

# Open Science Grid

## Engage Virtual Organization

### New User Guide

Mats Rynge, RENCi

<rynge@renci.org>

Revision 1.2 – September 28, 2007

### Table of Contents

1. Overview.....	2
2. Getting a Certificate.....	2
3. Joining the Engage VO.....	3
4. Submit Host.....	3
5. Initializing a User Proxy.....	3
6. Submitting a Job.....	4
6.1. Condor.....	5
6.2. globus-job-run.....	7
7. Managing Condor Jobs.....	7
7.1. condor_grid_overview.....	7
7.2. condor_q / condor_status.....	8
7.3. condor_rm.....	8

## 1. Overview

The Engage VO has been established to help new communities make use of Open Science Grid. For any questions or comments, please email Engage support at [osg@renci.org](mailto:osg@renci.org)

The purpose of this document is to provide some quick notes on things you need to do to run jobs on OSG as a member of the Engage VO (Virtual Organization). If you need more detailed information than this document provides, see the OSG documentation.

Getting Started guide:

[http://www.opensciencegrid.org/Getting\\_Started](http://www.opensciencegrid.org/Getting_Started)

End User documentation:

<https://twiki.grid.iu.edu/twiki/bin/view/Documentation/UserDocEndUser>

## 2. Getting a Certificate

Open Science Grid authentication is a single-sign on system based on X.509 certificates<sup>1</sup>. In order to submit jobs you need a certificate. To apply for one, use a web browser (Mozilla Firefox or Internet Explorer), and go to:

<https://pki1.doegrids.org/>

Fill out the form. For Affiliation, select 'OSG', and for VO name, select 'Engage'. Please contact the Engage team at [osg@renci.org](mailto:osg@renci.org) for the sponsor name, e-mail address, and phone number as required by the certificate request form.

It will take a day or two for the certificate to be approved. Once you receive the email about the certificate being ready, follow these instructions on how to export the certificate to a Grid public/private key pair:

<http://www.doegrids.org/pages/cert-request.html>

Under the “Exporting your key pair for use by Globus grid-proxy-init” heading.

---

<sup>1</sup> <http://www.globus.org/security/overview.html>

### 3. Joining the Engage VO

Once you have your certificate, you can join the Engage Virtual Organization (VO). Go to:

<https://osg-engage.renci.org:8443/voms/Engage/webui/request/user/create>

The browser should present your certificate to the server, and you should see something like this on the top of the form (but with your name):

```
DN: /DC=org/DC=doegrids/OU=People/CN=Mats Rynge 722233
```

Fill out and submit the form. You will get a confirmation email, which asks you to click a link to verify your email address. Once that is done, in less than a day, you should receive an email stating that you are now part of the Engage VO.

### 4. Submit Host

The submit host is the machine managing your jobs. RENCi has a submit node for you to use with all of the client software pre-installed and configured. You could run your own, but until you are more familiar with the tools and how they work together, it is recommended to use RENCi's submit host. The current host is [nantahala.renci.org](http://nantahala.renci.org). To get an account on this host, please email [rynge@renci.org](mailto:rynge@renci.org) for instructions.

The most important step is to set up your environment. In `~/.bash_profile`, make sure you have a line reading:

```
. /opt/osg/current/setup.sh
```

If you are `csch` user, you should have the following in your `~/.cshrc`:

```
source /opt/osg/current/setup.csh
```

### 5. Initializing a User Proxy

Note that you will have to have your certificate exported as mentioned above, and placed in `~/.globus/` on the submit host. A proxy is a time limited version of your

certificate. The proxy is what is actually presented to the remote site during authentication. To create a proxy, use the voms-proxy-init command:

```
[rynge@nantahala ~]$ voms-proxy-init -voms Engage -valid 72:00
Cannot find file or dir: $prefix/etc/vomses
Your identity: /DC=org/DC=doe grids/OU=People/CN=Mats Ryng e 722233
Enter GRID pass phrase:
Creating temporary proxy ..... Done
Contacting osg-engage.renci.org:15001 [/DC=org/DC=doe grids/OU=Services/CN=osg-engage.renci.org] "Engage" Done

Creating proxy ..... Done
Your proxy is valid until Thu Apr 5 10:29:08 2007
```

Note that you can specify how long the proxy should be valid for. It should be long enough for the job run to finish, but should not be longer than 72 hours.

You might see some warnings in the output of voms-proxy-init. Do not worry about them as long as you get a proxy at the end. You can check the proxy with voms-proxy-info:

```
[rynge@nantahala ~]$ voms-proxy-info
WARNING: Unable to verify signature! Server certificate possibly not installed.
Error: Cannot find certificate of AC issuer for vo Engage
subject   : /DC=org/DC=doe grids/OU=People/CN=Mats Ryng e 722233/CN=proxy
issuer    : /DC=org/DC=doe grids/OU=People/CN=Mats Ryng e 722233
identity  : /DC=org/DC=doe grids/OU=People/CN=Mats Ryng e 722233
type      : proxy
strength  : 512 bits
path      : /tmp/x509up_u1031
timeleft  : 23:20:17
```

## 6. Submitting a Job

There are several tools that you can use to submit jobs, including Condor and Globus. For new users, we recommend using Condor as the primary submission mechanism.

