

# Getting started on Swegrid

Olav Vahtras

June 4, 2009

# Contents

## Introduction

## Getting started

Obtaining certificate

Test your installation

## Test jobs

Installed software

Gaussian

Molcas

Dalton

User programs

## About Swegrid

- ▶ Swegrid is a distributed computational resource provided by SNIC distributed over five different sites
  - ▶ Lunarc, Lund
  - ▶ UNICC, Göteborg
  - ▶ NSC, Linköping
  - ▶ Uppmax, Uppsala
  - ▶ HPC2N, Umeå
- ▶ Intended mostly for through-put computing, typically serial jobs in large numbers (data-parallelism)
- ▶ MPI jobs possible, but typically with a slow network

# How to access the grid

## Client software

- ▶ All interaction with Swegrid is done from your workstation

# How to access the grid

## Client software

- ▶ All interaction with Swegrid is done from your workstation
- ▶ You have a grid identity but no unique account on the clusters

# How to access the grid

## Client software

- ▶ All interaction with Swegrid is done from your workstation
- ▶ You have a grid identity but no unique account on the clusters
- ▶ Access to the cluster via grid middleware ARC (Advanced Resource Connector).

# How to access the grid

## Client software

- ▶ All interaction with Swegrid is done from your workstation
- ▶ You have a grid identity but no unique account on the clusters
- ▶ Access to the cluster via grid middleware ARC (Advanced Resource Connector).
- ▶ Can be viewed as high-level batch system

# How to access the grid

## Client software

- ▶ All interaction with Swegrid is done from your workstation
- ▶ You have a grid identity but no unique account on the clusters
- ▶ Access to the cluster via grid middleware ARC (Advanced Resource Connector).
- ▶ Can be viewed as high-level batch system
- ▶ Can be downloaded precompiled from <http://download.nordugrid.org>

# How to access the grid

## Client software

- ▶ All interaction with Swegrid is done from your workstation
- ▶ You have a grid identity but no unique account on the clusters
- ▶ Access to the cluster via grid middleware ARC (Advanced Resource Connector).
- ▶ Can be viewed as high-level batch system
- ▶ Can be downloaded precompiled from <http://download.nordugrid.org>
- ▶ Available under `/pkg/swegrid`

## Your very first grid command:grid-cert-request

- ▶ Initialize your environment (theochem)  
\$ `cd /pkg/swegrid/co4; . setup.sh` (CentOS4)
- ▶ Obtain a certificate for access to Swegrid  
\$ `grid-cert-request -int`  
Follow the instructions. Two files are generated

## Your very first grid command:grid-cert-request

- ▶ Initialize your environment (theochem)  
`$ cd /pkg/swegrid/co4; . setup.sh (CentOS4)`
- ▶ Obtain a certificate for access to Swegrid  
`$ grid-cert-request -int`  
Follow the instructions. Two files are generated
  - ▶ `userkey.pm` (private key)  
to be saved in directory `~/.globus`

## Your very first grid command:grid-cert-request

- ▶ Initialize your environment (theochem)  
`$ cd /pkg/swegrid/co4; . setup.sh (CentOS4)`
- ▶ Obtain a certificate for access to Swegrid  
`$ grid-cert-request -int`  
Follow the instructions. Two files are generated
  - ▶ `userkey.pm` (private key)  
to be saved in directory `~/.globus`
  - ▶ `usercert_request.pem`  
to be mailed to `ngca-ra@pdc.kth.se`

