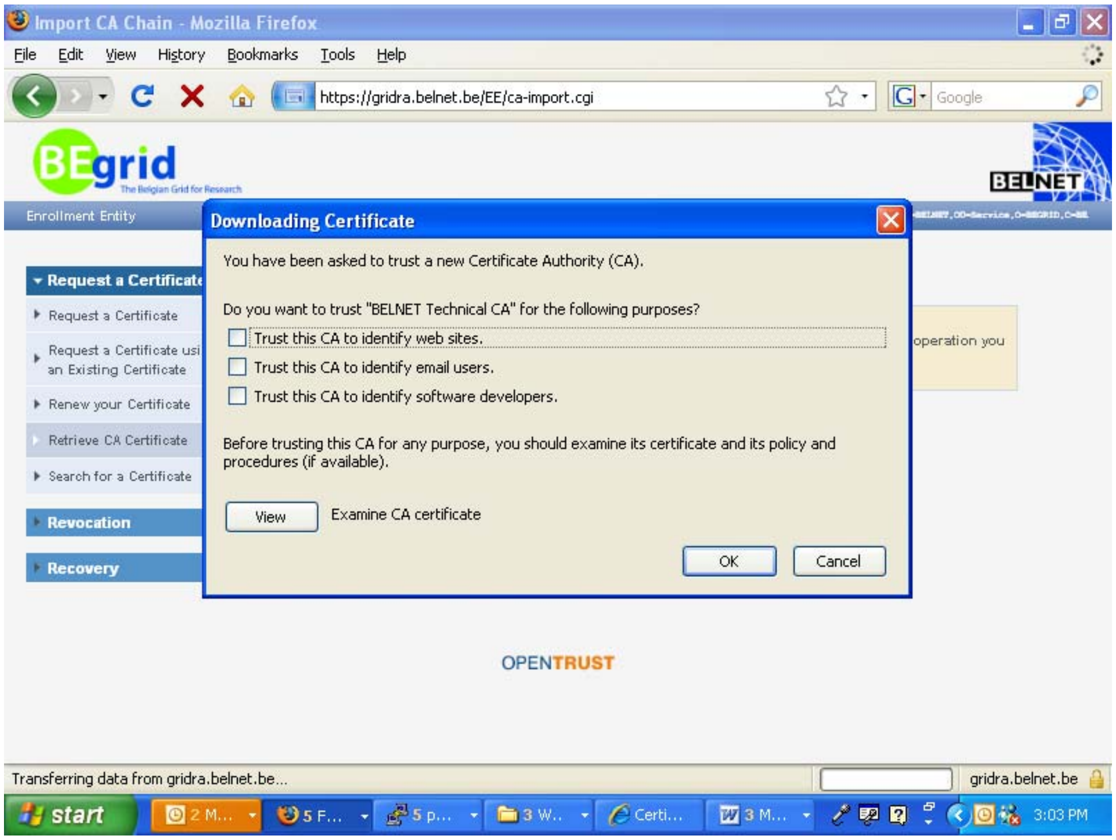

Import CA Chain - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/ca-import.cgi
BEgrid The Belgian Grid for Research
BELNET
Enrollment Entity
You did not provide a certificate.
Request a Certificate
Request a Certificate
Request a Certificate using an Existing Certificate
Retrieve CA Certificate
Search for a Certificate
Revocation
Recovery
Import all trusted CA certificates
Select trusted CA certificates
Done
gridra.belnet.be
start
11:17 AM

Import CA Chain

Here you can import the Certification Authority chain into your browser. During this c
prompted to accept CA certificates.

You can either “Import all trusted CA certificates” or “Select trusted CA certificates”

Importing all trusted CA certificates


Import CA Chain - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/ca-import.cgi
BEgrid The Belgian Grid for Research
BELNET
Enrollment Entity
Request a Certificate
Request a Certificate
Request a Certificate usi an Existing Certificate
Renew your Certificate
Retrieve CA Certificate
Search for a Certificate
Revocation
Recovery
operation you
OPENTRUST
Transferring data from gridra.belnet.be...
gridra.belnet.be
start
3:03 PM

Downloading Certificate

You have been asked to trust a new Certificate Authority (CA).

Do you want to trust "BELNET Technical CA" for the following purposes?

- [ ] Trust this CA to identify web sites.
- [ ] Trust this CA to identify email users.
- [ ] Trust this CA to identify software developers.

Before trusting this CA for any purpose, you should examine its certificate and its policy and procedures (if available).

View Examine CA certificate

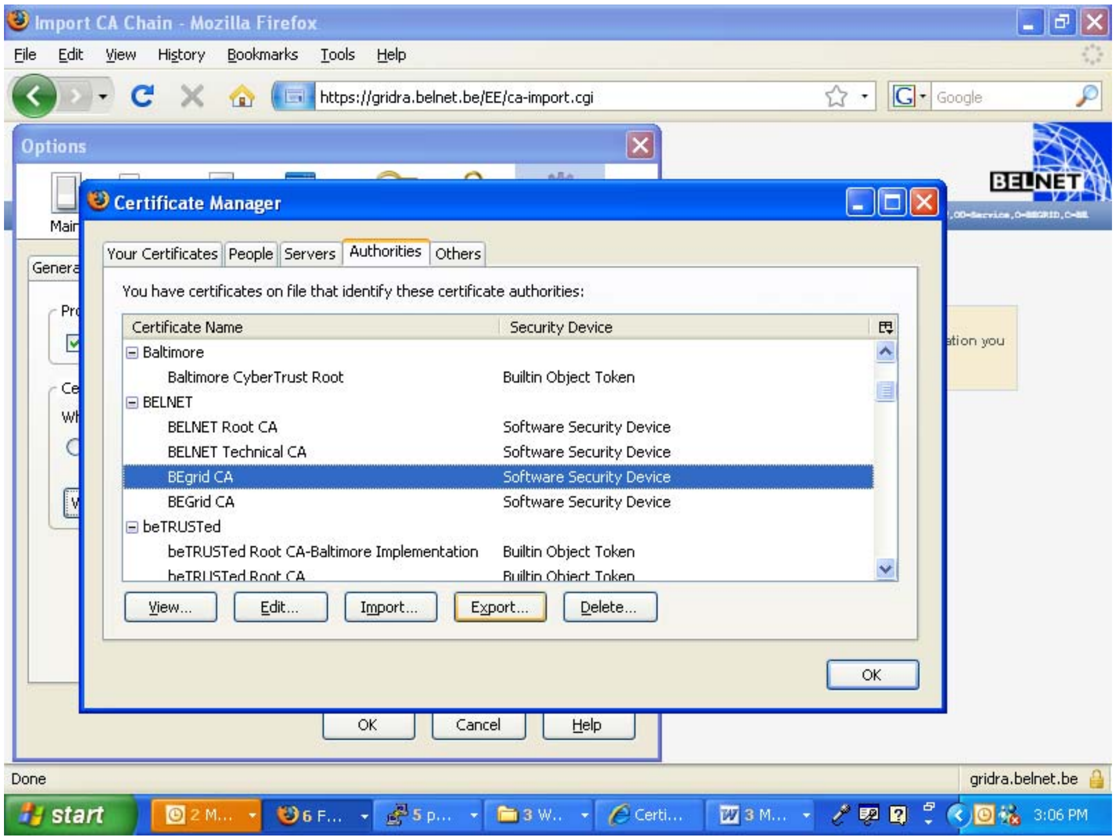
OK Cancel

You are prompted to trust the BEgrid CA, the BELNET Root CA, the BELNET technical CA

– To identify websites
- to identify email users
- to identify software developers

Select those options and click on OK

The chosen CA certificates are installed in your browser followed by a check of their availability by executing the following actions:
Open “Tools”, “Options”, Encryption”, “View certificates”.
On the “Certificate Manager”, open “Authorities” where your newly imported CA certificates are installed.
You receive a window that is comparable with the window below.


Import CA Chain - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/ca-import.cgi
Options
Certificate Manager
Your Certificates People Servers Authorities Others
View... Edit... Import... Export... Delete...
OK
OK Cancel Help
Done
gridra.belnet.be
start
3:06 PM

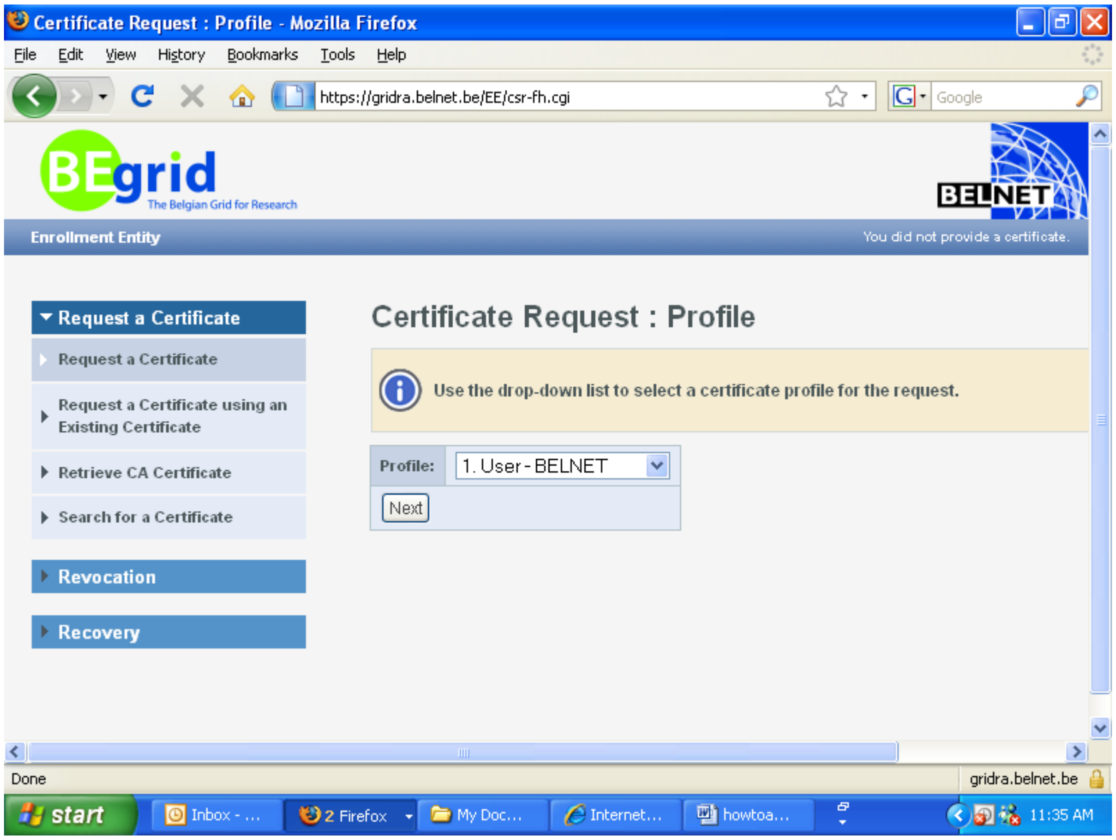
You have certificates on file that identify these certificate authorities:

Certificate Name	Security Device
⊟ Baltimore	
Baltimore CyberTrust Root	Builtin Object Token
⊟ BELNET	
BELNET Root CA	Software Security Device
BELNET Technical CA	Software Security Device
BEgrid CA	Software Security Device
BEGrid CA	Software Security Device
⊟ beTRUSTed	
beTRUSTed Root CA-Baltimore Implementation	Builtin Object Token
beTRUSTed Root CA	Builtin Object Token

Note that you also can use the CA certificates into a file and store them in the directory you indicate.

1.1.3 Getting a user certificate

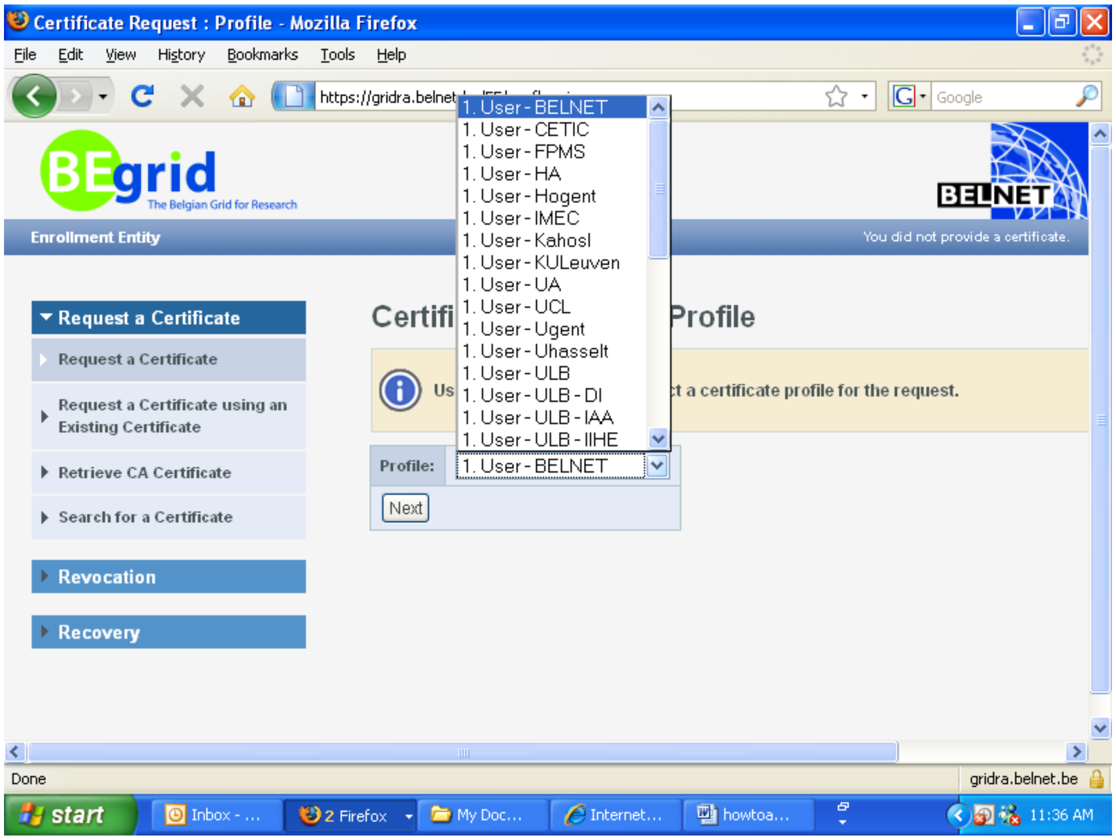
On the BEgrid CA page click on “Request a certificate” and then chose your profile. In the example we make the request as a BELNET user. You should choose “your” organization.
We included most of the institutes/organisations even with departments, that have Begrid users. If yours is not defined then choose BELNET. Afterwards we will add your organisation, you can then revoke your certificate and request a new one with the correct identification.


Certificate Request : Profile - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/csr-fh.cgi
BEgrid The Belgian Grid for Research
BELNET
Enrollment Entity
You did not provide a certificate.
Request a Certificate
Request a Certificate
Request a Certificate using an Existing Certificate
Retrieve CA Certificate
Search for a Certificate
Revocation
Recovery
Next
Done
gridra.belnet.be

Certificate Request : Profile

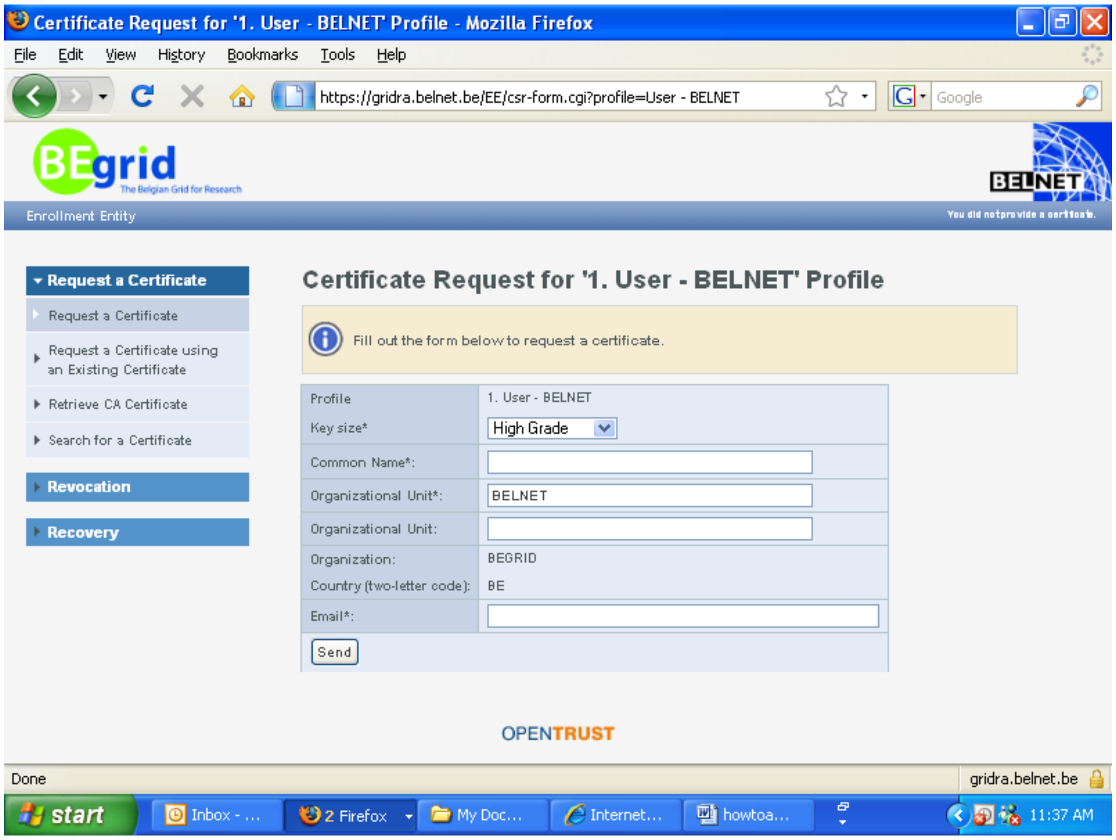
Use the drop-down list to select a certificate profile for the request.

Profile: 1. User - BELNET


Certificate Request : Profile - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet
Google
BEgrid The Belgian Grid for Research
BELNET
Enrollment Entity
You did not provide a certificate.
Request a Certificate
Request a Certificate
Request a Certificate using an Existing Certificate
Retrieve CA Certificate
Search for a Certificate
Revocation
Recovery
Certifi Profile
Us t a certificate profile for the request.
1. User - BELNET
1. User - CETIC
1. User - FPMS
1. User - HA
1. User - Hogent
1. User - IMEC
1. User - Kahosl
1. User - KULeuven
1. User - UA
1. User - UCL
1. User - Ugent
1. User - Uhasselt
1. User - ULB
1. User - ULB - DI
1. User - ULB - IAA
1. User - ULB - IIHE
Profile: 1. User - BELNET
Next
Done
gridra.belnet.be
start Inbox - ... 2 Firefox My Doc... Internet... howtoa... 11:36 AM

Here you have to fill in the form with your personal data:
- Your name (Family and First name),
- Your email (please give your institution email, not your private email)
- Keep the default key size “High Grade”
- And all empty fields (depending of the chosen profile)

Comment: Do not use any special character in your name or organization as the VOMS (see next chapter) will not be able to cope with those characters.