## 6.1. Condor

Condor includes a match making system, which matches your jobs to compute resources and submits and manages the jobs. You will need to describe the job requirements and behavior in a job description file. Condor will use that information to submit your job to a remote resource, handle simple data transfers, and in case of a problem, move your job to another resource.

For detailed information about the submit file format, see the Condor documentation (especially the Condor-G section):

[http://www.cs.wisc.edu/condor/manual/v6.8.4/5\\_3Grid\\_Universe.html](http://www.cs.wisc.edu/condor/manual/v6.8.4/5_3Grid_Universe.html)

The Engagement VO is providing an example package, which can be downloaded from:

<http://osg-engage.renci.org/quickstart/generic-job-example.tar.gz>

The package contains three scripts:

- submit – creates the Condor submit files and a Condor DAG. The DAG is used to control the jobs, and check the stdout/stderr of the run to make sure the application ran successfully. Note that you can not trust the exit code as it sometimes is the exit code from the job scheduling system on the remote system, and not the exit code from your application. The example package is using makeself to create a self-extracting executable for the job.
- job-wrapper – wraps your executable. Tars inputs and outputs, and redirects stdout/stderr.
- job-success-check – Used by the DAG to check stdout/stderr to see if the job was successful. If the job failed, it will get resubmitted.

The Condor submit file created will look something like:

```

universe          = grid
grid_type         = gt2
globusscheduler  = $$ (GlueCEInfoContactString)
globusrs1        = (maxWallTime=900)
requirements     = (TARGET.GlueCEInfoContactString != UNDEFINED)

# when retrying, remember the last 4 resources tried
match_list_length = 4
Rank              = (TARGET.Rank) - ((TARGET.Name =?= LastMatchName0) * 1000) -
((TARGET.Name =?= LastMatchName1) * 1000) - ((TARGET.Name =?= LastMatchName2) *
1000) - ((TARGET.Name =?= LastMatchName3) * 1000)

# make sure the job is being retried and rematched
periodic_release = (NumGlobusSubmits < 10)
globusresubmit   = (NumSystemHolds >= NumJobMatches)
rematch          = True
globus_rematch   = True

# only allow for the job to be queued for a while, then try to move it
periodic_hold    = ( ((JobStatus==1) && ((CurrentTime - EnteredCurrentStatus) >
(4*60*60))) || ((JobStatus==2) && ((CurrentTime - EnteredCurrentStatus) >
(5*60*60))) )

# stay in queue on failures
on_exit_remove   = (ExitBySignal == False) && (ExitCode == 0)

executable       = /bin/date
arguments        = '-R'

WhenToTransferOutput = ON_EXIT
TransferExecutable = false

output           = job.out
error            = job.err
log              = job.log

notification     = NEVER

queue

```

Note:

- Rank and match\_list\_length together specifies that if the job has to be retried, it should be done on a resource which was not matched against the last 4 tries.
- periodic\_hold is an expression which specifies when a job should be put on hold. Once the job is in the held state, it can be released (see periodic\_release) and retired on another resource. In the example, we want the job to be held if it has been queued for a long time (JobStatus==1, 4 hours) or the job has been running for an excessive amount of time (JobStatus==2, 5 hours)

See chapter 7 for information on how to monitor jobs.

## **6.2. globus-job-run**

globus-job-run can be used to run quick jobs on resources, such as staging data. First, find the Globus resource contact by using 'condor\_status -l'. You are looking for either JobManager or GlueCEInfoContactString. In the string, replace jobmanager-\* with jobmanger-fork. The end result should look like:

```
fermigrid1.fnal.gov:2119/jobmanager-fork
```

Use that as the first argument to globus-job-run. The second argument is the executable with full path. Example:

```
globus-job-run fermigrid1.fnal.gov:2119/jobmanager-fork /bin/date
```

## **7. Managing Condor Jobs**

Assuming you are using Condor to manage your jobs, there are several commands for you to monitor the jobs.

### **7.1. condor\_grid\_overview**

The Engage VO provides a script which will give you a quick overview of jobs in the queue, and available resources. The command is 'condor\_grid\_overview' and below is an example output.

```
$ condor_grid_overview

ID      Owner      Resource      Status
===== =====
19364   huxz       FNAL_GPFARM   Running
19365   huxz       FNAL_GPFARM   Running
19371   huxz       CIT_CMS_T2    Running

Site                Jobs  Subm  Pend  Run  Stage
=====
CIT_CMS_T2          1    0    0    1    0
FNAL_FERMIGRID      0    0    0    0    0
FNAL_GPFARM         21   0    0    21   0
Nebraska            70   0    20   50   0
TTU-ANTAEUS         70   0    0    70   0

576 jobs; 475 idle, 96 running, 5 held
```

**7.2. condor\_q / condor\_status**

condor\_q and condor\_status give information about jobs and resources respectively. Both accept the -l flag to give a full set of data instead of summaries.

**7.3. condor\_rm**

To remove Condor jobs from the queue, use condor\_rm. It can either be used with a job id (from condor\_grid\_overview or condor\_q):

```
condor_rm 22436
```

or to remove all your jobs:

```
condor_rm -all
```