## Your very first grid command:grid-cert-request

- ▶ Initialize your environment (theochem)  
`$ cd /pkg/swegrid/co4; . setup.sh` (CentOS4)
- ▶ Obtain a certificate for access to Swegrid  
`$ grid-cert-request -int`  
Follow the instructions. Two files are generated
  - ▶ `userkey.pm` (private key)  
to be saved in directory `~/.globus`
  - ▶ `usercert_request.pem`  
to be mailed to `ngca-ra@pdc.kth.se`
- ▶ The file returned from the CA (after about a week) contains the signed certificate saved as `~/.globus/usercert.pem`

```
$ grid-cert-request -int
```

```
$ grid-cert-request -int
```

```
A certificate request and private key is being cr
You will be asked to enter a PEM pass phrase.
This pass phrase is akin to your account password
and is used to protect your key file.
If you forget your pass phrase, you will need to
obtain a new certificate.
```

```
Generating a 1024 bit RSA private key
```

```
.....+++++
.....+++++
```

```
writing new private key to '/home/vahtras/.globus
```

```
Enter PEM pass phrase:
```

```
$ grid-cert-request -int
```

```
A certificate request and private key is being cr
You will be asked to enter a PEM pass phrase.
This pass phrase is akin to your account password
and is used to protect your key file.
If you forget your pass phrase, you will need to
obtain a new certificate.
```

```
Generating a 1024 bit RSA private key
```

```
.....+++++
.....+++++
```

```
writing new private key to '/home/vahtras/.globus
```

```
Enter PEM pass phrase:XXXXXX
```

```
$ grid-cert-request -int
```

```
A certificate request and private key is being cr  
You will be asked to enter a PEM pass phrase.  
This pass phrase is akin to your account password  
and is used to protect your key file.  
If you forget your pass phrase, you will need to  
obtain a new certificate.
```

```
Generating a 1024 bit RSA private key
```

```
.....++++++  
.....++++++
```

```
writing new private key to '/home/vahtras/.globus
```

```
Enter PEM pass phrase:XXXXXX
```

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

```
$ grid-cert-request -int
```

```
A certificate request and private key is being cr
You will be asked to enter a PEM pass phrase.
This pass phrase is akin to your account password
and is used to protect your key file.
If you forget your pass phrase, you will need to
obtain a new certificate.
```

```
Generating a 1024 bit RSA private key
```

```
.....+++++
.....+++++
```

```
writing new private key to '/home/vahtras/.globus
```

```
Enter PEM pass phrase:XXXXXX
```

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

## grid-cert-request: User data

```
Level 0 Organization Name (do not modify) [Grid]:
```

## grid-cert-request: User data

```
Level 0 Organization Name (do not modify) [Grid]:
```

```
Level 1 Organization Name (do not modify) [NorduGrid]:
```

## grid-cert-request: User data

```
Level 0 Organization Name (do not modify) [Grid]:  
Level 1 Organization Name (do not modify) [Nordugrid]:  
Your Domain (e.g. example.org) []:
```

## grid-cert-request: User data

```
Level 0 Organization Name (do not modify) [Grid]:  
Level 1 Organization Name (do not modify) [NorduGrid]:  
Your Domain (e.g. example.org) []:biotech.kth.se
```

## grid-cert-request: User data

```
Level 0 Organization Name (do not modify) [Grid]:  
Level 1 Organization Name (do not modify) [NorduGrid]:  
Your Domain (e.g. example.org) []:biotech.kth.se  
Name (e.g., Hans Christian Andersen) []:
```

## grid-cert-request: User data

```
Level 0 Organization Name (do not modify) [Grid]:  
Level 1 Organization Name (do not modify) [NorduGrid]:  
Your Domain (e.g. example.org) []:biotech.kth.se  
Name (e.g., Hans Christian Andersen) []:First Last
```

## grid-cert-request: User data

```
Level 0 Organization Name (do not modify) [Grid]:  
Level 1 Organization Name (do not modify) [NorduGrid]:  
Your Domain (e.g. example.org) []:biotech.kth.se  
Name (e.g., Hans Christian Andersen) []:First Last  
Email address (e.g., h.c.andersen@example.org) []:
```

## grid-cert-request: User data

```
Level 0 Organization Name (do not modify) [Grid]:  
Level 1 Organization Name (do not modify) [NorduGrid]:  
Your Domain (e.g. example.org) []:biotech.kth.se  
Name (e.g., Hans Christian Andersen) []:First Last  
Email address (e.g., h.c.andersen@example.org) []:fl@theoche
```

## grid-cert-request: User data

```
Level 0 Organization Name (do not modify) [Grid]:  
Level 1 Organization Name (do not modify) [Nordugrid]:  
Your Domain (e.g. example.org) []:biotech.kth.se  
Name (e.g., Hans Christian Andersen) []:First Last  
Email address (e.g., h.c.andersen@example.org) []:fl@theoche
```