Certificate Request for '1. User - BELNET' Profile - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/csr-form.cgi?profile=User - BELNET
BEgrid The Belgian Grid for Research
BELNET
Enrollment Entity
You did not provide a certificate.
Request a Certificate
Request a Certificate
Request a Certificate using an Existing Certificate
Retrieve CA Certificate
Search for a Certificate
Revocation
Recovery
OPENTRUST
Done
gridra.belnet.be
start
Inbox - ...
2 Firefox
My Doc...
Internet...
howtoa...
11:37 AM

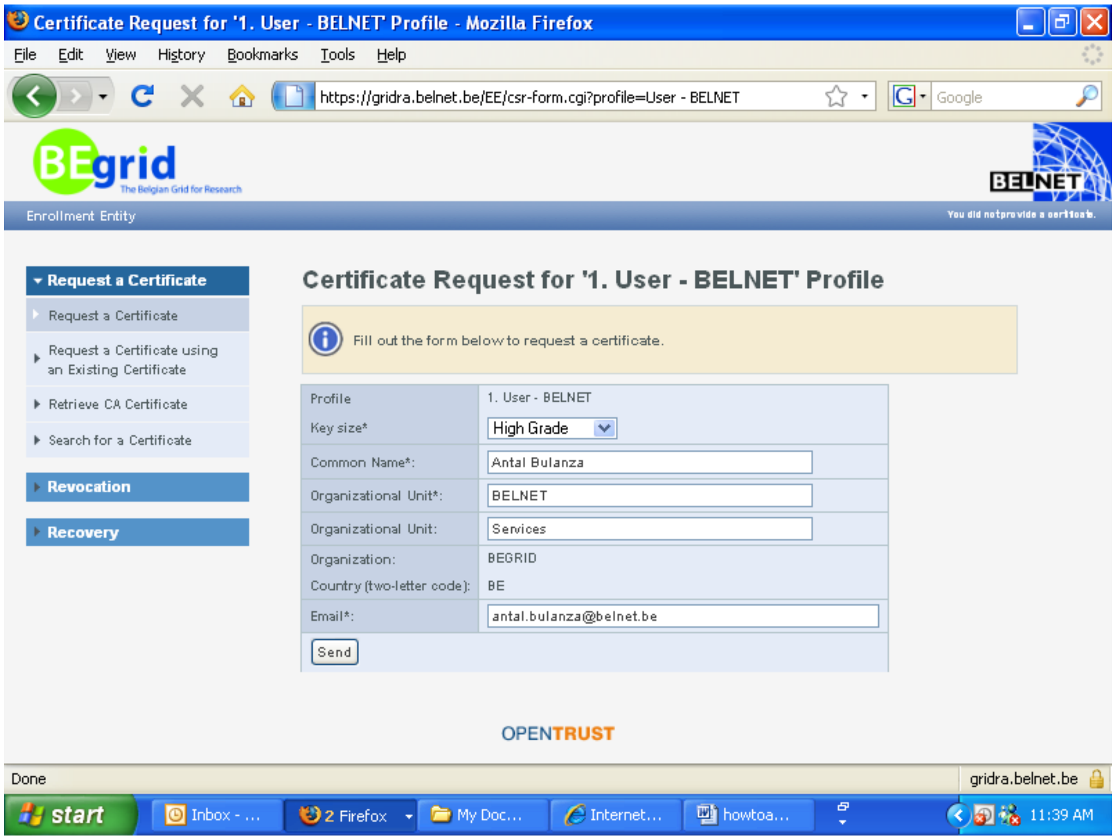
Certificate Request for '1. User - BELNET' Profile

Fill out the form below to request a certificate.

Profile	1. User - BELNET
Key size*	High Grade
Common Name*:	
Organizational Unit*:	BELNET
Organizational Unit:	
Organization:	BEGRID
Country (two-letter code):	BE
Email*:	

Send

Example:

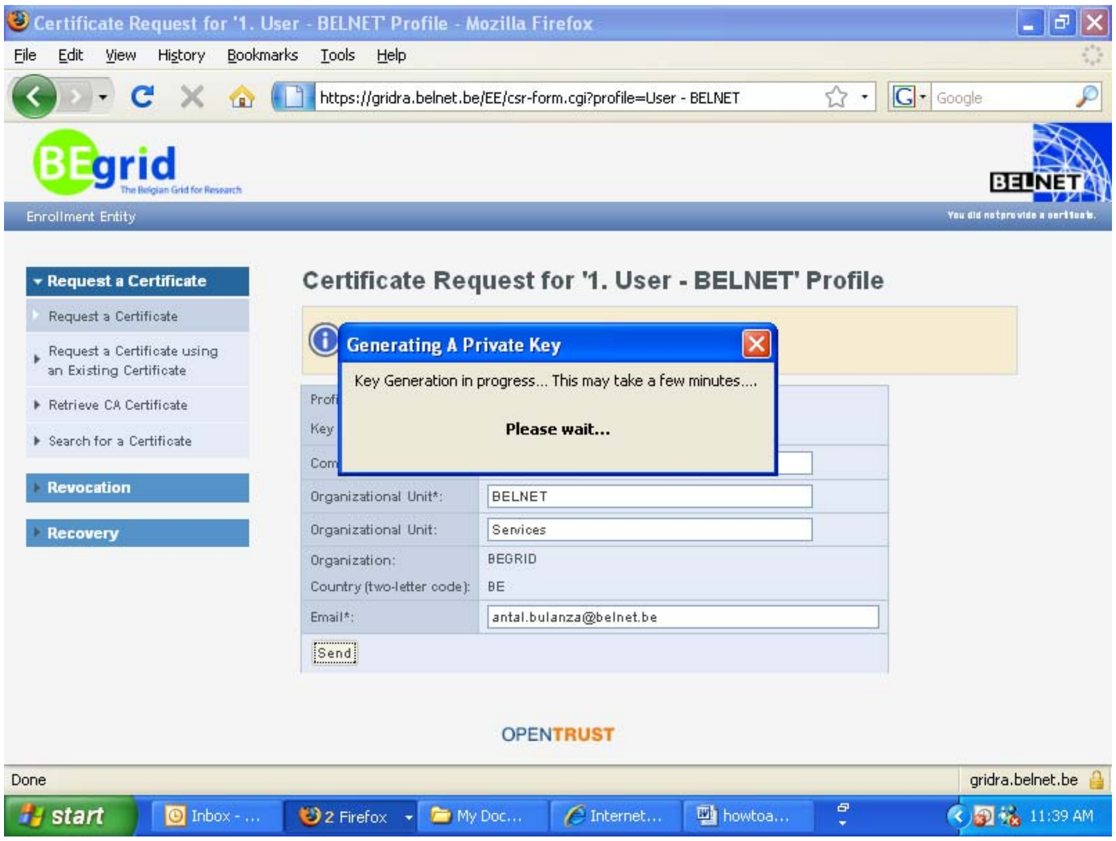

Certificate Request for '1. User - BELNET' Profile - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/csr-form.cgi?profile=User - BELNET
BEgrid The Belgian Grid for Research
BELNET
Enrollment Entity
You did not provide a certificate.
Request a Certificate
Request a Certificate
Request a Certificate using an Existing Certificate
Retrieve CA Certificate
Search for a Certificate
Revocation
Recovery
Send
OPENTRUST
Done
gridra.belnet.be
start
Inbox - ...
2 Firefox
My Doc...
Internet...
howtoa...
11:39 AM

Certificate Request for '1. User - BELNET' Profile

Fill out the form below to request a certificate.

Profile	1. User - BELNET
Key size*	High Grade
Common Name*:	Antal Bulanza
Organizational Unit*:	BELNET
Organizational Unit:	Services
Organization:	BEGRID
Country (two-letter code):	BE
Email*:	antal.bulanza@belnet.be

The key generation process starts when you push send. The generation of the private key may take a few minutes. Then the certificate request is sent to the RA and the user will receive an e-mail when the certificate is ready (or rejected).


Certificate Request for '1. User - BELNET' Profile - Mozilla Firefox
https://gridra.belnet.be/EE/csr-form.cgi?profile=User - BELNET
Enrollment Entity
Request a Certificate
Request a Certificate using an Existing Certificate
Retrieve CA Certificate
Search for a Certificate
Revocation
Recovery
Send
OPENTRUST
Done
gridra.belnet.be

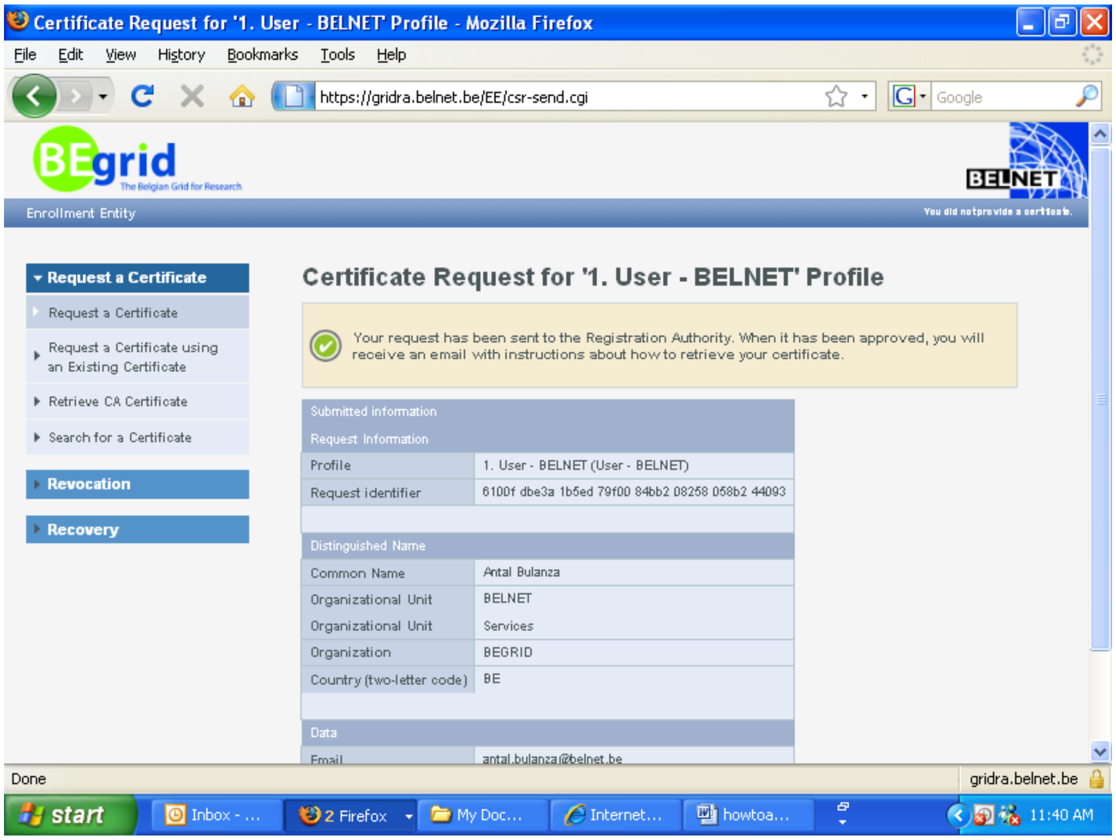
Certificate Request for '1. User - BELNET' Profile

Generating A Private Key

Key Generation in progress... This may take a few minutes....

Please wait...

Organizational Unit*:	BELNET
Organizational Unit:	Services
Organization:	BEGRID
Country (two-letter code):	BE
Email*:	antal.bulanza@belnet.be


Certificate Request for '1. User - BELNET' Profile - Mozilla Firefox
https://gridra.belnet.be/EE/csr-send.cgi
Enrollment Entity
Request a Certificate
Request a Certificate using an Existing Certificate
Retrieve CA Certificate
Search for a Certificate
Revocation
Recovery
Done
gridra.belnet.be

Certificate Request for '1. User - BELNET' Profile

Your request has been sent to the Registration Authority. When it has been approved, you will receive an email with instructions about how to retrieve your certificate.

Submitted information	
Request Information	
Profile	1. User - BELNET (User - BELNET)
Request identifier	6100f dbe3a 1b5ed 79f00 84bb2 08258 058b2 44093
Distinguished Name	
Common Name	Antal Bulanza
Organizational Unit	BELNET
Organizational Unit	Services
Organization	BEGRID
Country (two-letter code)	BE
Data	
Email	antal.bulanza@belnet.be

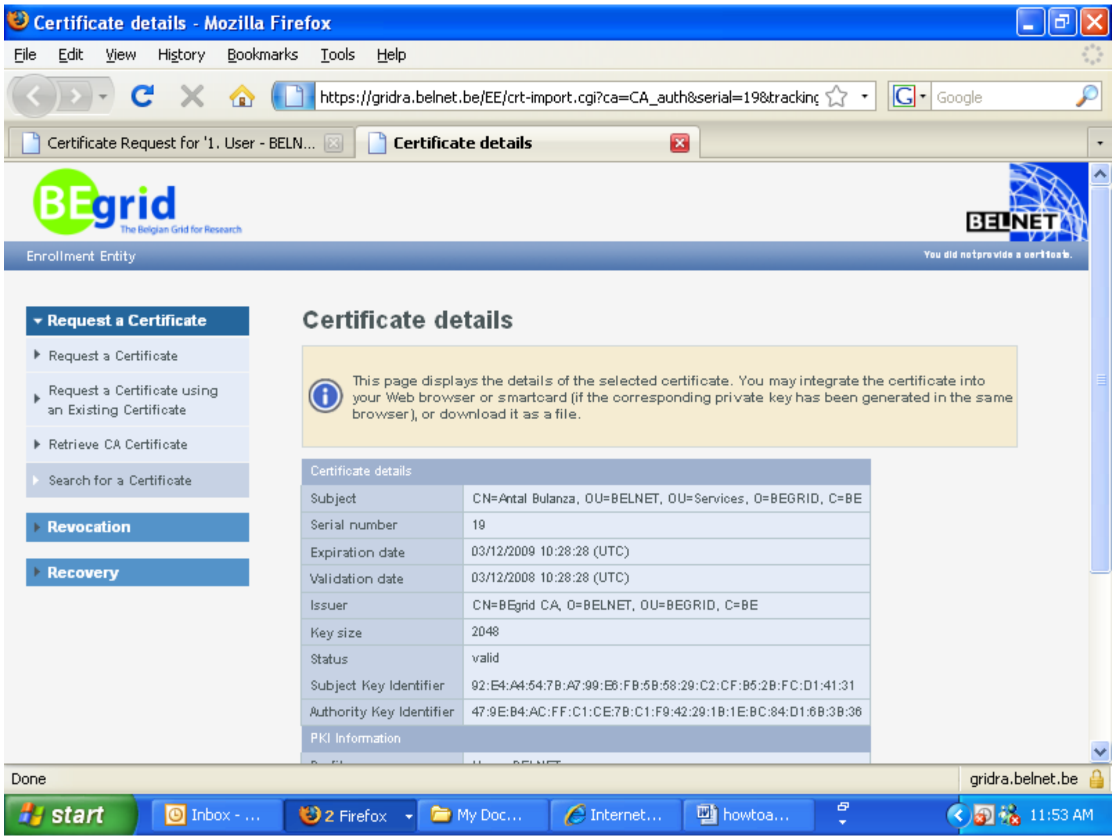
The content of the mail looks like:

Subject: Your certificate is ready
From: jean-christophe.real@belnet.be
Date: Wed, 3 Dec 2008 11:44:03 +0100 (CET)
To: antal.bulanza@belnet.be
Hello,

Your certificate CN=Antal Bulanza,OU=BELNET,OU=Services,O=BEGRID,C=BE has been signed.
You can fetch it at the following URL: https://gridra.belnet.be/EE/crt-import.cgi?ca=CA_auth&serial=19&tracking=6100fdbe3a1b5ed79f0084bb208258058b244093

1.1.4 Fetch the certificate

Use the URLspecified in the e-mail to fetch the certificate:
The page provides the certificate details:

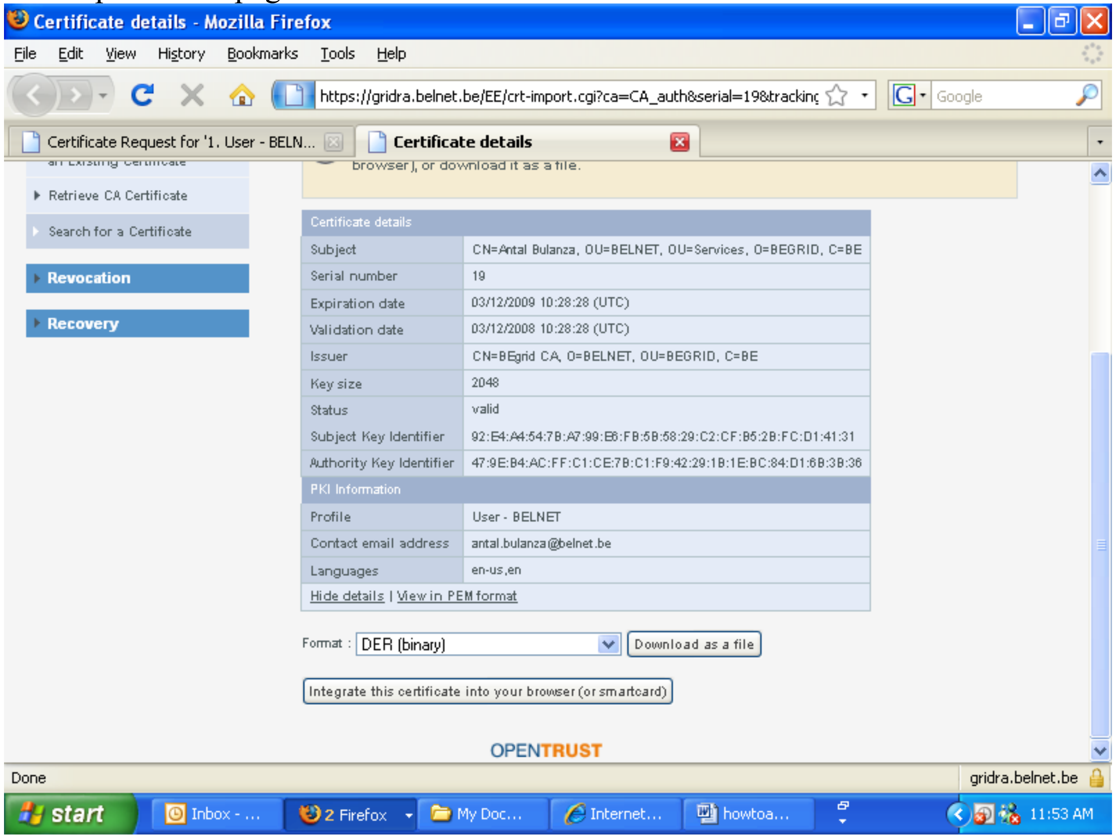

Certificate details - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/crt-import.cgi?ca=CA_auth&serial=19&tracking
Google
Certificate Request for '1. User - BELN...
Certificate details
BEgrid The Belgian Grid for Research
BELNET
Enrollment Entity
You did not provide a certificate.
Request a Certificate
Request a Certificate
Request a Certificate using an Existing Certificate
Retrieve CA Certificate
Search for a Certificate
Revocation
Recovery
Done
gridra.belnet.be
start
Inbox - ...
2 Firefox
My Doc...
Internet...
howtoa...
11:53 AM

Certificate details

This page displays the details of the selected certificate. You may integrate the certificate into your Web browser or smartcard (if the corresponding private key has been generated in the same browser), or download it as a file.