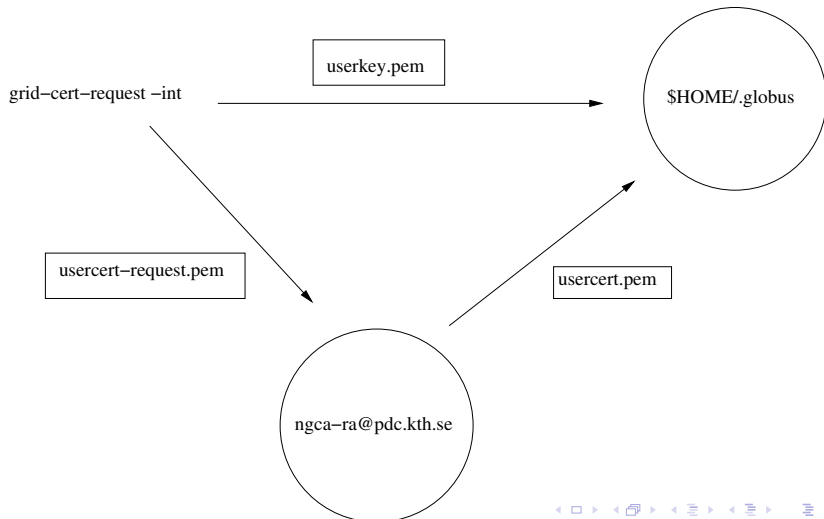
IMPORTANT      IMPORTANT      IMPORTANT

Please read the instructions in:

```
/home/vahtras/.globus/usercert_r
```

carefully in order to finalize and get your certi

# Grid certificate request process



# Log in to the grid

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

# Log in to the grid

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

```
Your identity: /O=Grid/O=NorduGrid/OU=cdc.kth.se/CN=Olav Va
```

## Log in to the grid

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

```
Your identity: /O=Grid/O=NorduGrid/OU=cdc.kth.se/CN=Olav Vahtras
```

```
Enter GRID pass phrase for this identity:
```

## Log in to the grid

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

```
Your identity: /O=Grid/O=NorduGrid/OU=pcd.kth.se/CN=Olav Va
```

```
Enter GRID pass phrase for this identity:XXXXXX
```

## Log in to the grid

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

```
Your identity: /O=Grid/O=NorduGrid/OU=pcd.kth.se/CN=Olav Va
```

```
Enter GRID pass phrase for this identity:XXXXXX
```

```
Creating proxy .....
```

```
Your proxy is valid until: Fri Jan 11 23:35:01 20
```

## Examine certicate

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

## Examine certificate

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

```
subject   : /O=Grid/O=NorduGrid/OU=cdc.kth.se/CN=O  
issuer    : /O=Grid/O=NorduGrid/OU=cdc.kth.se/CN=O  
identity  : /O=Grid/O=NorduGrid/OU=cdc.kth.se/CN=O  
type      : Proxy draft (pre-RFC) compliant impers  
strength  : 512 bits  
path      : /home/vahtras/x509up\_u1000  
timeleft  : 8:35:06  
subject   : /O=Grid/O=NorduGrid/OU=cdc.kth.se/CN=O
```

## Where are you allowed to go?

The certificate identifies who you are but not what you are allowed to do.

```
$ ngtest -R
```

## Where are you allowed to go?

The certificate identifies who you are but not what you are allowed to do.

```
$ ngtest -R
```

```
Resource authorizations:
```

```
Certificate-subjectname:
```

```
/O=Grid/O=NorduGrid/OU=cdc.kth.se/CN=Olav Vahtras
```

```
Retrieving information...
```

```
You are authorized at the following clusters:
```

```
bluesmoke.nsc.liu.se  grid.tsl.uu.se  hagrid.it.uu.se
```

```
hive.unicc.chalmers.se
```

```
ingrid.hpc2n.umu.se  sigrid.lunarc.lu.se
```