Certificate details	
Subject	CN=Antal Bulanza, OU=BELNET, OU=Services, O=BEGRID, C=BE
Serial number	19
Expiration date	03/12/2009 10:28:28 (UTC)
Validation date	03/12/2008 10:28:28 (UTC)
Issuer	CN=BEgrid CA, O=BELNET, OU=BEGRID, C=BE
Key size	2048
Status	valid
Subject Key Identifier	92:E4:A4:54:7B:A7:99:E6:FB:5B:58:29:C2:CF:B5:2B:FC:D1:41:31
Authority Key Identifier	47:9E:B4:AC:FF:C1:CE:7B:C1:F9:42:29:1B:1E:BC:84:D1:6B:3B:36
PKI Information	

Lower part of the page:



browser), or download it as a file.

Certificate details	
Subject	CN=Antal Bulanza, OU=BELNET, OU=Services, O=BEGRID, C=BE
Serial number	19
Expiration date	03/12/2009 10:28:28 (UTC)
Validation date	03/12/2008 10:28:28 (UTC)
Issuer	CN=BEgrid CA, O=BELNET, OU=BEGRID, C=BE
Key size	2048
Status	valid
Subject Key Identifier	92:E4:A4:54:7B:A7:99:E6:FB:5B:58:29:C2:CF:B5:2B:FC:D1:41:31
Authority Key Identifier	47:9E:B4:AC:FF:C1:CE:7B:C1:F9:42:29:1B:1E:BC:84:D1:6B:3B:36
PKI Information	
Profile	User - BELNET
Contact email address	antal.bulanza@belnet.be
Languages	en-us,en

Hide details | View in PEM format

Format : DER (binary) Download as a file

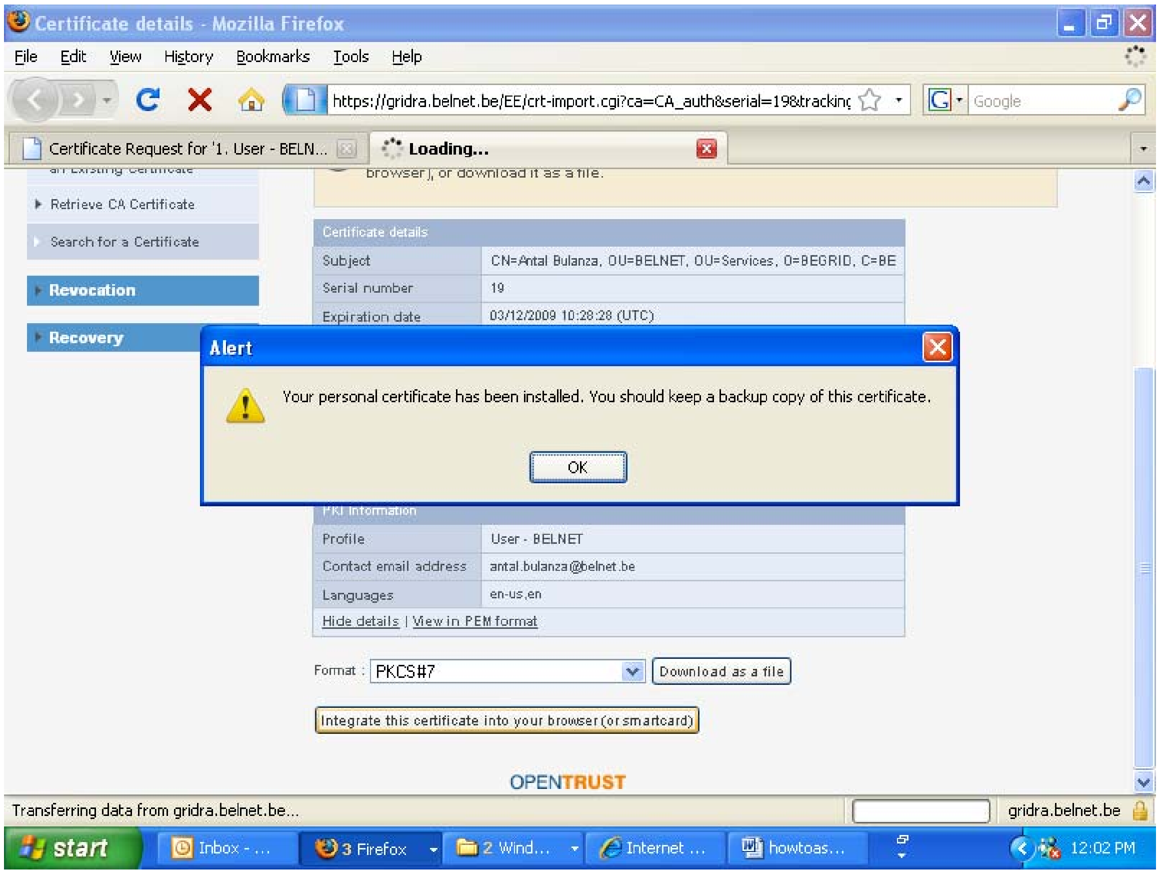
Integrate this certificate into your browser (or smartcard)

Certificate details - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/crt-import.cgi?ca=CA_auth&serial=19&tracking
Google
Certificate Request for '1. User - BELN...
Certificate details
Retrieve CA Certificate
Search for a Certificate
Revocation
Recovery
OPENTRUST
Done
gridra.belnet.be
start
Inbox - ...
2 Firefox
My Doc...
Internet...
howtoa...
11:53 AM

You can either import the certificate into your browser or download it
Different formats are available to download the certificate.

Import the certificate

Import the certificate in the browser by clicking on "Integrate this certificate into your browser or smartcard)". The result can be seen below.


Certificate details - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/crt-import.cgi?ca=CA_auth&serial=19&tracking
Certificate Request for '1. User - BELN...
Loading...
Retrieve CA Certificate
Search for a Certificate
Revocation
Recovery

browser), or download it as a file.

Certificate details	
Subject	CN=Antal Bulanza, OU=BELNET, OU=Services, O=BEGRID, C=BE
Serial number	19
Expiration date	03/12/2009 10:28:28 (UTC)

Alert

Your personal certificate has been installed. You should keep a backup copy of this certificate.

OK

PKI information	
Profile	User - BELNET
Contact email address	antal.bulanza@belnet.be
Languages	en-us,en

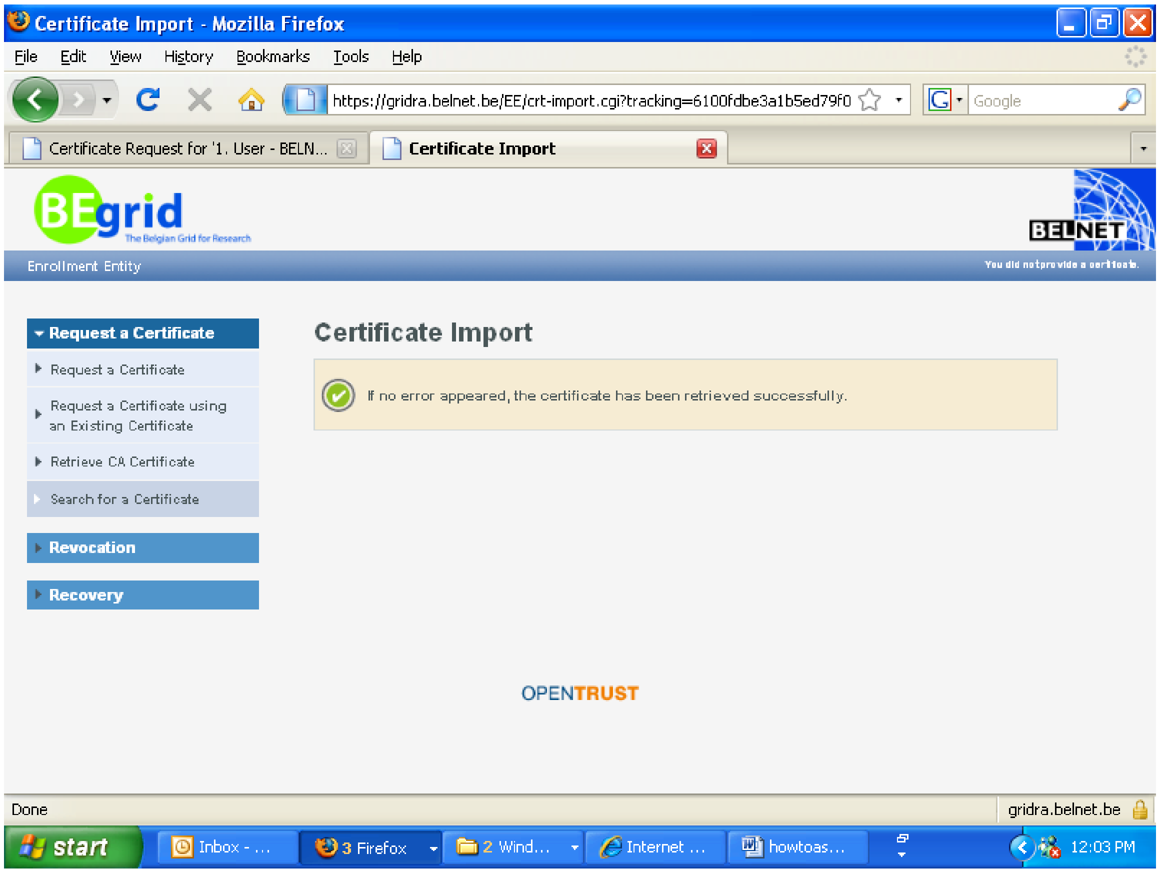
Hide details | View in PEM format

Format : PKCS#7 Download as a file

Integrate this certificate into your browser (or smartcard)

OPENTRUST
Transferring data from gridra.belnet.be...
gridra.belnet.be
start Inbox - ... 3 Firefox 2 Wind... Internet ... howtoas...
12:02 PM

Click "OK" and when the import is successful you obtain the following page:


Certificate Import - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/crt-import.cgi?tracking=6100fdbe3a1b5ed79f0
Certificate Request for '1. User - BELN...
Certificate Import
Enrollment Entity
Request a Certificate
Request a Certificate
Request a Certificate using an Existing Certificate
Retrieve CA Certificate
Search for a Certificate
Revocation
Recovery
OPENTRUST
Done
gridra.belnet.be

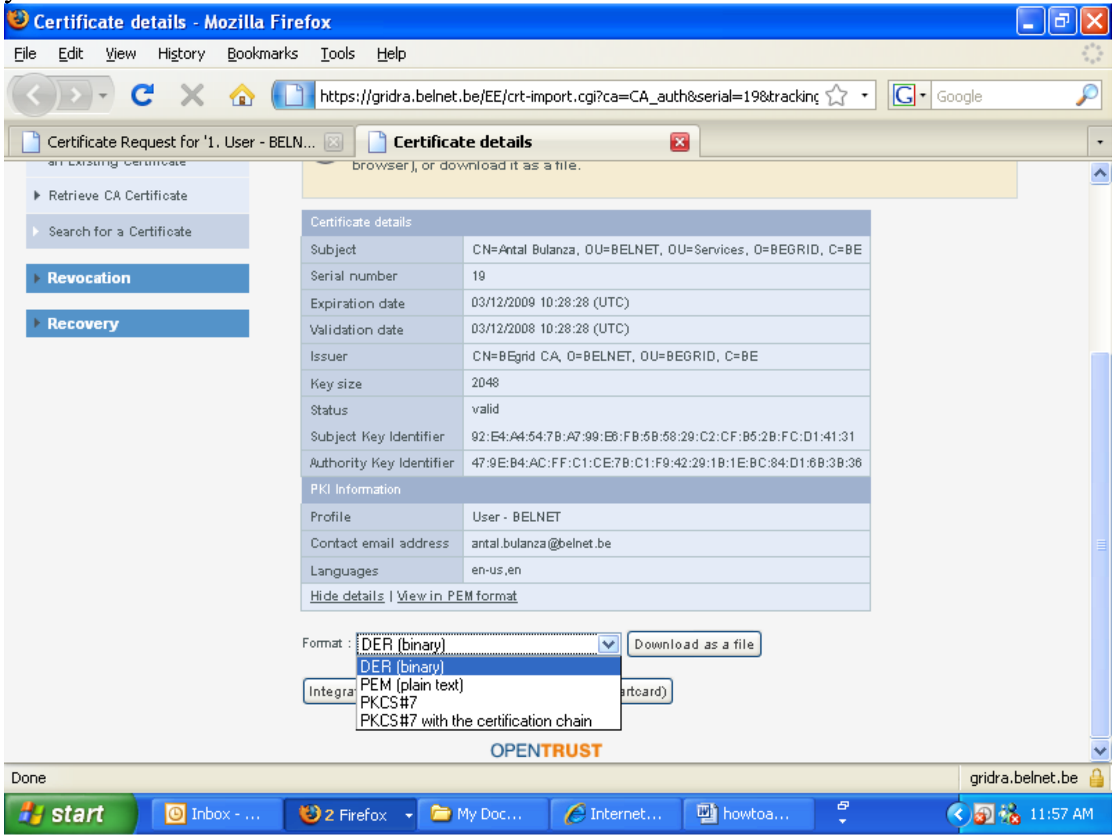
Certificate Import

If no error appeared, the certificate has been retrieved successfully.

Download certificate

You can download your certificate in several formats and save it as a file in your preferred directory. When requesting the "save file" the certificate will be stored on your desktop with the name indicated in the window (see below).

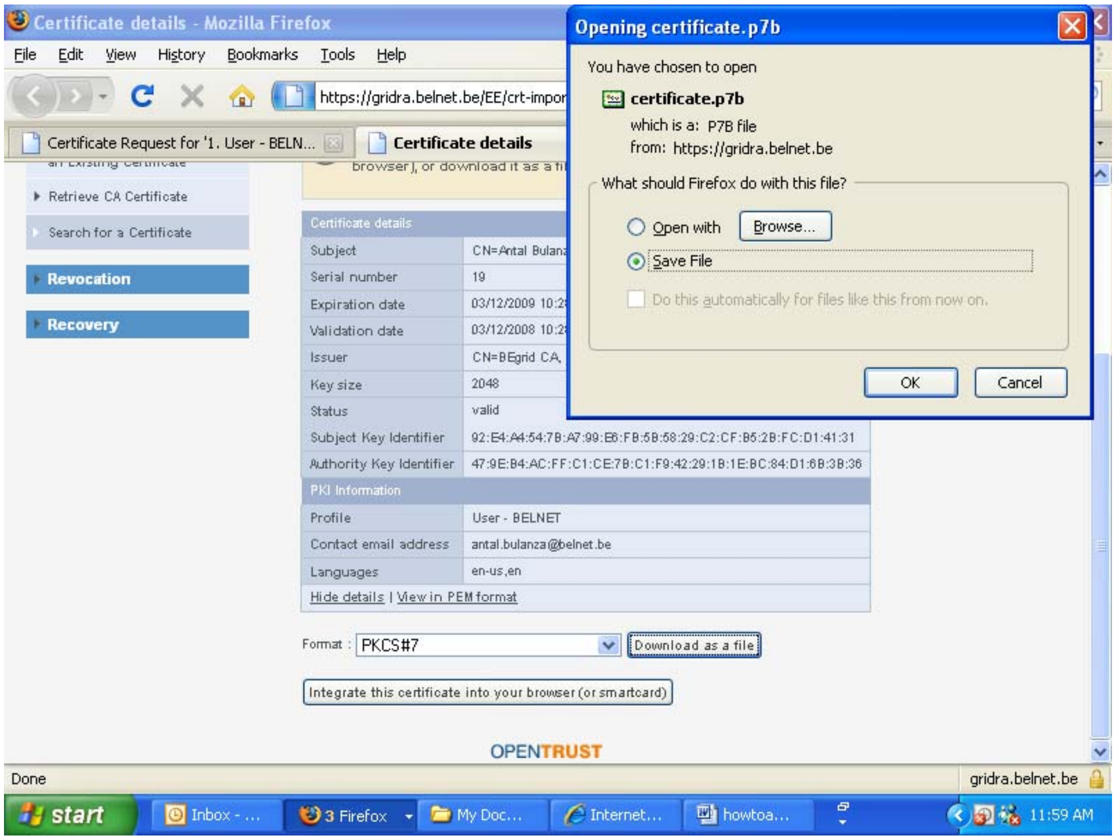
Depending on the application, different formats of the certificate can be needed. You can always go back to the CA web page, lookup your certificate(s) and download the format you need.


Certificate details - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/crt-import.cgi?ca=CA_auth&serial=19&tracking
Certificate Request for '1. User - BELN...
Certificate details
Retrieve CA Certificate
Search for a Certificate
Revocation
Recovery
browser), or download it as a file.

Certificate details	
Subject	CN=Antal Bulanza, OU=BELNET, OU=Services, O=BEGRID, C=BE
Serial number	19
Expiration date	03/12/2009 10:28:28 (UTC)
Validation date	03/12/2008 10:28:28 (UTC)
Issuer	CN=BEgrid CA, O=BELNET, OU=BEGRID, C=BE
Key size	2048
Status	valid
Subject Key Identifier	92:E4:A4:54:7B:A7:99:E6:FB:5B:58:29:C2:CF:B5:2B:FC:D1:41:31
Authority Key Identifier	47:9E:B4:AC:FF:C1:CE:7B:C1:F9:42:29:1B:1E:BC:84:D1:6B:3B:36
PKI Information	
Profile	User - BELNET
Contact email address	antal.bulanza@belnet.be
Languages	en-us,en
Hide details | View in PEM format	

Format : DER (binary)

Choosing PKCS#7


Certificate details - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/crt-impor
Certificate Request for '1. User - BELN...
Certificate details
Retrieve CA Certificate
Search for a Certificate
Revocation
Recovery
Format : PKCS#7
Download as a file
Integrate this certificate into your browser (or smartcard)
OPENTRUST
Done
gridra.belnet.be
start
11:59 AM

Opening certificate.p7b

You have chosen to open

certificate.p7b

which is a: P7B file

from: https://gridra.belnet.be

What should Firefox do with this file?