## Where are you allowed to go?

```
$ ngtest -R (Continued)
```

```
...
```

```
You are authorized at the following storage-elements:  
se1:bluesmoke.nsc.liu.se   vol10:harry.hagrid.it.uu.se  
vol5:harry.hagrid.it.uu.se  vol6:harry.hagrid.it.uu.se  
vol7:harry.hagrid.it.uu.se  vol8:harry.hagrid.it.uu.se  
vol9:harry.hagrid.it.uu.se  se2:storage.bluesmoke.nsc.liu.se  
se5:storage.bluesmoke.nsc.liu.se  se6:storage.bluesmoke.nsc.liu.se  
se3:storage2.bluesmoke.nsc.liu.se  se8:storage2.bluesmoke.nsc.liu.se
```

## Grid resources

\$ `ngtest -R` generates output on two types of grid resources

## Grid resources

- \$ `ngtest -R` generates output on two types of grid resources
- ▶ Computational elements (CE) - clusters at different computational centers which dedicated to grid calculations

## Grid resources

- \$ `ngtest -R` generates output on two types of grid resources
- ▶ Computational elements (CE) - clusters at different computational centers which dedicated to grid calculations
  - ▶ Storage elements (SE) - disks accessed by URL protocol, can be used to store large data-sets, codes etc.

## xRSL scripts

Job submission must be prepared in script files in xRSL (Extended Resource Specification Language)

```
$ cat hello.xrsl
```

## xRSL scripts

Job submission must be prepared in script files in xRSL (Extended Resource Specification Language)

```
$ cat hello.xrsl
```

```
&(executable="/bin/echo")(arguments="Hello world!")
```

## xRSL scripts

Job submission must be prepared in script files in xRSL (Extended Resource Specification Language)

```
$ cat hello.xrsl
```

```
&(executable="/bin/echo")(arguments="Hello world!")
```

```
$ ngsb -f hello.xrsl
```

## xRSL scripts

Job submission must be prepared in script files in xRSL (Extended Resource Specification Language)

```
$ cat hello.xrsl
```

```
&(executable="/bin/echo")(arguments="Hello world!")
```

```
$ ngsb -f hello.xrsl
```

```
Job submitted with jobid: gsiftp://grid.tsl.uu.se:2811/jobs
```

## xRSL scripts

Job submission must be prepared in script files in xRSL (Extended Resource Specification Language)

```
$ cat hello.xrsl
```

```
&(executable="/bin/echo")(arguments="Hello world!")
```

```
$ ngsb -f hello.xrsl
```

```
Job submitted with jobid: gsiftp://grid.tsl.uu.se:2811/jobs
```

```
$ ngstat -a
```

## xRSL scripts

Job submission must be prepared in script files in xRSL (Extended Resource Specification Language)

```
$ cat hello.xrsl
```

```
&(executable="/bin/echo")(arguments="Hello world!")
```

```
$ ngsbub -f hello.xrsl
```

```
Job submitted with jobid: gsiftp://grid.tsl.uu.se:2811/jobs
```

```
$ ngstat -a
```

```
Job~gsiftp://grid.tsl.uu.se:2811/jobs/43681199973
```

```
Status: FAILED
```

```
Error: LRMS error: (1) Job was killed by PBS.
```

```
Job information not found:
```

Next try:

```
$ cat hello.xrsl
```

Next try:

```
$ cat hello.xrsl  
&(executable="/bin/echo")  
(arguments="Hello world!")  
(cputime="1 minute")
```

Next try:

```
$ cat hello.xrsl
```

```
&(executable="/bin/echo")  
(arguments="Hello world!")  
(cputime="1 minute")
```

```
$ ngsbub -f hello.xrsl
```

Next try:

```
$ cat hello.xrsl
```

```
&(executable="/bin/echo")  
(arguments="Hello world!")  
(cputime="1 minute")
```

```
$ ngsb -f hello.xrsl
```

Job submitted with jobid:

```
gsiftp://hive.unicc.chalmers.se:2811/jobs/5878120
```

Next try:

```
$ cat hello.xrsl
```

```
&(executable="/bin/echo")  
(arguments="Hello world!")  
(cputime="1 minute")
```

```
$ ngsbub -f hello.xrsl
```

```
Job submitted with jobid:
```

```
gsiftp://hive.unicc.chalmers.se:2811/jobs/58781200
```

```
$ ngstat gsiftp://hive.unicc.chalmers.se:2811/jobs/58781200
```