Open with [Browse...]

[✓ Save File]

Do this automatically for files like this from now on.

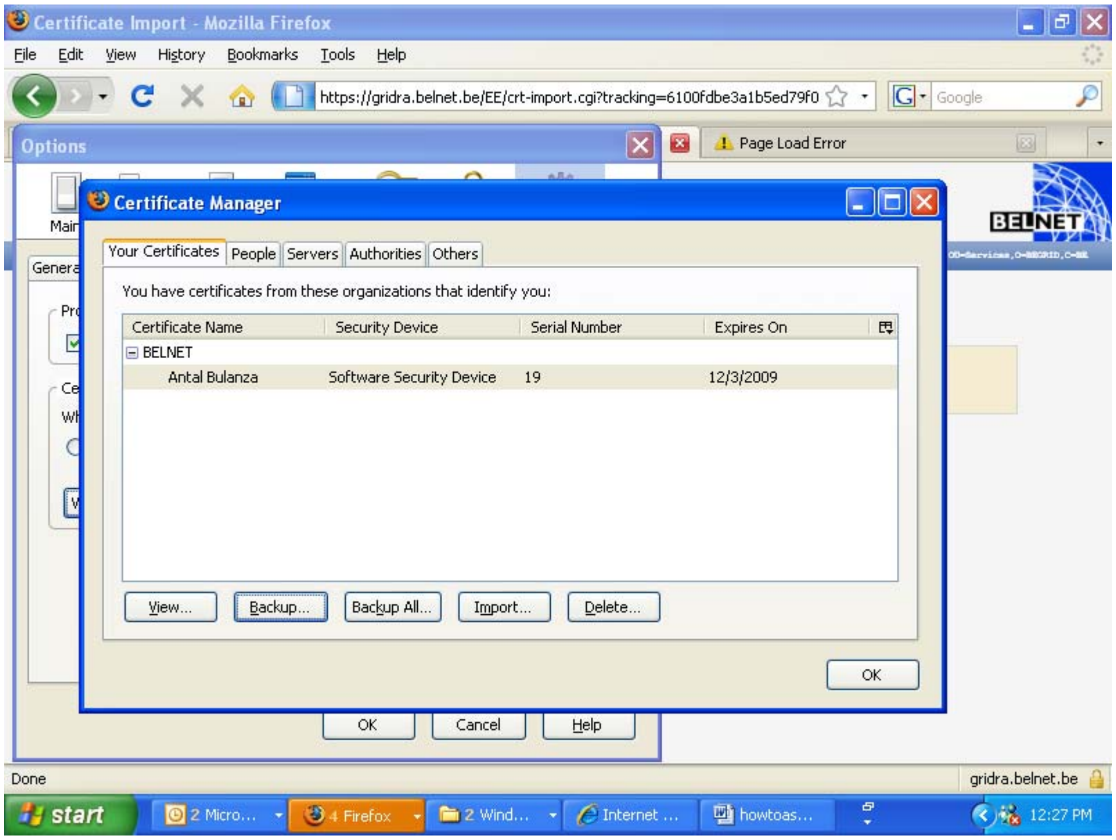
[OK] [Cancel]

Certificate details	
Subject	CN=Antal Bulanz
Serial number	19
Expiration date	03/12/2009 10:2
Validation date	03/12/2008 10:2
Issuer	CN=BEgrid CA,
Key size	2048
Status	valid
Subject Key Identifier	92:E4:A4:54:7B:A7:99:E6:FB:5B:58:29:C2:CF:B5:2B:FC:D1:41:31
Authority Key Identifier	47:9E:B4:AC:FF:C1:CE:7B:C1:F9:42:29:1B:1E:BC:84:D1:6B:3B:36
PKI Information	
Profile	User - BELNET
Contact email address	antal.bulanza@belnet.be
Languages	en-us,en

Hide details | View in PEM format

Backup of the certificate and private key

The backup/export button of permits you to make a copy of your scertificate and private key which are then protected by a password. See below:

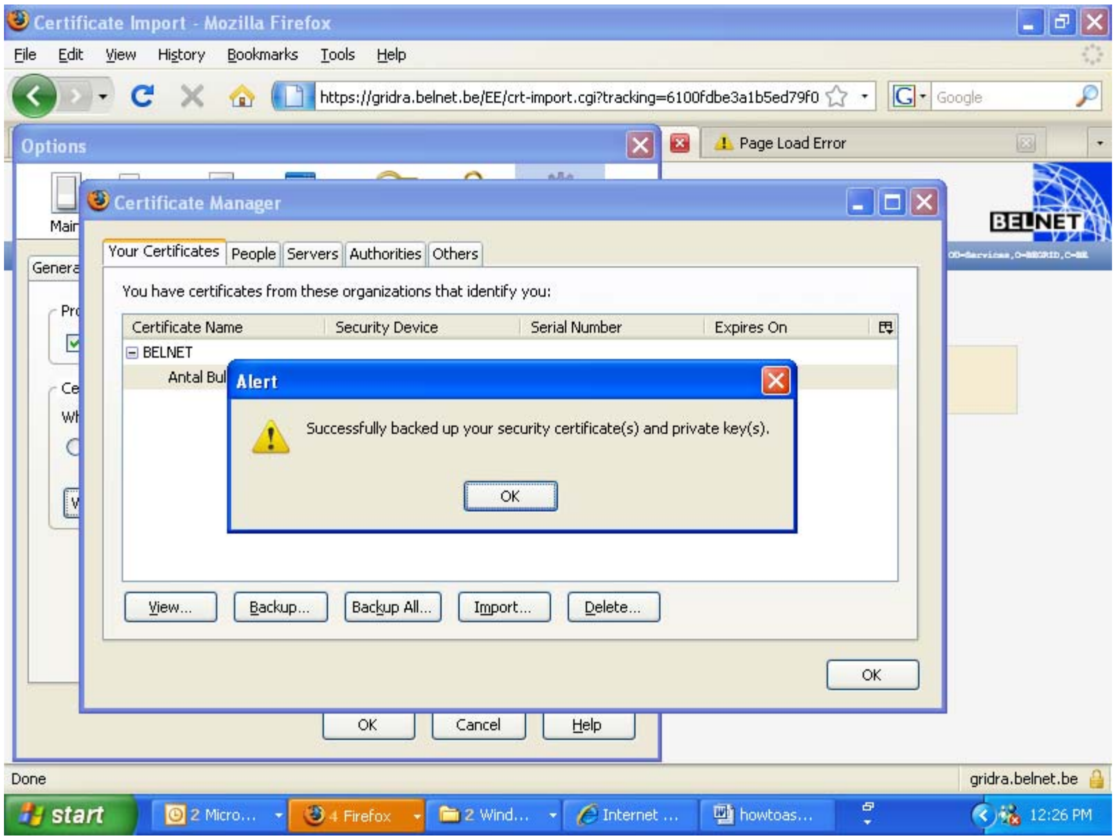

Certificate Import - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Options
Page Load Error
Certificate Manager
Your Certificates People Servers Authorities Others
View... Backup... Backup All... Import... Delete...
OK
OK Cancel Help
Done
gridra.belnet.be
start
12:27 PM

https://gridra.belnet.be/EE/crt-import.cgi?tracking=6100fdbe3a1b5ed79f0

You have certificates from these organizations that identify you:

Certificate Name	Security Device	Serial Number	Expires On
BELNET			
Antal Bulanza	Software Security Device	19	12/3/2009

Selecting the certificate


Certificate Import - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/crt-import.cgi?tracking=6100fdbe3a1b5ed79f0
Options
Page Load Error
Certificate Manager
Your Certificates
People
Servers
Authorities
Others
View... Backup... Backup All... Import... Delete...
OK
OK Cancel Help
Done
gridra.belnet.be
start
12:26 PM

You have certificates from these organizations that identify you:

Certificate Name	Security Device	Serial Number	Expires On
BELNET			
Antal Bul			

Alert

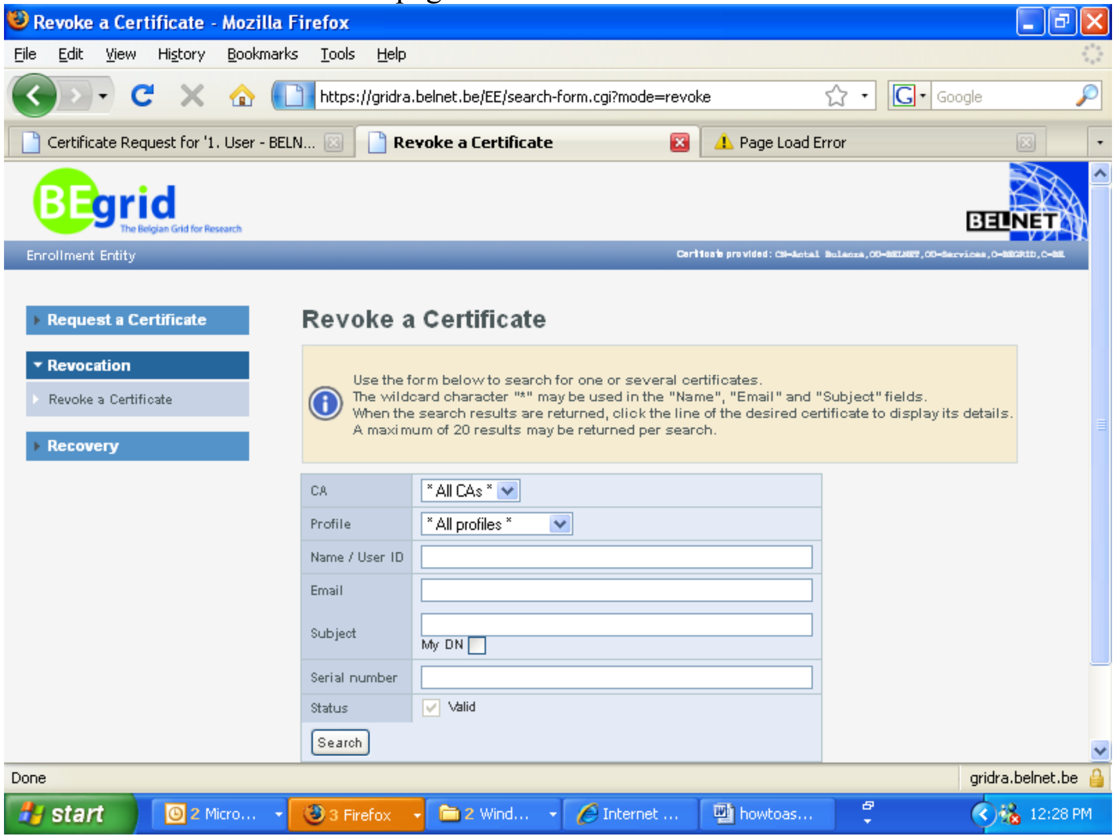
Successfully backed up your security certificate(s) and private key(s).

OK

Revocation of a certificate

It can be necessary or useful to revoke a certificate, e. g. hacking of your desktop, leaving your organization, … For completeness of the course we show how to do a revocation but do not do it now!!!

Select "revocation" on the EE page.


Revoke a Certificate - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/search-form.cgi?mode=revoke
Google
Certificate Request for '1. User - BELN...
Revoke a Certificate
Page Load Error
BEgrid
The Belgian Grid for Research
BELNET
Enrollment Entity
Request a Certificate
Revocation
Revoke a Certificate
Recovery
Done
gridra.belnet.be
start
12:28 PM

Certificate provided: CN=Antal Bulanza,OU=BELNET,OU=Services,O=BEGRID,C=BE

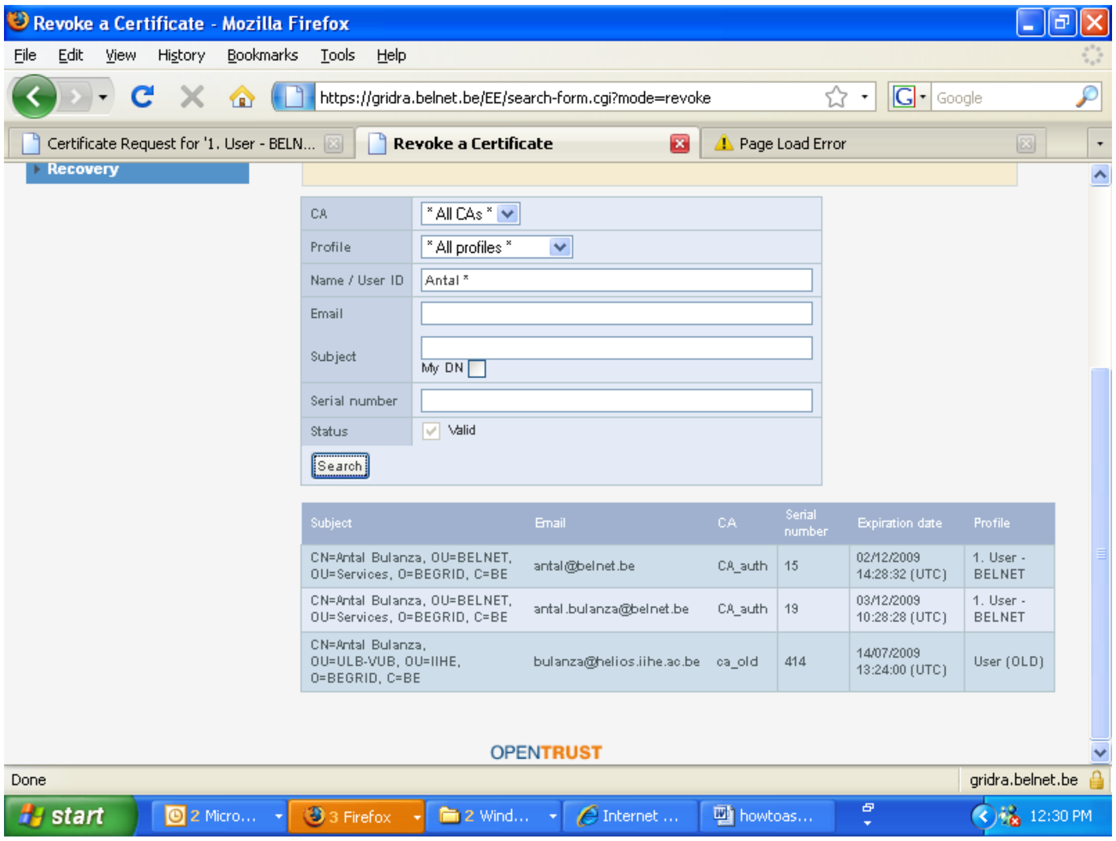
Revoke a Certificate

Use the form below to search for one or several certificates.
The wildcard character "*" may be used in the "Name", "Email" and "Subject" fields.
When the search results are returned, click the line of the desired certificate to display its details.
A maximum of 20 results may be returned per search.

CA	* All CAs *
Profile	* All profiles *
Name / User ID	
Email	
Subject	My DN ☐
Serial number	
Status	✓ Valid

Search

Search certificate you want to revoke using its serial number or by name:



Revoke a Certificate - Mozilla Firefox

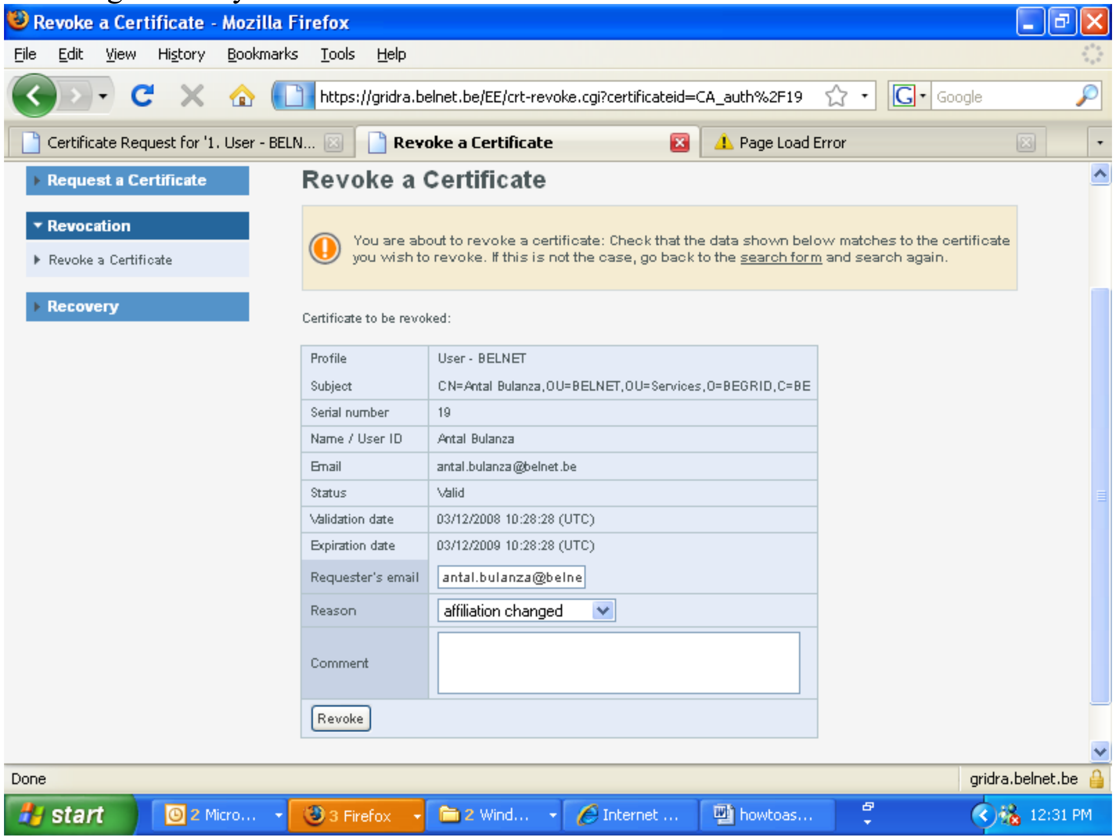
https://gridra.belnet.be/EE/search-form.cgi?mode=revoke

CA	* All CAs *
Profile	* All profiles *
Name / User ID	Antal *
Email	
Subject	My DN ☐
Serial number	
Status	✓ Valid

Search

Subject	Email	CA	Serial number	Expiration date	Profile
CN=Antal Bulanza, OU=BELNET, OU=Services, O=BEGRID, C=BE	antal@belnet.be	CA_auth	15	02/12/2009 14:28:32 (UTC)	1. User - BELNET
CN=Antal Bulanza, OU=BELNET, OU=Services, O=BEGRID, C=BE	antal.bulanza@belnet.be	CA_auth	19	03/12/2009 10:28:28 (UTC)	1. User - BELNET
CN=Antal Bulanza, OU=ULB-VUB, OU=IIHE, O=BEGRID, C=BE	bulanza@helios.iihe.ac.be	ca_old	414	14/07/2009 13:24:00 (UTC)	User (OLD)

Selecting the one you want to revoke:



Revoke a Certificate - Mozilla Firefox

https://gridra.belnet.be/EE/crt-revoke.cgi?certificateid=CA_auth%2F19

Revoke a Certificate

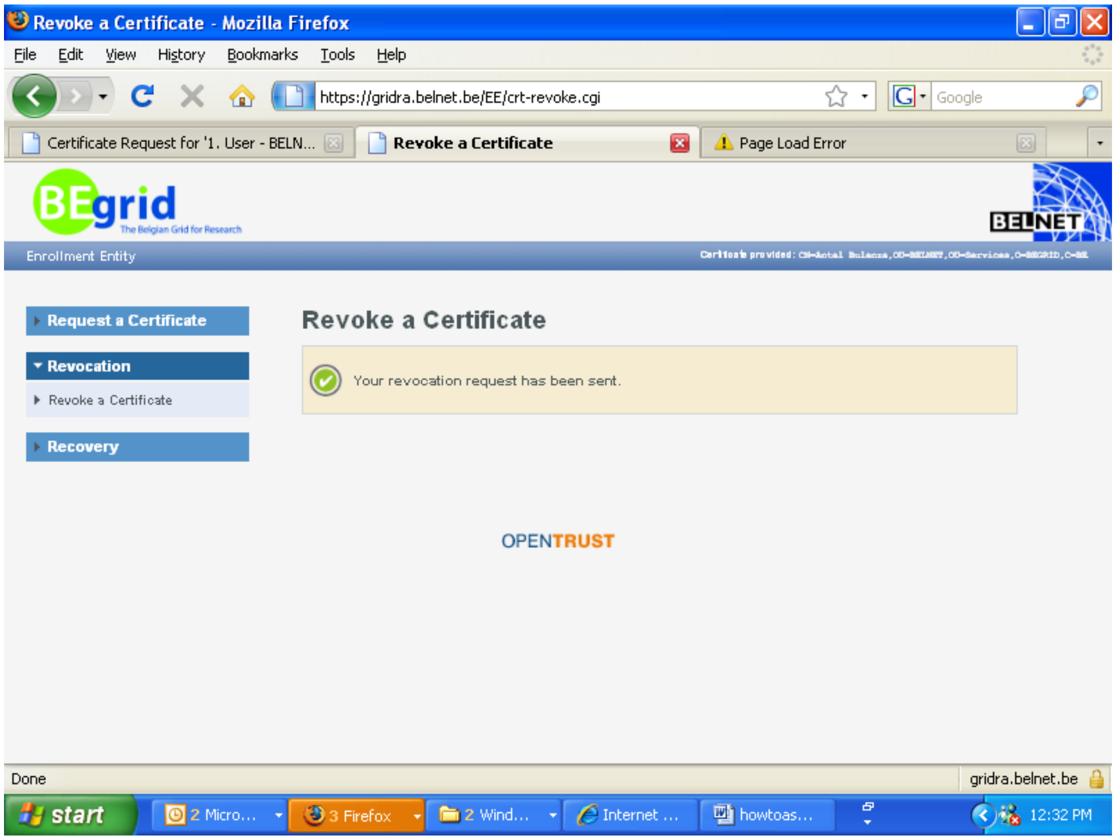
You are about to revoke a certificate: Check that the data shown below matches to the certificate you wish to revoke. If this is not the case, go back to the search form and search again.

Certificate to be revoked:

Profile	User - BELNET
Subject	CN=Antal Bulanza,OU=BELNET,OU=Services,O=BEGRID,C=BE
Serial number	19
Name / User ID	Antal Bulanza
Email	antal.bulanza@belnet.be
Status	Valid
Validation date	03/12/2008 10:28:28 (UTC)
Expiration date	03/12/2009 10:28:28 (UTC)
Requester's email	antal.bulanza@belne
Reason	affiliation changed
Comment	

Revoke

The owner of the certificate should receive an email to confirm this action.


Revoke a Certificate - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://gridra.belnet.be/EE/crt-revoke.cgi
Certificate Request for '1. User - BELN...
Revoke a Certificate
Page Load Error
Enrollment Entity
Certificate provided: CN=Antal Bulanza,OU=BELNET,OU=Services,O=BEGRID,C=BE
Request a Certificate
Revocation
Revoke a Certificate
Recovery
OPENTRUST
Done
gridra.belnet.be
start
12:32 PM

Revoke a Certificate

Your revocation request has been sent.

2 Exercise 2: Obtaining the betest VO membership

Goal: Request to become member of the betest VO (Virtual Organisation).

Requisites:

- The user must have his/her certificate installed on his/her computer./browser
- The user's email address is mandatory to request a certificate.
- The user must have access to his/her mail.

Important: *Don't ask to become member of betest if you are already a member.*

If you want to use BEgrid you should register with a Virtual Organization (VO) like *betest*. A Virtual Organization is a collection of individuals collaborating within a subject area, a single experiment or any other thematic grouping and using part or all of the grid resources.
To do this, you must authenticate with your certificate to VO management web site, and thus you should have your certificate available inside your web browser.

Practice:

2. Request
2.1
https://voms.begrid.be:8443/voms/betest

2.2
The system may ask you to choose your certificate. Select your certificate and continue.

2.3
You may receive a pop-up asking you "an application is trying to access to…". Choose "OK".
You enter on the VOMS webpage.

2.4
On the "FOR VO USERS" menu select "New User Registration"

You may receive a pop-up asking to select your certificate. Choose the certificate and continue.

You will get a new page with some pre-filled information:

- DN. Ex. /C=BE/O=BEGRID/OU=IIHE/OU=ULB-VUB/CN=Cours Test 01.
- CA Ex. /C=BE/O=BELNET/OU=BEGrid/CN=BEGrid CA/Email=gridca@belnet.be.
- CA URI. Ex. https://gridra.belnet.be/pub/crl/cacrl.crl

You now have to fill a form with some personal information:

- Family Name
- Given Name
- Institute
- Phone Number
- Email
- Comment

Note that some fields could be already filled. The "comment" field can be left empty.
Click on the "**I have read and agree to the VO' Usage Rules**".

A new page shows you the information that you have entered and your request number.

You have created a new request (id 65):

DN:	/C=BE/O=BEGRID/OU=IIHE/OU=ULB-VUB/CN=Cours Test 01
CA:	/C=BE/O=BELNET/OU=BEGrid/CN=BEGrid CA/Email=gridca@belnet.be
CA URI:	┘ 40200 . ,†*https://gridra.belnet.be/pub/crl/cacrl.crl
Name:	Test, Cours
Email:	gridusername@helios.iihe.ac.be
comment:	

Your registration request is stored, but it will not be processed until you confirm your email address.
An email message is sent to you with the id of your request and a random cookie, which you can use as a password for the confirmation. To simplify the process you may simply point your browser (with the credentials, you have used to initiate the registration) to the URL in the email.
Once you have confirmed your email address, the request will be forwarded to the VO administrators.

2.6
Check your email. You should receive an email from voms-admin@begrid.be similar to this one:

Email address confirmation for VO betest

A request for a VO membership on betest has been made using this email address.

If you have not made this request please ignore this message. It would be helpful if you would contact the VO registrar and tell us about this bogus request.

If the request was made by you, please click on the following URL to confirm this email address,

https://voms.begrid.be:8443/voms/betest/webui/request/user/confirm?cookie=9y6t8ucstv5kk7oh&reqid=159

Make sure you have your client certificate loaded in your browser.
One way to ensure this is to copy and paste the above URL into the same browser that you used to submit the request.

If you wish to confirm the request another way, then you need the following information:

Request number : 159
Confirmation cookie: 9y6t8ucstv5kk7oh

Once you have confirmed your email address the following information will be sent to the VO administrator:

Name : Test, Cours
Email : gridusername@helios.iihe.ac.be
Institute : ULB-VUB
Phone Number : 0498316938
Certificate DN : /C=BE/O=BEGRID/OU=IIHE/OU=ULB-VUB/CN=Cours Test 01
Certificate CA : /C=BE/O=BELNET/OU=BEGrid/CN=BEGrid CA/Email=gridca@belnet.be
Comment :

The VO administrator will probably contact you to confirm account creation.

Thank You,
VO Registration

2.7
Click on the indicated URL to confirm your email address.

You may receive a pop-up asking to select your certificate. Choose the certificate and continue.

You may receive a pop-up asking you “an application is trying to access to…”. Choose “OK”.

A new webpage shows you information similar to:

You have successfully confirmed the following request:

Id:	65
Status:	New
Container:	betest
Requester:	/C=BE/O=BEGRID/OU=IIHE/OU=ULB-VUB/CN=Cours Test 01
Description:	User creation: DN=/C=BE/O=BEGRID/OU=IIHE/OU=ULB-VUB/CN=Cours Test 01, CA=/C=BE/O=BELNET/OU=BEGrid/CN=BEGrid CA/Email=gridca@belnet.be

2.1.1 Email Address Confirmation

Your request for the LCG account is confirmed and the request has been forwarded to the Virtual Organisation.

This part of the registration process is complete. You should now be contacted directly by the VO manager who will complete the registration process.

2.8
Check your email. You should receive an email from voms-admin@begrid.be similar to this one:

Accepted email confirmation for VO

Dear Test, Cours,

Thank you for confirming your email address. Your request for an account on VO betest has been sent to the VO administrator(s).

The VO administrator will contact you to confirm account creation.

If you find any problems regarding the account registration, then please contact the VO registrar.

Thank You,
VO Registration

2.9
Once your request to be member of the betest VO has been accepted, you will receive an email from “voms-admin@begrid.be” like:

Accepted VO membership request

Dear Test, Cours,

Your request (65) for the betest VO has been accepted and allowed by the VO Administrator.

From this point you can use the edg-voms-proxy-init command to acquire the VO specific credentials, which will enable you to use the resources of this VO.

Good Luck,
VO Registration

Comment: Do not pay attention to the detailed text in these messages. The content of the messages concerning VOMS membership requests have not been adapted to the new gLite commands.

Proxy

3 Exercise 3: Create a proxy

Goal:

- Export your personal certificate and copy it into the UI (User Interface).
- Install the certificate on the UI.
- Initiate your proxy.
- See some basic commands related with the proxy.

Requisites:

- The user must have his certificate installed on his compute/browserr.
- Login with user/password received to access the UI.
- Have a “secure login shell” program installed on the user’s computer. (ex. Putty)
- Have a “secure file transfer” program installed on the user’s computer. (ex. WinSCP)

The Begrid certificate you got is valid for one year. For grid security reasons you don't use your certificate directly for authentication when you submit jobs or store data.. Instead you use a proxy certificate. This proxy is valid for several hours (12 hours by default).but can be extended.

Practice:

Exporting the certificate obtained and the private key

Note that you already did this in chapter 1 after you got a valid certificate. Below we repeat the procedure for different browsers.

3.1 Export the certificate using Internet Explorer

Important: *The following steps are for Internet Explorer, if you are using a different browser the steps may be different.*

If you are using Mozilla, please go to step 3.2
If you are using a Macintosh go to step 3.3

3.1.1
Open your browser.
Go to the “Tools” menu and take “Internet Options”.
Choose the “Content” tab.
Select the “Certificates” option.
Take “personal” tab and select your certificate.

Click the Export button and follow the instructions:

- Select the “Yes, export the private key” radio button
- Make sure the “Enable strong protection” box is checked.

When prompted for the file you want to export: provide the directory where to save the file and the filename of the certificate (user.p12, user.pfx OR yourname.p12, etc) in the export format.

*Usually the file is exported with the pkcs12 format (file *.p12), but may have the format *.pfx This file contains your public and private keys.*

You will be prompted for a password to export the certificate. Remember this password, as you will need it to re-import the certificate into the UI machine.

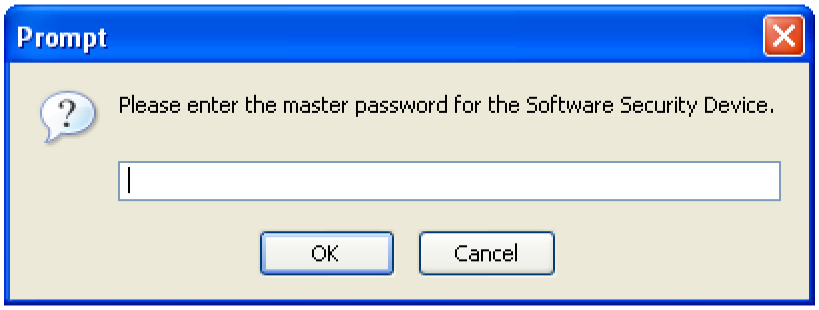
To continue with IE, please go to point 3.4.

3.2 Export the certificate using Mozilla

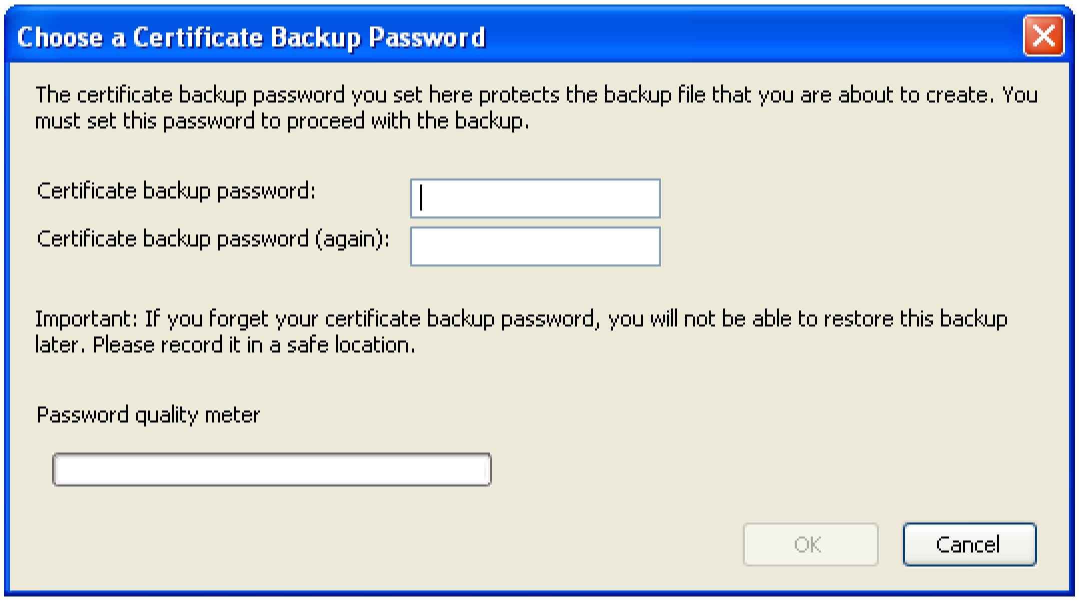
3.2.1
Open your browser.
Go to the tools menu.
Select Options, and then Advanced.
On the Security tab, choose select “Your Certificates”.
Choose yours form the list and click on the Backup button.

Save it on your disk selecting the “PKCS12” type option.

The master password may be required.


Please enter the master password for the Software Security Device.

Then you have to create your certificate backup password.
Remember this password, as you will need it to re-import the certificate into the UI machine.


Choose a Certificate Backup Password

The certificate backup password you set here protects the backup file that you are about to create. You must set this password to proceed with the backup.

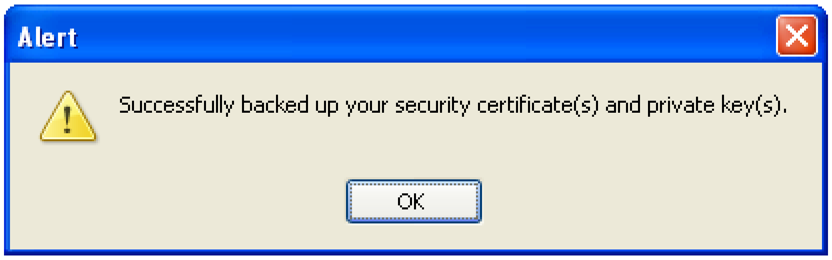
Certificate backup password:

Certificate backup password (again):

Important: If you forget your certificate backup password, you will not be able to restore this backup later. Please record it in a safe location.

Password quality meter

If the backup has been done correctly you will receive this message:


Alert

Successfully backed up your security certificate(s) and private key(s).

To continue with Mozilla, please go to point 3.4.

3.3 Export the certificate using a <u>Macintosh</u>

3.3.1
Open the program "Keychain Access" (it is on Applications/Utilities folder).
Choose the certificate that you have downloaded.
Export it selecting *.pem format.

Importing the personal certificate

3.4 Import the certificate into the User Interface (UI)

Important: *The name of the computer to log on (the UI), your login and your password are given on the paper you have received at the beginning of the training.*

3.4.1
Transfer your personal certificate from your PC to the UI.
Using the "secure file transfer" program (e.g. WinSCP) copy your *.p12 or *.pfx file from Windows to a subdirectory called ".globus" in your home directory on the UI.
user.p12 is the personal certificate requested and saved in a directory on your PC.

3.4.2
Use the "secure login shell" program to login into the UI (User Interface).

3.4.3
Move to the .globus directory and check that the certificate was well transferred

```
cd .globus
ls
```

3.4.4
When you export your private and public key from a browser it is exported in PKCS12 format (*.p12 or *.pfx), to use it with gLite software you have to convert it into PEM format.

The certificate format depends on the application, as there is no agreement on file format standards. In general, the PEM formats are mostly used in the Unix/Linux world and PCKS12 in the Microsoft world.

When executing the openssl command you will extract your public key from the user.p12 file and you will save it on the usercert.pem file.
Don't change the name of the usercert.pem file!!

```
openssl pkcs12 -nokeys -clcerts -in user.p12 -out usercert.pem
```

*The user file may be with a *.pfx extension instead of *.p12.*

Important*: For the Macintosh users. Please use the following command instead of the previous one:*

```
openssl x509 -in certificate.pem -pubkey >usercert.pem
```

You will now be prompted to enter the passphrase (password) protecting the certificate:

```
Enter Import Password: *****
MAC verified OK
```

3.4.5
Convert the private key file to the PEM format.

Don't change the name of the userkey.pem file!!

```
openssl pkcs12 -nocerts -in user.p12 -out userkey.pem
```

You will now be prompted to enter the passphrase (password - the same used to export the certificate on 3.1.1 and 3.2.1) protecting the certificate:

```
Enter Import Password: *****
MAC verified OK
```

You will now need to create a strong passphrase (password) that will protect your private key.