Next try:

```
$ cat hello.xrsl
```

```
&(executable="/bin/echo")  
(arguments="Hello world!")  
(cputime="1 minute")
```

```
$ ngsb -f hello.xrsl
```

```
Job submitted with jobid:
```

```
gsiftp://hive.unicc.chalmers.se:2811/jobs/5878120
```

```
$ ngstat gsiftp://hive.unicc.chalmers.se:2811/jobs/5878120
```

```
Job gsiftp://hive.unicc.chalmers.se:2811/jobs/587
```

```
Status: FINISHED
```

```
Exit Code: 0
```

Next try:

```
$ cat hello.xrsl
```

```
&(executable="/bin/echo")
(arguments="Hello world!")
(cputime="1 minute")
```

```
$ ngsb -f hello.xrsl
```

```
Job submitted with jobid:
```

```
gsiftp://hive.unicc.chalmers.se:2811/jobs/5878120
```

```
$ ngstat gsiftp://hive.unicc.chalmers.se:2811/jobs/5878120
```

```
Job gsiftp://hive.unicc.chalmers.se:2811/jobs/587
```

```
Status: FINISHED
```

```
Exit Code: 0
```

Apparently a successful job - but where is the output?

Next try:

```
$ cat hello2.xrsl
```

Next try:

```
$ cat hello2.xrsl  
&(executable="/bin/echo")  
(arguments="Hello world!")  
(cputime="1 minute")  
(stdout="hello.out")
```

Next try:

```
$ cat hello2.xrsl  
&(executable="/bin/echo")  
(arguments="Hello world!")  
(cputime="1 minute")  
(stdout="hello.out")  
  
$ ngsb -f hello2.xrsl
```

Next try:

```
$ cat hello2.xrsl
```

```
&(executable="/bin/echo")  
(arguments="Hello world!")  
(cputime="1 minute")  
(stdout="hello.out")
```

```
$ ngsb -f hello2.xrsl
```

```
Job submitted with jobid: gsiftp://grid.tsl.uu.se
```

Next try:

```
$ cat hello2.xrsl
```

```
&(executable="/bin/echo")  
(arguments="Hello world!")  
(cputime="1 minute")  
(stdout="hello.out")
```

```
$ ngsb -f hello2.xrsl
```

```
Job submitted with jobid: gsiftp://grid.tsl.uu.se
```

```
$ ngget gsiftp://grid.tsl.uu.se:2811/jobs/13530120112024911
```

Next try:

```
$ cat hello2.xrsl
```

```
&(executable="/bin/echo")  
(arguments="Hello world!")  
(cputime="1 minute")  
(stdout="hello.out")
```

```
$ ngsb -f hello2.xrsl
```

```
Job submitted with jobid: gsiftp://grid.tsl.uu.se
```

```
$ ngget gsiftp://grid.tsl.uu.se:2811/jobs/13530120112024911
```

```
Results stored at /home/vahtras/swegrid/13530120112024911
```

```
Jobs processed: 1, successfully downloaded: 1
```

Next try:

```
$ cat hello2.xrsl
```

```
&(executable="/bin/echo")
(arguments="Hello world!")
(cputime="1 minute")
(stdout="hello.out")
```

```
$ ngsb -f hello2.xrsl
```

```
Job submitted with jobid: gsiftp://grid.tsl.uu.se
```

```
$ ngget gsiftp://grid.tsl.uu.se:2811/jobs/13530120112024911
```

```
Results stored at /home/vahtras/swegrid/135301201
```

```
Jobs processed: 1, successfully downloaded: 1
```

```
$ ls 1353012011202491134775914
```

Next try:

```
$ cat hello2.xrsl
```

```
&(executable="/bin/echo")
(arguments="Hello world!")
(cputime="1 minute")
(stdout="hello.out")
```

```
$ ngsb -f hello2.xrsl
```

```
Job submitted with jobid: gsiftp://grid.tsl.uu.se
```

```
$ ngget gsiftp://grid.tsl.uu.se:2811/jobs/13530120112024911
```

```
Results stored at /home/vahtras/swegrid/1353012011
```

```
Jobs processed: 1, successfully downloaded: 1
```

```
$ ls 1353012011202491134775914
```

```
hello.out
```