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

3.4.6
Change the rigths of the usercert.pem file to …

```
chmod 644 usercert.pem
```

3.4.7
Change the rigths of the userkey.pem file to …

```
chmod 600 userkey.pem
```

3.4.8
Verify that the file rights are correct.

```
ls -la
```

You should have something like this:

```
total 32
drwxr-xr-x   2 gridusername betest      4096 Oct 10 16:13 .
drwxr-xr-x   6 gridusername betest      4096 Feb 23 10:53 ..
-rw-r--r--   1 gridusername betest      1925 Oct 10 16:12 usercert.pem
-rw-------   1 gridusername betest      1202 Oct 10 16:13 userkey.pem
```

3.5 Verify the certificate

3.5.1
Execute the following openssl command to verify your certificate.

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

The result should be something like this:

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

3.5.2
Verify the content of your certificate

```
grid-cert-info
```

The result should be something like:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 188 (0xbc)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=BE, O=BELNET, OU=BEGrid, CN=BEGrid CA/Email=gridca@belnet.be
        Validity
            Not Before: Sep 21 10:05:55 2005 GMT
            Not After : Sep 21 10:05:55 2006 GMT
        Subject: C=BE, O=BEGRID, OU=IIHE, OU=ULB-VUB, CN=Genius Gridusername
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:dc:7e:b6:07:54:e9:09:2c:30:b7:d6:b3:9a:4f:
                    57:e8:05:75:18:75:fc:dd:80:fa:8b:59:31:fc:f7:
                    fa:45:1b:9a:20:3a:21:ed:be:91:83:28:40:c2:34:
                    08:11:cd:01:ce:a1:0f:9c:34:43:e8:84:99:e1:4b:
```

```
                55:40:26:f6:f7:d8:59:37:e3:ec:c6:0c:13:79:ff:
                74:50:33:4a:d1:b1:dc:da:b0:99:9a:1b:3a:8b:ed:
                f0:4b:8d:b0:f6:52:ae:be:dc:b6:22:94:5b:88:ea:
                68:2b:da:c7:1f:e2:5d:cc:36:22:3c:b1:34:57:29:
                53:60:c8:d3:17:1b:e1:87:c5
            Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Cert Type:
                SSL Client, S/MIME
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Client Authentication, E-mail Protection, Microsoft Smartcardlogin
            Netscape Comment:
                User Certificate of BEGRID
            X509v3 Subject Key Identifier:
                BC:B3:41:0A:7C:C5:9A:AB:99:E8:3C:83:A2:81:E6:C1:D7:BA:F5:A2
            X509v3 Authority Key Identifier:
                keyid:95:E9:E7:D8:07:BA:03:CA:82:17:B3:C6:C9:C5:96:C5:7D:EC:08:0F
                DirName:/C=BE/O=BELNET/OU=BEGrid/CN=BEGrid CA/Email=gridca@belnet.be
                serial:00

            X509v3 Subject Alternative Name:
                email:gridusername@helios.iihe.ac.be
            X509v3 Issuer Alternative Name:
                email:gridca@belnet.be
            Netscape CA Revocation Url:
                https://gridra.belnet.be/pub/crl/cacrl.crl
            Netscape Revocation Url:
                https://gridra.belnet.be/pub/crl/cacrl.crl
            X509v3 CRL Distribution Points:
                URI:https://gridra.belnet.be/pub/crl/cacrl.crl

    Signature Algorithm: sha1WithRSAEncryption
        0a:eb:98:82:52:98:d0:34:a0:2c:44:48:08:5f:a6:e3:20:8f:
        46:49:ac:73:d6:ec:43:39:8f:37:12:d2:30:95:d3:e0:ac:3e:
        c5:52:3b:c8:b2:46:cd:f2:88:10:4b:7b:b8:5d:aa:55:b7:99:
        b5:f4:f9:57:f8:72:33:f7:a4:18:bc:4a:73:d5:ab:1f:0f:99:
        7d:50:42:38:2d:f8:b1:bc:a0:f5:02:86:3f:a1:11:19:e4:f6:
        37:af:79:f3:6e:54:2d:b3:c5:41:d6:a4:80:b7:22:97:1e:95:
        a5:f8:0c:e8:68:ca:f0:ab:71:09:9b:e4:73:34:9b:db:fa:ad:
        0b:f6:a6:a7:b3:5d:70:e0:6f:6e:7b:43:58:87:d2:11:7e:d7:
        d1:5d:34:9e:ef:96:b5:32:83:da:0a:e2:92:b8:f6:60:13:25:
        4f:b3:66:cb:73:81:56:8a:16:b1:7d:89:15:05:98:bf:92:73:
        42:81:af:ef:d5:7a:9f:0e:97:47:f5:db:12:30:27:e6:fc:c2:
        1a:df:51:12:39:71:00:c7:a9:35:4a:79:60:ac:93:d3:9f:95:
        72:57:d3:45:a5:a5:30:95:77:87:cc:62:af:ab:65:a6:4d:be:
        a6:a9:1b:20:ff:a2:8d:50:8a:1d:d0:5f:ac:3d:50:71:f8:5f:
        3b:84:b4:80
```

3.5.3
The grid-cert-info command takes many options. Use “grid-cert-info -help” for a full list.
For example, the -subject option:

```
grid-cert-info -subject
```

Returns the certificate subject:

```
/C=BE/O=BEGRID/OU=IIHE/OU=ULB-VUB/CN=Genius Gridusername
```

This means:

```
/C=BE/O=BEGRID/OU=IIHE/OU=ULB-VUB/CN= Genius Gridusername
    |        |           |          |               |
    |        |           |          |               --- CN =Common Name
    |        |           |          |
    |        |           |          ----- OU = Organisation Unit
    |        |           |
    |        |           ----- OU = Organization Unit
    |        |
    |        --- O = Organization
    |
    ---- C = Country
```

This information above is useful because when after a year you have to request a new certificate it is important that you make the request with the same parameters.

3.6. Create a proxy certificate

3.6.1
Create your proxy. With the VOMS “voms-proxy-init” command.

```
voms-proxy-init –voms betest
```

The system will ask you for a password (passphrase).

The password is the one you have chosen on page 39 when creating a strong password to protect the private key.

```
Your identity: /C=BE/O=BEGRID/OU=IIHE/OU=ULB-VUB/CN= Genius Gridusername
Enter GRID pass phrase:
Creating temporary proxy ................................... Done
Contacting  voms.begrid.be:18003
[/C=BE/O=BEGRID/OU=Begrid/OU=Belnet/CN=voms.begrid.be] "betest" Done
Creating proxy ............................................................ Done
Your proxy is valid until Thu Nov 15 05:51:15 2007
```

The proxy has a default time left of 12 hours.

3.7 Obtain information about proxy

3.7.1
Execute the voms-proxy-info command.

```
voms-proxy-info
```

The result should be similar to:

```
subject   : /C=BE/O=BEGRID/OU=IIHE/OU=ULB-VUB/CN= Genius Gridusername /CN=proxy
issuer    : /C=BE/O=BEGRID/OU=IIHE/OU=ULB-VUB/CN= Genius Gridusername
identity  : /C=BE/O=BEGRID/OU=IIHE/OU=ULB-VUB/CN= Genius Gridusername
type      : proxy
strength  : 512 bits
path      : /tmp/x509up_u19694
timeleft  : 11:57:35
```

Verify timeleft.

3.7.2
Change proxy settings: use "voms-proxy-init" command with arguments that can be obtained using "-help".
Change the proxy lifetime to 17 hours.

!!! the maximum lifetime for the VO betest is 24 hours. The maximum lifetime in beapps is 96 hours. For jobs that need a longer run time there exist mechanisms (MyProxy) to prolong the proxy.

3.8 Destroying the proxy certificate

3.8.1

```
voms-proxy-destroy
```

3.9 Combined exercise

3.10.1
Create a new proxy certificate with a duration of 14 hours.

Remember

voms-proxy-init	to get a key, a pass phrase will be required
voms-proxy-info	gives information of the proxy certificate in use
voms-proxy-destroy	destroys the proxy certificate for this session
voms-proxy-xxx -help	shows the usage of the command grid-proxy-xxx

For the corresponding command for "**grid-proxy-xxx**", replace the word "voms" by "grid"

Comment: You might also encounter the commands "grid-proxy-xxx". Those commands are predecessors of the voms-proxy-xxx commands. These last ones include VOMS

information into the proxy and are backwards compatible with the grid-info-xxx commands.

Job Submission

Exercise 4: Submit jobs to the grid

Goal:

- Create jobs.
- Check for available resources.
- Send jobs to the grid.
- Retrieve results.

Requisites:

- User has his certificate installed on the UI.
- User has a valid proxy certificate.

As of Dec 2007, the gLte/WMS middleware is available and and has now replaced the former RB (Resource Broker). A WMS key component is the WMProxy which is responsible for accepting incoming requests from the User Interface (e.g. job submission, job removal).
What happens when users submit jobs to the Grid:

1. User submits the job from the User Interface (UI) to the Workload Management System (WMS). The WMS does the matchmaking to find out where the job might be executed. After having found such a Computing Element (CE) the Job is transferred to the Job Submission System (JSS). At the JSS a file is created in Resource Specification Language (RSL). Also at this stage the Input SandBox is created in which all needed files by the job are specified.
2. This RSL file, together with the Input SandBox, is then transferred to the Gatekeeper of the CE and the Gatekeeper submits the job to the Local Resource Management System (LRMS).
3. The LRMS sends the job to one of the free Worker Nodes of the Computing Element.
4. When the job has finished, the files produced by the job are available on the LRMS. The job manager running on the CE notifies the WMS that the job has completed.
5. The WMS subsequently retrieves those files specified in the OutputSandBox.
6. The WMS sends the results (the OutputSandBox) back to the user on the User Interface machine.
7. Queries by the user on the status of the job are sent to the Logging and Bookkeeping Service.

Users access the GRID through a UI machine that allows the user to submit a job, to monitor its status, and to retrieve the output from the Worker Node back to a local directory on the UI machine. To do so, a simple Job Description Language (JDL) file is compiled. In this file all parameters to run the job are specified.

To submit a job, you need to have a valid proxy. Furthermore each job submitted to the gLite WMS must be associated to a proxy previously delegated by the owner of the job to the WMProxy server. This proxy is then used any time WMproxy needs to interact with other services for job related operations.
There are two ways to delegate credentials to WMProxy: either by having the delegation performed automatically each time a job is submitted, or by explicitly creating a named delegated credential on the WMProxy server, and subsequently referring to it each time a job is submitted. The advantage of the former method is simplicity, while that of the latter is better performance.

First create a valid proxy (see previous chapter how to do).

To explicitly delegate a user proxy to WMProxy the command to use is:

```
glite-wms-job-delegate-proxy –d <delegID>
```

Where <delegID> is a string chosen by the user. Subsequent invocations of glite-wms-job-submit and glite-job-list-match can bypass the delegation of a new proxy if the same <delegID> is given to the –d option.

Fr example
glite-wms-job-delegate-proxy –d mydelegID

and

glite-wms-job-submit –d mydelegID test.jdl

The automatic delegation can be used immediately in the job submission command:
glite-wms-job-submit -a <jdl_file>

We will use the method with automatic delegation using the **-a** option during the following exercises..

Practice:

For exercises 4.1 - 4.6 we will use the HelloWorld.jdl file

```
cd exercise1
```

Verify that you have the HelloWorld.jdl file on your directory!!

```
ls
```

4.1 Create a job

We use the JDL language to declare our job and its requirements. You can find more information about the JDL language on the *reference* page you have received.

4.1.1
See the content of the HelloWorld.jdl file.

```
more HelloWorld.jdl
```

The content must be:

```
    Executable = "/bin/echo";
    Arguments = "Hello World";
    Stdoutput = "message.txt";
    StdError = "error.txt";
    OutputSandbox = {"message.txt","error.txt"};
```

We specify:
- The program to run : /bin/echo
- Its arguments : Hello World
- Where to save the result of the program : message.txt
- Where to save the errors of our program, if there are : error.txt

---

Important: *The JDL is sensitive to blank characters and tabs. No blank characters or tabs should follow the semicolon at the end of a line.*

This is close to the minimum job description to be able to run a job on the Grid. The Executable in this case is a simple Unix echo command and the Argument to this command is the “Hello World” text. You have to specify at least two files: an output file and a file where the possible error messages go. The OutputSandbox contains the files that will go with the job to the node where the program will be executed and migrated back to the user when the execution has finished.

4.1.2
JDL example:

```
VirtualOrganisation = "betest";         # the VO you belong to
JobType = "normal";                     # Can be “normal”, “mpich”, “Interactive”,
                                        # “Checkpointable”
Type = "Job";                           # this is a job :-)
Executable = "yourexecutablefile";      # name of the command without arguments
Arguments = "arg1 arg2";                # arguments if necessary
StdOutput = "output";                   # the standard output (display) in the file we call output
StdError = "error";                     # to put the standard error in the file we call error
InputSandbox = "/bin/cat";              # to send the binary (or the script) and all necessary input
                                        # files
OutputSandbox = { "output","error" };# to receive the output files
```

In LCG-2, job description files (.jdl files) are used to describe jobs for execution on the Grid. These files are written using a Job Description Language (JDL). The JDL is used in LCG-2 to specify the desired job characteristics and constraints, which are used by the match-making process to select the resources that the job can use.
The JDL syntax consists on statements like:

```
attribute = value;
```

In a job description file, some attributes are mandatory, while some others are optional. Essentially, one must at least specify the name of the executable, the files where to write the standard output and the standard error of the job (they can even be the same file). For example:

```
Executable = "test.sh";
StdOutput = "std.out ";
StdError = "std.err ";
```

If needed, arguments can be passed to the executable:

```
Arguments = "hello 10";
```

Files to be transferred between the UI and the WN before (Input Sandbox) and after (Output Sandbox) the job execution can be specified:

```
InputSandbox = {" test.sh","std.in "};
OutputSandbox = {" std.out","std.err "};
```

Wildcards are allowed only in the InputSandbox attribute. The list of files in the Input Sandbox is specified relatively to the current working directory. Absolute paths cannot be specified in the Output-Sandbox attribute. Neither the Input Sandbox nor the Output Sandbox lists can contain two files with the same name (even if in different paths) as when transferred they would overwrite each other.

The environment of the job can be modified using the Environment attribute.

```
Environment = {"BETEST_PATH=$HOME/betest",
"BETEST_DB=$BETEST_PATH/betestdb "};
```

To express any kind of requirement on the resources where the job can run, there is the Requirements attribute. Its value is a Boolean expression that must evaluate to true for a job to run on that specific CE.

To send a job to a WN running Linux and with at least 128 MB of RAM:

```
Requirements = (other.GlueHostOperatingSystemRelease == "LINUX")
```

```
            && (other.GlueHostMainMemoryRAMSize >= 128);
```

To run on a CE using PBS as the LRMS, whose WNs have at least two CPUs and the job can run for at least two hours (we indicate the time in seconds) then in the job description file one could put:

```
Requirements = other. GlueCEInfoLRMSType == "PBS" &&
other. GlueCEInfoTotalCPUs > 1 &&
other. GlueCEPolicyMaxCPUTime > 7200;
```

To send a job to a particular CE we use the following expression:

```
Requirements = other. GlueCEUniqueID ==
" gridce.vliz.be:2119/jobmanager-lcgpbs-betest";
```

4.2 Finding a Computing Element for the job

4.2.2
The command “glite-wms-job-list-match” returns the CEs where the job could run.

```
glite-wms-job-list-match -a  HelloWorld.jdl
```

Important: *The user must be member of a VO.*

You should receive a list of CEs (Computer Element) that match with your job like this one:

```
Connecting to the service https://wms.begrid.be:7443/glite_wms_wmproxy_server

==========================================================================

                    COMPUTING ELEMENT IDs LIST
 The following CE(s) matching your job requirements have been found:

       *CEId*
 - gridce.atlantis.ugent.be:2119/jobmanager-pbs-betest
 - gridce.iihe.ac.be:2119/jobmanager-pbs-betest
 - gridce01.vliz.be:2119/jobmanager-pbs-betest
 - hwepc119.vub.ac.be:2119/jobmanager-lcgpbs-betest
 - kg-ce01.cc.kuleuven.be:2119/jobmanager-pbs-betest
 - ce01.cmi.ua.ac.be:2119/jobmanager-pbs-betest
 - ce01.grid.hogent.be:2119/jobmanager-pbs-betest

==========================================================================
```

Our job will be sent to one of those CEs.

The CE is identified by <CE>, which is a string with the following format:
<full_hostname >:< port_number >/ jobmanager -<service >-<queue\name >

Where:

- <full hostname> and <port> are the hostname of the machine and the port where the Globus Gatekeeper is running.
- <queue name> is the name of one of the queue of jobs available in that CE.
- <service> could refer to the LRMS, such as (lsf, pbs, condor), but can also be a different string as it is freely set by the site administrator when the queue is set-up.

4.3 Send the job to be executed

```
glite-wms-job-submit -a HelloWorld.jdl
```

If the submission is successful, the output is similar to:

```
Connecting to the service https://wms.begrid.be:7443/glite_wms_wmproxy_server


===================== glite-wms-job-submit Success ======================

The job has been successfully submitted to the WMProxy
Your job identifier is:

https://wms.begrid.be:9000/iKRnHi0eSnXIMYdAb0QWdg

==========================================================================
```

The command returns to the user the Job Identifier (JobId), which defines uniquely the job and can be used to perform further operations on the job, like interrogating the system about its status, or cancelling it. The format of the jobId is:

```
https :// Lbserver_address [: port ]/unique_string
```

```
https://wms.begrid.be:9000/5w5jQc7OIhBnFPO5g6PoLQ
^^^^^^
```
The https here does not identify a secure webpage

After a job is submitted, it is possible to see its status and its history, and to retrieve logging information about it. Once the job is finished the jobs output can be retrieved, although it is also possible to cancel it previously (we will see later how to do this).

Note: *Write down the JobId, or better copy it into your clipboard and paste it into a file on your PC, otherwise you won't be able to check the status of your job and even worst to retrieve its results!*

4.4 Get logging information about the current status of your Job

We can see at each moment the current status of the job, along with the time when that status was reached, and the reason for being in that state (which may be especially helpful for the ABORTED state).

```
glite-wms-job-status https://wms.begrid.be:9000/iKRnHi0eSnXIMYdAb0QWdg
```

You have to indicate the JobId that you have received in step 4.3.

```
*************************************************************
BOOKKEEPING INFORMATION:

Status info for the Job : https://wms.begrid.be:9000/iKRnHi0eSnXIMYdAb0QWdg
Current Status:     Done (Success)
Exit code:          0
Status Reason:      Job terminated successfully
Destination:        kg-ce01.cc.kuleuven.be:2119/jobmanager-pbs-betest
Submitted:          Wed Oct 29 16:08:53 2008 CET
*************************************************************
```

The destination field contains the ID of the CE where the job has been submitted. In this case our job has been sent to the CE: gridce01.vliz.be, but in your case it may be sent to another CE.

Check the different steps that the job follows.
Execute the command "glite-wms-job-status" to check them.

The possible job states are:

Status	Definition
SUBMITTED	The job has been submitted by the user but not yet processed by the Network Server
WAITING	The job has been accepted by the Network Server but not yet processed by the Workload Manager
READY	The job has been assigned to a Computing Element but not yet transferred to it
SCHEDULED	The job is waiting in the Computing Element's queue
RUNNING	The job is running
DONE	The job has finished
ABORTED	The job has been aborted
CANCELED	The job has been canceled by the user

CLEARED	The Output Sandbox has been transferred to the User Interface

4.5 Retrieve the output/result of your job

After our job has finished ("Current Status: Done (Success)"), its output can be copied to the UI with the command glite-wms-job-output.

Note: *You can only retrieve the results once the job has finished.*

```
************************************************************
glite-wms-job-output https://wms.begrid.be:9000/iKRnHi0eSnXIMYdAb0QWdg

Connecting to the service https://wms.begrid.be:7443/glite_wms_wmproxy_server


Warning - JobPurging not allowed
 (The Operation is not allowed: Unable to complete job purge)



================================================================================

                    JOB GET OUTPUT OUTCOME

Output sandbox files for the job:
https://wms.begrid.be:9000/iKRnHi0eSnXIMYdAb0QWdg
have been successfully retrieved and stored in the directory:
/tmp/abulanza_iKRnHi0eSnXIMYdAb0QWdg

================================================================================
```

Note that the warning message is a bug for WMS in SLC4.

The system transfers back the output from the WN to the WMS, and then to the UI.

```
ls /tmp/abulanza_e7E9E2XafzjjQNZw6QvKzg/
toknowhost.err  toknowhost.out
```

Verify that the output file is in the corresponding local tempory directory on the User Interface and that no errors have occurred.

```
ls /tmp/abulanza_iKRnHi0eSnXIMYdAb0QWdg
error.txt  message.txt
more /tmp/abulanza_iKRnHi0eSnXIMYdAb0QWdg/message.txt
```

Where “Gridusername_wK15luBpsZfsqjs6ce3img” is the JobId of your job.

The result must be

```
Hello World
```

The file error.txt file should be empty.

4.6 Options when sending jobs

4.6.1 Using a file to save the JobId

Very often you do not want to submit just one job but a whole series of jobs. This exercise will show how to allow you to look up the status of any of the jobs you submitted or to retrieve the output of any of those jobs without having to remember the JobId’s of each individual job that was submitted.
Instead of “remembering” the job identifier JobId each time, you will saved it in a file with the -o option in the submit command.

You can choose the name of the file that you want. For example “followid”.

```
glite-wms-job-submit –a -o followid HelloWorld.jdl
```

Verify that the JobId is stored on the “followid” file.

```
more followid
```

Check the status of your job and retrieve results once it has finished.

```
glite-wms-job-status -i followid

glite-wms-job-output -i followid
```

Note: *Don’t forget to delete the JobId entry on the file or the file once the job has finished. You can store more than one JobId on the file.*

4.6.1.1
Execute twice the “HelloWord.jdl” job saving JobIds on a file.

4.6.1.2
Check the status of the first job.

```
glite-wms-job-status -i followid

------------------------------------------------------------------
1 : https://wms.begrid.be:9000/8Qm1cNj7494upNBqjoz3XQ
2 : https://wms.begrid.be:9000/AmLczYHmv72qvMYnYri5dQ
a : all
q : quit
------------------------------------------------------------------

Choose one or more jobId(s) in the list - [1-2]all:1



*************************************************************
BOOKKEEPING INFORMATION:

Status info for the Job : https://wms.begrid.be:9000/8Qm1cNj7494upNBqjoz3XQ
Current Status:     Done (Success)
Exit code:          0
Status Reason:      Job terminated successfully
Destination:        gridce01.vliz.be:2119/jobmanager-pbs-betest
Submitted:          Wed Oct 29 16:19:42 2008 CET
*************************************************************
```

4.6.1.3
Checking the status of both jobs at the same time.

```
glite-wms-job-status -i followid

------------------------------------------------------------------
1 : https://wms.begrid.be:9000/8Qm1cNj7494upNBqjoz3XQ
2 : https://wms.begrid.be:9000/AmLczYHmv72qvMYnYri5dQ
a : all
q : quit
------------------------------------------------------------------

Choose one or more jobId(s) in the list - [1-2]all:a



*************************************************************
BOOKKEEPING INFORMATION:

Status info for the Job : https://wms.begrid.be:9000/8Qm1cNj7494upNBqjoz3XQ
Current Status:     Done (Success)
Exit code:          0
Status Reason:      Job terminated successfully
Destination:        gridce01.vliz.be:2119/jobmanager-pbs-betest
Submitted:          Wed Oct 29 16:19:42 2008 CET
*************************************************************



*************************************************************
BOOKKEEPING INFORMATION:
```

```
Status info for the Job : https://wms.begrid.be:9000/AmLczYHmv72qvMYnYri5dQ
Current Status:     Running
Status Reason:      Job successfully submitted to Globus
Destination:        gridce01.vliz.be:2119/jobmanager-pbs-betest
Submitted:          Wed Oct 29 16:22:35 2008 CET
*************************************************************
```

4.6.1.4
Retrieve results.

```
glite-wms-job-output -i followid
```

4.6.1.5
Verify the results.

4.6.2 Send the job to a specific CE.

In this exercise you will run the same HelloWorld job but now on a pre-selected site. From the list of Computing Elements that could run your job you can select one and use one of the options of the job submission command to send your job to that site.

4.6.2.1 Verify the available CEs for your job.

```
glite-wms-job-list-match -a  HelloWorld.jdl
```

4.6.2.2
Choose one and send the job to it.

```
glite-wms-job-submit -a -r hwepc119.vub.ac.be:2119/jobmanager-lcgpbs-betest  -o followid
HelloWorld.jdl
```

Remember we specify:
- With “-r” option the CE to use.
- With “-o” the file where the JobID will be saved.

4.6.2.3
Check the status of the job.

```
glite-wms-job-status -i filename

or

glite-wms-job-status JobId
```

4.6.2.4
Retrieve the output.

```
glite-wms-job-output --dir outputdir -i filename
```

4.6.2.5
Check the output.

4.6.2.6
Send the job again to a different CE.
Check its status.
Retrieve results.

4.7 Using an executable {1/2}

For this exercise we will use knowhost.jdl and knowhost.sh files

Change the directory to the directory Exercise2.

```
cd ..
cd exercise2
```

Verify that both files are on your directory!!

```
ls
```

4.7.1
Check the content of both files.

```
more knowhost.sh
```

The result should be:

```
#!/bin/sh
echo "Job has been executed on the WN: `hostname -f`"
```

```
more knowhost.jdl
```

The result should be:

```
Executable = "knowhost.sh";
StdOutput = "knowhost.out";
StdError = "knowhost.err";
InputSandbox = {"knowhost.sh"};
```

```
OutputSandbox = {"knowhost.out","knowhost.err"};
```

4.7.2
Verify in which CEs you can send your job.

```
glite-wms-job-list-match -a knowhost.jdl
```

4.7.3
Submit the job to a specific CE and save the JobId on a file.

4.7.4
Check the status of your job.

```
glite-wms-job-status -i filename

or

glite-wms-job-status JobId
```

4.7.5
Retrieve the output with the --dir command.

```
glite-wms-job-output --dir outputdir JobId

or

glite-wms-job-output --dir outputdir -i filename
```

Outputdir is the directory where you want to retrieve and save your results. Note that the directory must exist. If necessary create first that directoty (mkdir …).
Filename is the file containing the job identifier.

4.7.6
Verify that the job has been executed on the site that you have chosen.

You can check the result file and verify that the WN belongs to the same domain that the CE you have chosen.

```
CE: gridce.atlantis.ugent.be

WN: worker68.atlantis.ugent.be

The domain is atlantis.ugent.be
```

4.7.7
Verify the output file of your job.

4.7.8
Verify that the error file of your job is empty.

4.8 Small cascade of jobs

For this exercise we will use HelloWorld.jdl and submitter.sh files

4.8.1
Change the directory to Exercise3.

4.8.2
Verify content of the files.

HelloWorld.jdl

```
Executable = "/bin/echo";
Arguments = "Hello World";
Stdoutput = "message.txt";
StdError = "stderror";
OutputSandbox = {"message.txt","stderror"};
```

submitter.sh

```
#!/bin/bash
 i=0
 while [ $i -lt $1 ]
    do  glite-wms-job-submit -a -c wms-test.conf -o $2 HelloWorld.jdl
    i=`expr $i + 1`
 done
```

Look at the submitter.sh script and note that in this case we don't submit this script to the Grid directly but instead run this script locally and it submits jobs to the Grid.

Note that the script takes two parameters:

$1 The number of times that you want to submit the "HelloWorld" job.
$2 The file where JobIds will be stored.

4.8.3
Execute the script

```
./submitter.sh 3 /foldername/jobid.txt
```

We specify:
- With "3" that we want to send three times the "HelloWorld" job. (argument $1)
- With "/foldername/" the full path where our file will be saved. (argument $2)
- With "jobid.txt" the name of the file. (argument $2)

The command will send three times the HelloWorld.jdl script and will save the JobId on the jobid.txt file.

4.8.4
Verify the job status.

4.8.5
Retrieve the output and save it on your directory.

4.8.6
Check results.

4.9 Specifying Job Requirements {1}

For this exercise we will use pinger.jdl and pinger.sh files

In this exercise we ping a host from a Worker Node, to exercise the execution of simple operating system commands on the nodes.

4.9.1
Change the directory to the exercise4 directory.
Verify that both files are on your directory!!

4.9.2
Check the content of both files.

```
more pinger.sh
```

The result should be:

```
#!/bin/sh
/bin/ping -c 6 $1
```

```
more pinger.jdl
```

The result should be:

```
Executable = "pinger.sh";
Arguments = "www.google.com";
RetryCount = 7;
Stdoutput = "pingmessage1.txt";
StdError = "stderror";
InputSandbox = "pinger.sh";
OutputSandbox = {"pingmessage1.txt","stderror"};
Requirements = other.CEId=="gridce.iihe.ac.be:2119/jobmanager-pbs-betest";
```

We specify:

- With “Requirements = other.CEId” the CE where we want to send the job.

4.9.3
Submit the job to the grid and save the JobId on a file.

Note: *Depending on the installation ping ends by itself or has to be stopped with Ctrl-C.*

4.9.4
Check the status of your job.
Verify that the job has been submitted to the specified CE.

4.9.5
Retrieve the output with the --dir command.

```
glite-wms-job-output --dir outputdir JobId
```

4.9.6
Check the results.

4.10 Specifying Job Requirements {2}

For this exercise we will use pinger.jdl, pinger1.jdl, pinger2.jdl and pinger.sh files

Directory: Exercise4. Verify that all files are on your directory!!

4.10.1
Check the content of all files.

```
more pinger1.jdl
```

The result must be:

```
Executable = "/bin/bash";
Arguments = "pinger.sh www.google.com";
RetryCount = 7;
Stdoutput = "pingmessage1.txt";
StdError = "stderror";
InputSandbox = "pinger.sh";
OutputSandbox = {"pingmessage1.txt","stderror"};
Requirements = other.Architecture= =“INTEL” && other.OpSys= =“LINUX” && other.FreeCpus
>=2;
```

We specify:

- With “other.Architecture” the architecture where we want to run our job.
- With “other.OpSys” the operating system.
- With “other.FreeCpus” the number of free CPUs needed to execute our job.

- With “&&” we indicate “and”.

```
more pinger2.jdl
```

The result must be:

```
Executable = "/bin/bash";
Arguments = "pinger.sh www.google.com";
RetryCount = 7;
Stdoutput = "pingmessage1.txt";
StdError = "stderror";
InputSandbox = "pinger.sh";
OutputSandbox = {"pingmessage1.txt","stderror"};
Requirements = other.GlueHostMainMemoryRAMSize >= 512;
```

4.10.2
Execute the job pinger1.jdl

```
glite-wms-job-submit -a  pinger1.jdl
```

You should receive a JobId and your job should be aborted.

```
*************************************************************
BOOKKEEPING INFORMATION:

Status info for the Job : https://wms.begrid.be:9000/W4OiktURtw1CdMw_okY48Q
Current Status:     Waiting
Status Reason:      BrokerHelper: no compatible resources
Submitted:          Thu Oct 30 22:44:50 2008 CET
*************************************************************
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

There are no resources that fill all your requirements.

4.10.3
Execute the job pinger2.jdl

```
glite-wms-job-submit -a  pinger2.jdl
```

4.10.4
Check the status of the job.

4.10.5
Retrieve the results.

4.10.6
Check results.

4.11 Specifying Job Requirements {3}

For this exercise you will create your own *.jdl file

4.11.1
Create a new file *.jdl

Make your own student.sh script in which you don't ping a host but executes some other commands like for example /bin/pwd or /usr/bin/who.

The JDL must have:
- Arguments.
- Requirements.

For creating this file you can use the VI editor on the UI or you can create the file with Wordpad under Windows and then transfer the file to the UI with WinSCP.

4.11.2
Submit this script to the grid.

4.11.3
Retrieve the results.
Make sure that the output you get back is what you expected.

4.11.4
Other requirements:

Other requirements that you can use
other.GlueCEInfoTotalCPUs
other.GlueCEPolicyMaxWallClockTime >= <job_wall_clock_time_in_mins>

To find the whole list of possible requirements:
lcg-info --list-attrs

4.12 Canceling a job

For this exercise you can use the *.jdl that you want

4.12.1
Submit a job.

4.12.2
Cancel the job.

```
glite-wms-job-cancel JobId
```

The command cancels a job previously submitted using glite-wms-job-submit.

Before cancellation, it prompts the user for confirmation.

4.12.3
Try to cancel the job in different status of the job.

Data Management

Exercise 5: Using the Storage Element (SE)

Goal: Have a minimum knowledge of how to copy, retrieve files, create replicas and create directories on the Grid.

Requisites:

- The User’s Certificate must be installed on the UI.
- A valid Proxy.

Practice:

Most commands are similar to Unix/Linux commands.

In a Grid environment, the data files are replicated, possibly on a temporary basis, to many different sites depending on where the data is needed. The users or applications do not need to know where the data is located. They use logical names for the files and the Data Management services are responsible for locating and accessing the data. Large data files are available on the Grid and known to other users only if they are stored on Storage Elements (SEs) and registered in the Replica Management System catalogues. In order to optimize data access and to introduce fault-tolerance and redundancy, data files can be replicated on the Grid. The Replica Manager (RM), the Replica Location Service (RLS) and the Replica Metadata Catalogue (RMC) are the tools available for performing these tasks. Only anonymous access to the data catalogues is supported: the user proxy is not used to control the access to them.

Note: The Input/Output Sandbox is a mechanism for transferring small data files needed to start the job or to check the final status over the Grid.

5.1 Directory management

5.1.1 Export the environment needed variables

Export the needed variables to use the LFC catalog:

```
export LFC_HOST=gridy8.begrid.be
export LCG_CATALOG_TYPE=lfc
```

Important: *Please pay attention to the spelling of the LFC_HOST and LCG_CATALOG. Note that the first one is LFC and the second one LCG!!*

5.1.2 Create a directory on the Storage Element (SE)

```
lfc-mkdir /grid/betest/yourname
```

We specify:
- With “/grid/betest/” the full path where your directory will be created.
- With “yourname” the name of the directory.

The directory will be created on the gridy8.begrid.be machine. As we have defined this machine as LFC_HOST in point 5.1.1.

Note: *All members of a given VO have read-write permissions under their directory.*

5.1.3 Verify that the directory has been created

```
lfc-ls /grid/betest/
```

You should see an entry with your name:

```
abulanza
bartvandender
bdemaersch
bgoossens
bmalengier
boulahfatinajim
yourname ←
dgoris
dries
…
```

lfc-ls: List contents of a directory.

List contents of the “grid” directory.

```
lfc-ls -l /grid
```

The result should be similar to:

```
drwxrwxr-x   7 root     102                      0 Oct 22 10:08 beapps
drwxrwxr-x   1 root     103                      0 Apr 05  2007 becms
drwxrwxr-x  49 root     101                      0 Jun 17 09:23 betest
drwxrwxr-x   0 root     104                      0 Oct 31  2007 cms
drwxrwxr-x   0 root     105                      0 Oct 31  2007 hone
```

In this case there is an entry for each different VO.
List the content of the “grid/betest/yourname” directory and its subdirectories (if they exist).

```
lfc-ls -l -R /grid/betest/yourname
```

The -R option is used to list recursive directories on the /grid/betest/yourname

If your own directory is empty you can list the recursive contents of the /grid/betest/abulanza directory.

The result should be similar to:

```
lfc-ls -l -R /grid/betest/Gridusername

/grid/betest/ Gridusername:
-rw-rw-r--   1 106      101                     20 Oct 23 16:48 file1
-rw-rw-r--   1 106      101                     20 Oct 23 17:08 file3
drwxrwxr-x   0 106      101                      0 Oct 23 15:01 test001
drwx------   0 106      101                      0 Oct 23 15:06 test002

/grid/betest/Gridusername/test001:

/grid/betest/Gridusername/test002:
```

5.1.4 Create a new directory inside your directory

```
lfc-mkdir /grid/betest/yourname/directoryname
```

Verify that the directory has been created.

If you executed now the lfc-ls with the “-l -R” options you should see your subdirectory.

```
lfc-ls -l -R /grid/betest/yourname
```

5.1.5 Change the name of a directory

```
lfc-rename /fullpath/directoryoldname /fullpath/directorynewname

lfc-rename /grid/betest/yourname/directoryname /grid/betest/yourname/directoryname2
```

Verify that you have correctly changed the name of your directory.
Use the lfc-ls command to verify it.

5.1.6 Check the rights of a directory

On a Linux/Unix environment files and directories have rights. Such as:

drwxrwxr-x

The first character identifies the file type:

-	Regular file
b	Block special file
c	Character special file
d	Directory
l	Symbolic link
n	Network file
p	FIFO
s	Socket

The next nine characters are in three groups of three; they describe the permissions on the file. The first group of three describes owner permissions; the second describes group permissions; the third describes other (or world) permissions. Characters that may appear are:

r	Permission to read file
w	Permission to write to file
x	Permission to execute file
a	Archive bit is on (file has not been backed up)
c	Compressed file
s	System file
h	Hidden file
t	Temporary file

5.1.6.1 Check the rights of your directory

You can use the lfc-ls with the -l option to check them.
The result should be similar to:

```
drwxrwxr-x   0 106      101                     0 Oct 23 15:01 directoryname2
```

Or

You can use the lfc-getacl command:

```
lfc-getacl /grid/betest/yourname/directoryname2
```

The result should be like:

```
# file: /grid/betest/yourname/directoryname2
# owner: /C=BE/O=BEGRID/OU=IIHE/OU=ULB-VUB/CN=your name
# group: betest
user::rwx
group::rwx              #effective:rwx
other::r-x
default:user::rwx
default:group::rwx
default:other::r-x
```

5.1.7 Change the rights of a directory

We will use the lfc-chmod command to change rights:

```
lfc-chmod 700 /grid/betest/yourname/directoryname2
```

Note: *symbolic modes like you can use with the chmod command in Linux/Unix, such as +rw, are not supported yet.*

We indicate:

- With 700 the rights in binary: 111 000 000
  - 111: are the owner rights, the owner has the right to read (1), write (1) and execute (1).
  - 000: other users have not the permission to read (0), write (0) and execute (0).