Next try:

```
$ cat hello2.xrsl
```

```
&(executable="/bin/echo")  
(arguments="Hello world!")  
(cputime="1 minute")  
(stdout="hello.out")
```

```
$ ngsbub -f hello2.xrsl
```

```
Job submitted with jobid: gsiftp://grid.tsl.uu.se
```

```
$ ngget gsiftp://grid.tsl.uu.se:2811/jobs/1353012011202491134775914
```

```
Results stored at /home/vahtras/swegrid/1353012011202491134775914
```

```
Jobs processed: 1, successfully downloaded: 1
```

```
$ ls 1353012011202491134775914
```

```
hello.out
```

```
$ cat 1353012011202491134775914/hello.out
```

Next try:

```
$ cat hello2.xrsl
```

```
&(executable="/bin/echo")  
(arguments="Hello world!")  
(cputime="1 minute")  
(stdout="hello.out")
```

```
$ ngsbub -f hello2.xrsl
```

```
Job submitted with jobid: gsiftp://grid.tsl.uu.se
```

```
$ ngget gsiftp://grid.tsl.uu.se:2811/jobs/1353012011202491134775914
```

```
Results stored at /home/vahtras/swegrid/1353012011202491134775914
```

```
Jobs processed: 1, successfully downloaded: 1
```

```
$ ls 1353012011202491134775914
```

```
hello.out
```

```
$ cat 1353012011202491134775914/hello.out
```

```
Hello world!
```

Next try:

```
$ cat hello2.xrsl
```

```
&(executable="/bin/echo")  
(arguments="Hello world!")  
(cputime="1 minute")  
(stdout="hello.out")
```

```
$ ngsbub -f hello2.xrsl
```

```
Job submitted with jobid: gsiftp://grid.tsl.uu.se
```

```
$ ngget gsiftp://grid.tsl.uu.se:2811/jobs/1353012011202491134775914
```

```
Results stored at /home/vahtras/swegrid/1353012011202491134775914
```

```
Jobs processed: 1, successfully downloaded: 1
```

```
$ ls 1353012011202491134775914
```

```
hello.out
```

```
$ cat 1353012011202491134775914/hello.out
```

```
Hello world!
```

## input/output

Jobs that require input/output files need to specify this in the script with (filename location) pairs

Example

- ▶ Input files

```
(inputFiles=("test.dal" " ")("test.mol" " "))
```

- ▶ Output files

```
(outputFiles=("test.out" " ")("test.tgz" " "))
```

Empty input location means current dir

Empty output location means "save for later"

## Runtime environments

To use preinstalled software one specifies so called runtime environments. Examples

- ▶ Gaussian  
(runtimeenvironment="APPS/CHEM/GAUSSIAN-03.C02")
- ▶ Dalton  
(runtimeenvironment="APPS/CHEM/DALTON-2.0-1.0")
- ▶ Molcas  
(runtimeenvironment>="APPS/CHEM/MOLCAS")

See also the runtime environment registry: <http://gridrer.csc.fi/>

# A Gaussian test job

```
$ cat g03.xrsl
```

# A Gaussian test job

```
$ cat g03.xrsl
&(runtimeenvironment="APPS/CHEM/GAUSSIAN-03.C02"
(executable="$g03root/g03/bsd/g031")
(arguments="test000.com")
(count=4)
(inputfiles=("test000.com" ""))
(outputfiles=("test000.log" ""
              ("test000.chk" ""))
(cpuTime="5 minutes")
(jobname="test000")
```

```
$ ngsb -f g03.xrsl
```

```
$ ngsbub -f g03.xrsl
```

```
Job submitted with jobid: gsiftp://bluesmoke.nsc.
```

```
$ ngsb -f g03.xrsl
```

```
Job submitted with jobid: gsiftp://bluesmoke.nsc.
```

```
$ ngstat test000
```

```
$ ngsbub -f g03.xrsl
```

```
Job submitted with jobid: gsiftp://bluesmoke.nsc.
```

```
$ ngstat test000
```

```
Job gsiftp://bluesmoke.nsc.liu.se:2811/jobs/16743
```

```
Job Name: test000
```

```
Status: FINISHED
```

```
$ ngsbub -f g03.xrsl
```

```
Job submitted with jobid: gsiftp://bluesmoke.nsc.
```

```
$ ngstat test000
```

```
Job gsiftp://bluesmoke.nsc.liu.se:2811/jobs/16743
```

```
  Job Name: test000
```

```
  Status: FINISHED
```

```
$ ngget test000
```

```
$ ngsb -f g03.xrsl
```

```
Job submitted with jobid: gsiftp://bluesmoke.nsc.
```

```
$ ngstat test000
```

```
Job gsiftp://bluesmoke.nsc.liu.se:2811/jobs/16743
```

```
Job Name: test000
```

```
Status: FINISHED
```

```
$ ngget test000
```

```
Results stored at /home/vahtras/swegrid/167431200
```

```
Jobs processed: 1, successfully downloaded: 1
```

```
$ ngsb -f g03.xrsl
```

```
Job submitted with jobid: gsiftp://bluesmoke.nsc.
```

```
$ ngstat test000
```

```
Job gsiftp://bluesmoke.nsc.liu.se:2811/jobs/16743
```

```
Job Name: test000
```

```
Status: FINISHED
```

```
$ ngget test000
```

```
Results stored at /home/vahtras/swegrid/167431200
```

```
Jobs processed: 1, successfully downloaded: 1
```

```
$ ls 167431200654261715239193
```

```
$ ngsb -f g03.xrsl
```

```
Job submitted with jobid: gsiftp://bluesmoke.nsc.
```

```
$ ngstat test000
```

```
Job gsiftp://bluesmoke.nsc.liu.se:2811/jobs/16743
```

```
Job Name: test000
```

```
Status: FINISHED
```

```
$ ngget test000
```

```
Results stored at /home/vahtras/swegrid/167431200
```

```
Jobs processed: 1, successfully downloaded: 1
```

```
$ ls 167431200654261715239193
```

```
test000.log
```

```
$ ngsb -f g03.xrsl
```

```
Job submitted with jobid: gsiftp://bluesmoke.nsc.
```

```
$ ngstat test000
```

```
Job gsiftp://bluesmoke.nsc.liu.se:2811/jobs/16743
```

```
Job Name: test000
```

```
Status: FINISHED
```

```
$ ngget test000
```

```
Results stored at /home/vahtras/swegrid/167431200
```

```
Jobs processed: 1, successfully downloaded: 1
```

```
$ ls 167431200654261715239193
```

```
test000.log
```

# A molcas test job

```
$ cat molcas.xrsl
```

## A molcas test job

```
$ cat molcas.xrsl
&(executable="$MOLCAS/bin/molcas")
(arguments="test001.input")
(stderr="stderr.txt")
(stdout="stdout.txt")
(gmlog="gmlog")
(inputfiles= ("test001.input" ""))
(outputfiles= ("stdout.txt" ""))
(runtimeEnvironment >="APPS/CHEM/MOLCAS")
(environment=(CPUS 1))
(jobName="test001")
(CPUTime=10)
(count=1)
```