As this is not a Linux/Unix training we will not enter on the chmod details.

Verify that you have properly changed them with the lfc-ls command or lfc-getacl command.

The result with the lfc-ls command must be:

```
drwx------   0 106     101                    0 Oct 23 15:06 directoryname2
```

The result with the lfc-getacl command must be:

```
# file: /grid/betest/yourname/directoryname2
# owner: /C=BE/O=BEGRID/OU=IIHE/OU=ULB-VUB/CN=Your name
# group: betest
user::rwx
group::---              #effective:---
other::---
default:user::rwx
default:group::rwx
default:other::r-x
```

5.1.8 Adding/deleting metadata information

“lfc-setcomment” adds/replaces a comment associated with a file/directory in the LFC Catalog.

Note: *if the file/directory has already a comment associated, it will be replaced by the new one.*

```
lfc-setcomment path “comment”

lfc-setcomment /grid/betest/yourname/directoryname2 "a test"
```

5.1.8.1
Check the comment.

```
lfc-ls --comment /grid/betest/yourname/
```

5.1.8.2
Replace the associated comment of the directoryname2 by a new one and check that the change has been done properly.

5.1.8.3
Deleting a comment.

“lfc-delcomment” deletes a comment previously added.

Note: *If there is no comment associated with the file, this command returns the error “No such file or directory”.*

```
lfc-delcomment  /grid/betest/yourname/directoryname2
```

Check that the comment has been correctly deleted.

Note: *This command can also be used to add/delete metadata information in files.*

Important: *A full list of the lfc-* commands can be found in the Appendix.*
Almost all commands used for directories can also be used for files management.

5.2 File management

A file at a SE:

- Has a Globally Unique Identifier (GUID).
  All the replicas of a file will share the same GUID.
  guid:<40_bytes_unique_string>

  guid:19ed73h0-c422-11d7-a5l0-km3lk5a56e1a

- Can have several replicas at different sites, each having a different Physical File Name (PFN) of the general form:

  <sfn | srm>://<SE_hostname>/<some_string>

  Where the prefix will be sfn for files located in SEs without SRM interface and srm for SRM-managed SEs.

  sfn://<SE_hostname><SE_Accesspoint><VO_path><filename>

  sfn://computer1.begrid.be/flatfiles/SE00/betest/generated/19ed73h0-c422-11d7-a5l0-km3lk5a56e1a

- In order to locate a Grid accessible file, the human user will normally use a LFN. LFNs are usually more intuitive, human-readable strings, since they are allocated by the user as GUID aliases
  Can be given one or several Logical File Names (LFN) or User Alias, which can be used to refer to a file in the place of the GUID. We specify the name of the file but not where the file is stored.

  lfn:<anything_you_want>

The information is stored in the Replica Metadata Catalog (RMC) and the Replica Location Service (RLS).

5.2.1 Copy and replica of a file from the UI to your directory on the SE

5.2.1.1
Change directory to exercise5.

You should find two files:
filetocopy.txt
filetocopy2.txt

The content of the files is not relevant for the exercise.

5.2.1.2
Finding available Storage Elements for the VO.

```
#lcg-infosites --vo betest se

Avail Space(Kb) Used Space(Kb)  Type    SEs
----------------------------------------------------------
1399810000      13768          n.a     kg-se01.cc.kuleuven.be
1               1              n.a     kg-se01.cc.kuleuven.be
286390000       400126         n.a     gridse.atlantis.ugent.be
521045430       5417820599     n.a     maite.iihe.ac.be
521045430       5417820599     n.a     maite.iihe.ac.be
521045430       5417820599     n.a     maite.iihe.ac.be
```

5.2.1.2
Upload a file to the SE and register it into the catalog.

```
lcg-cr -v --vo betest file:`pwd`/filetocopy.txt -l lfn:/grid/betest/yourname/thefile -d
gridse.atlantis.ugent.be
```

We specify:
- With "-v" the verbose mode. More information.
- With "file:`pwd`/filetocopy.txt" the source file [with " ` " and not " ' "].
- With "-l lfn:/grid/betest/yourname/thefile" the directory and the destination file.
- With "-d gridce.atlantis.ugent.be" the SE.

Note: *The destination directory is already created in the catalog.*

The result should be similar to:

```
Using grid catalog type: lfc
Using grid catalog : gridy8.begrid.be
Using LFN : /grid/betest/yourname/thefile.txt
Using SURL : srm://maite.iihe.ac.be/pnfs/iihe//betest/generated/2008-10-31/file352c6964-9199-
46f7-972a-81d15adcbb82
Alias registered in Catalog: lfn:/grid/betest/yourname/thefile.txt
Source URL: file:/home/yourname/exercise5/filetocopy.txt
File size: 20
VO name: betest
Destination specified: maite.iihe.ac.be
Destination URL for copy: gsiftp://behar2.iihe.ac.be:2811//pnfs/iihe/betest/generated/2008-10-
31/file352c6964-9199-46f7-972a-81d15adcbb82
# streams: 1
# set timeout to 0 seconds
        20 bytes      0.05 KB/sec avg      0.05 KB/sec inst
Transfer took 1010 ms
Destination URL registered in Catalog: srm://maite.iihe.ac.be/pnfs/iihe//betest/generated/2008-
10-31/file352c6964-9199-46f7-972a-81d15adcbb82
guid:aafaf209-3db5-4e6f-b509-1d82946d1376   ←
```

Write down or copy the guid as you will need it in point 5.2.2.1.

5.2.2 Download a file from a SE to your own directory in the UI

```
lcg-cp -v --vo betest lfn:/grid/betest/yourname/thefile file:`pwd`/nameofthesestinationfile

example:
lcg-cp -v --vo betest lfn:/grid/betest/gridusername/ ultimate1.txt file:`pwd`/ newfile1
```

The result of the command should be similar to:

```
Using grid catalog type: lfc
Using grid catalog : gridy8.begrid.be
VO name: betest
Source URL: lfn:/grid/betest/yourname/ultimate1.txt
File size: 20
Source URL for copy: gsiftp://behar021.iihe.ac.be:2811//pnfs/iihe/betest/generated/2008-10-
31/file352c6964-9199-46f7-972a-81d15adcbb82
Destination URL: file:/home/yourname/exercise5/newfile1
# streams: 1
# set timeout to  0 (seconds)
          0 bytes      0.00 KB/sec avg      0.00 KB/sec inst
Transfer took 1020 ms
```

Check that the file has been properly copied.

5.2.2 Replicas

5.2.2.1
Obtaining LFN from GUID.

```
lcg-la --vo betest guid:guidnumber
```

Where guidnumber is the GUID your have received in the point 5.2.1.2.

For example:

```
lcg-la --vo betest guid:aafaf209-3db5-4e6f-b509-1d82946d1376
```

The result should be similar to:

```
lfn:/grid/betest/yourname/ultimate1.txt
```

Instead of yourname you must have your name directory and instead of file2 the name of the file you registered.

5.2.2.2
Obtaining SFN from LFN.

```
lcg-lr --vo betest lfn:/grid/betest/yourname/fultimate1.txt
```

Instead of yourname you must have your name and instead of file3 the name of the file you registered.

The result should be something like this:

```
srm://maite.iihe.ac.be/pnfs/iihe//betest/generated/2008-10-31/file352c6964-9199-46f7-972a-
81d15adcbb82
```

5.2.2.3
Copy a file from one Storage Element to another Storage Element and registers it in the LFC.

```
lcg-rep -d kg-se01.cc.kuleuven.be -v --vo betest lfn:/grid/betest/yourname/yourfile
```

The result should be similar to:

```
Using grid catalog type: lfc
Using grid catalog : gridy8.begrid.be
Source URL: lfn:/grid/betest/abulanza/ultimate1.txt
File size: 20
VO name: betest
Destination specified: kg-se01.cc.kuleuven.be
Source URL for copy: gsiftp://behar020.iihe.ac.be:2811//pnfs/iihe/betest/generated/2008-10-
31/file352c6964-9199-46f7-972a-81d15adcbb82
Destination URL for copy: gsiftp://kg-se01.cc.kuleuven.be/kg-
se01.cc.kuleuven.be:/storage2/betest/2008-10-31/file9375dd8a-e24d-4f0b-9d82-
3d0fc2bd6cf0.40761.0
# streams: 1
# set timeout to 0
        20 bytes      0.01 KB/sec avg      0.01 KB/sec inst
Transfer took 4270 ms
Destination URL registered in LRC: srm://kg-
se01.cc.kuleuven.be/dpm/cc.kuleuven.be/home/betest/generated/2008-10-31/file9375dd8a-
e24d-4f0b-9d82-3d0fc2bd6cf0
```

It copies the “test” file from the “maite.iihe.ac.be” SE to the “kg-se01.cc.kuleuven.be” SE.

You can use the lcg-infosites command to check for available SE.

```
lcg-infosites --vo betest se
```

The result is similar to:

```
****************************************************************
These are the available SEs for betest:
****************************************************************

lcg-infosites --vo betest se
Avail Space(Kb) Used Space(Kb)  Type    SEs
----------------------------------------------------------
1399810000      13768          n.a     kg-se01.cc.kuleuven.be
1               1              n.a     kg-se01.cc.kuleuven.be
286390000       400126         n.a     gridse.atlantis.ugent.be
521045430       5417820599     n.a     maite.iihe.ac.be
521045430       5417820599     n.a     maite.iihe.ac.be
521045430       5417820599     n.a     maite.iihe.ac.be
853883544       61922744       n.a     gridce.vliz.be
28294528        42700756       n.a     se01.cmi.ua.ac.be
```

Note: *The lcg-infosites command can also to be used to print the list of available CEs:*

```
lcg-infosites --vo betest ce
```

5.2.2.4
List all known replicas that are associated with a particular LFN.

```
lcg-lr lfn:/grid/betest/yourname/thefile --vo betest
```

You must have as output the list of replicas with two results that looks like:

```
srm://kg-se01.cc.kuleuven.be/dpm/cc.kuleuven.be/home/betest/generated/2008-10-
31/file9375dd8a-e24d-4f0b-9d82-3d0fc2bd6cf0
srm://maite.iihe.ac.be/pnfs/iihe//betest/generated/2008-10-31/file352c6964-9199-46f7-972a-
81d15adcbb82
```

5.2.2.5
Delete replicas.

Delete one replica.

```
lcg-del -v --vo betest srm://maite.iihe.ac.be/pnfs/iihe//betest/generated/2008-10-31/file352c6964-
9199-46f7-972a-81d15adcbb82
```

You must get as result:

```
VO name: betest
Using GUID : aafaf209-3db5-4e6f-b509-1d82946d1376
set timeout to 0 seconds
```

Verify that the replica has been deleted.

```
lcg-lr --vo betest lfn:/grid/betest/yourname/thefile
```

You should get only one replica!!

```
srm://kg-se01.cc.kuleuven.be/dpm/cc.kuleuven.be/home/betest/generated/2008-10-
31/file9375dd8a-e24d-4f0b-9d82-3d0fc2bd6cf0
```

Delete the other replica:

```
lcg-del -v --vo betest -a lfn:/grid/betest/yourname/thefile
```

The result should be:

```
VO name: betest
Using GUID : aafaf209-3db5-4e6f-b509-1d82946d1376
set timeout to 0 seconds
srm://kg-se01.cc.kuleuven.be/dpm/cc.kuleuven.be/home/betest/generated/2008-10-
31/file9375dd8a-e24d-4f0b-9d82-3d0fc2bd6cf0 is deleted
srm://kg-se01.cc.kuleuven.be/dpm/cc.kuleuven.be/home/betest/generated/2008-10-
31/file9375dd8a-e24d-4f0b-9d82-3d0fc2bd6cf0 is unregistered
```

Verify that the replica has been deleted.

```
lcg-lr --vo betest lfn:/grid/betest/yourname/thefile
```

The result must be:

```
gridy8.begrid.be: /grid/betest/yourname/ultimate1.txt: No such file or directory
lcg_lr: No such file or directory
```

Note: *You can use the guid or the lfn identifier to delete your file.*
If you delete the guid, the lfn and replicas will be deleted automatically.
If you delete the lfn, the guid and replicas will be deleted too.

For example:

```
lcg-del -v --vo betest -a guid:b7c5548f-f65f-4d2e-b5d7-cc2d07d6eedc

lcg-del –v –vo betest –a lfn:/grid/betest/yourname/thefile
```

Examples

6.1 Using a data file as input of your program.

6.1.1The file is on the UI.

```
Executable    = "short.sh";
Arguments     = "none";
StdOutput     = "std.out";
StdError      = "std.err";
InputSandbox  = {"short.sh", "short.dat"};
OutputSandbox = {"std.out","std.err"};
```

We specify:
- With “short.sh” our executable.
- With “short.dat” the input data to the program.

Note that the short.dat file is on the UI.

6.1.2 The file is somewhere on the Grid.

```
Executable    = "short.sh";
StdOutput     = "std.out";
StdError      = "std.err";
InputSandbox  = {"short.sh"};
OutputSandbox = {"std.out","std.err"};
InputData = "lfn:youraliasfile";
DataAccessProtocol = "gridftp";
```

We specify:
- With InputData = "lfn:youraliasfile" the alias of your file.
- With DataAccessProtocol = "gridftp" the protocol to use.

We can also use the GUID to identify your data. In this case the job description file must contain a line like:

```
InputData = “guid:19f3-4a3d-846m47ud-3bk36b48a645-8gnn";
```

On the DataAccessProtocol option the only supported protocols are:

- File, meaning that the files must be in an NFS-mounted directory.
- Gridftp, the GSI version of ftp.
- Rfio, Remote File Input/Output.

6.2 Saving the output of the program on the Grid

```
Executable    = "short.sh";
StdOutput     = "std.out";
StdError      = "std.err";
InputSandbox  = {"short.sh"};
OutputSandbox = {"std.out","std.err"}
OutputData = {
 [
        Outputfile = “filename”;
        LogicalName = “lfn:logicalfilename”;
        StorageElement = “maite.iihe.ac.be”;
 ]};
```

The Outputfile parameter is mandatory.
The LogicalName and StorageElement parameters are optional.

Note: *when the result of the program is more than a few Mega, it must be saved on a SE!*

6.3 The job does the data management

The job retrieves a file previously registered into the catalog.

Script.sh

```
#!/bin/sh
/bin/hostname

#Change the LFN_NAME to download from the Catalog.
echo "Start downloading.."
lcg-cp --vo betest lfn:/grid/betest/yourname/<lfn you choose> file:`pwd`/output.dat
echo "Done.."
```

Script.jdl

```
Executable = "/bin/sh";
Arguments = "script.sh";
VirtualOrganisation = "betest";
StdOutput = "std.out";
StdError = "std.err";
InputSandbox = {"script.sh"};
OutputSandbox = {"std.out","std.err","output.dat"};
```

APPENDIX

List of lfc-* commands:

lfc-chmod	Change access mode of the LFC file/directory
lfc-chown	Change owner and group of the LFC file/directory
lfc-delcomment	Delete the comment associated with the file/directory
lfc-getacl	Get file/directory access control list
lfc-ln	Make a symbolic link to a file/directory
lfc-ls	List file/directory entries in a directory
lfc-mkdir	Create a directory
lfc-rename	Rename a file/directory
lfc-rm	Remove a file/directory
lfc-setacl	Set file/directory access control list
lfc-setcomment	Add/replace a comment

List of lcg-* commands:

lcg-cr	copy and register a file
lcg-lr	lists the replicas for a given LFN, GUID or SURL
lcg-lg	lists the GUID for a given LFN or SURL
lcg-rep	copy a file from one SE to another SE and registers it in the LRC
lcg-aa	add an alias in RMC for a given GUID
lcg-ra	remove an alias in RMC for a given GUID
lcg-rf	register in the LRC (and optionally in the RMC) a file residing on an SE
lcg-uf	unregister in the LRC a file residing on an SE
lcg-cp	copy a Grid file to a local destination
lcg-gt	get the TURL for a given SURL and transfer protocol
lcg-la	lists the aliases for a given LFN, GUID or SURL

APPENDIX

Overview of glite-wms-* commands

glite-wms-job-list-match	Matching computing elements
	With the proxy delegation options: Add -a or -d
glite-wms-job-delegate-proxy	Proxy delegation
	This is the preferred form of proxy delegation. It is used to create and store a named proxy in WMProxy, to be specified later by the -d option of glite-wms-job-submit. Alternatively, the -a option of glite-wms-job-submit will automatically create and store an unnamed proxy in WMProxy.
glite-wms-job-submit	Job submission
	With the proxy delegation options: Add -a or -d. Job collections can be submitted in bulk for much faster submission times, by putting all JDLs in a single directory and specifying that directory in the --collection jdldir option of glite-wms-job-submit. WMProxy will submit a DAG whose nodes are the jobs in the collection, without any dependencies, and will return to the user a DAGID, having exactly the same format as a jobID.
glite-wms-job-status	Job status
glite-wms-job-get-logging-info	Job logging
glite-wms-job-perusal	Job perusal
glite-wms-job-cancel	Job cancellation
glite-wms-job-output	Job retrieval

References

BEgrid Hands-on, Version 0
June 2006, Ines Martinez Moreno and Rosette Vandenbroucke, BELNET

LHC Computing Grid Project
http://lcg.web.cern.ch/LCG/

GLite documentation
http://glite.web.cern.ch/glite/documentation/
https://edms.cern.ch/file/722398//gLite-3-UserGuide.pdf

LCG user overview and user guide
http://lcg.web.cern.ch/LCG/peb/grid_deployment/user_intro.htm
https://edms.cern.ch/file/498081/1.0/UserScenario2.html
https://edms.cern.ch/file/495216/1/LCG-Faq.html

General User Guides
http://goc.grid.sinica.edu.tw/gocwiki/User_Guides

Job Submission and Data Management
http://grid.desy.de/users/

Job Description Language
http://server11.infn.it/workload-grid/docs/DataGrid-01-TEN-0102-0_2-Document.pdf
http://server11.infn.it/workload-grid/docs/DataGrid-01-TEN-0142-0_2.pdf

BEgrid
http://www.begrid.be/

BEGrid Certification Authority
http://quattor.begrid.be/trac/centralised-begrid-v5/wiki/BEgrid_CA_info

GridFTP
http://www.globus.org/grid_software/data/gridftp.php

Real job examples: http://alipc1.ct.infn.it/edg-tutorial/exercises/

GLite-WMS and gLite-LB
https://wiki.gridpp.ac.uk/wiki/GLite-WMS_and_glite-LB