# A dalton test job

```
$ cat dalton.xrsl
```

## A dalton test job

```
$ cat dalton.xrsl
&(jobname="daltest")
(runtimeenvironment="APPS/CHEM/DALTON-2.0-1.0")
(cputime="10 minutes")
(executable="/usr/bin/time")
(arguments="dalton" "energy_nosymm" "energy_nosym")
(inputfiles=("energy_nosymm.dal" "")("energy_nosymm.dal" ""))
(outputfiles=("energy_nosymm.out" "")("energy_nosymm.out" ""))
(stdout="stdout")
(stderr="stderr")
(gmlog="gmlog")
```

```
$ ngsb -d 1 -f dalton.xrsl
```

```
$ ngsb -d 1 -f dalton.xrsl
```

```
[vahtras@anton swegrid]$ ngsb -d 1 -f dalton.xrsl  
Proxy subject name: /O=Grid/O=NorduGrid/OU=fdc.kt  
Proxy valid to: 2008-01-25 00:42:31  
Proxy valid for: 11 hours, 22 minutes, 49 seconds  
Queue selected: atlas@bluesmoke.nsc.liu.se  
Job submission failed to queue atlas@bluesmoke.nsc.liu.se  
Trying remaining queues  
Queue selected: workq@sigrid.lunarc.lu.se  
File uploaded: /tmp/vahtras/rsl.iFcMJI  
File uploaded: /home/vahtras/swegrid/energy_nosym  
File uploaded: /home/vahtras/swegrid/energy_nosym  
Job submitted with jobid: gsiftp://sigrid.lunarc.lu.se
```

## User software

Things to consider when running own software

- ▶ Portability is an issue - different clusters have different linux distributions, different versions etc.
- ▶ Statically linked 32-bit programs work usually fine.
- ▶ MPI support limited
- ▶ Small programs can be compiled as a part of the job.

```
$ cat mydal.xrsl
```

```
$ cat mydal.xrsl
&(executable="rundal ")
(arguments="energy_nosymm ")
(cputime="1 minute")
(stdout="log")
(stderr="err")
(inputFiles=
  ("energy_nosymm.dal" "")
  ("energy_nosymm.mol" "")
  ("cc-pVDZ" "/home/vahtras/dalton/HEAD/src/basis/
  ("ano-4" "/home/vahtras/dalton/HEAD/src/basis/an
  ("dalton" "bin/dalton")
  ("dalton.x" "/home/vahtras/dalton/HEAD/build/g77
  ("rundal" ""))
)
```

```
(outputFiles=  
  ("energy_nosymm.out" "")  
  ("energy_nosymm.tgz" "")  
)  
(jobname="rundal")
```

```
$ cat rundal
```

```
$ cat rundal
#!/bin/sh
uname -a
cat /proc/cpuinfo
cat /etc/*-release
pwd
ls -lR
printenv
chmod +x dalton dalton.x
export DALHOME=$PWD
./dalton -b $DALHOME $@
```

```
$ ngsb -d 1 -f mydal.xrsl
```

```
$ ngsbub -d 1 -f mydal.xrsl
```

```
Proxy subject name: /O=Grid/O=NorduGrid/OU=pdc.kt
```

```
Proxy valid to: 2008-02-01 22:48:09
```

```
Proxy valid for: 11 hours, 55 minutes, 41 seconds
```

```
Queue selected: atlas@bluesmoke.nsc.liu.se
```

```
Job submission failed to queue atlas@bluesmoke.nsc.liu.se
```

```
Trying remaining queues
```

```
Queue selected: ngrid@ingrid.hpc2n.umu.se
```

```
File uploaded: /tmp/vahtras/rsl.YetI6T
```

```
File uploaded: /home/vahtras/dalton/HEAD/build/g7
```

```
File uploaded: /home/vahtras/dalton/HEAD/src/basi
```

```
File uploaded: /home/vahtras/dalton/HEAD/src/basi
```

```
File uploaded: /home/vahtras/swegrid/bin/dalton
```

```
File uploaded: /home/vahtras/swegrid/energy_nosym
```

```
File uploaded: /home/vahtras/swegrid/energy_nosym
```

```
File uploaded: /home/vahtras/swegrid/rundal
```

```
$ ngget -d 1 rundal
```

```
$ ngget -d 1 rundal
```

```
Proxy subject name: /O=Grid/O=NorduGrid/OU=pcdc.kth.se  
Proxy valid to: 2008-02-01 22:48:09  
Proxy valid for: 11 hours, 39 minutes, 59 seconds  
File downloaded: gsiftp://ingrid.hpc2n.umu.se:2811/j  
File downloaded: gsiftp://ingrid.hpc2n.umu.se:2811/j  
File downloaded: gsiftp://ingrid.hpc2n.umu.se:2811/j  
Results stored at /home/vahtras/swegrid/166011201  
Deleting job: gsiftp://ingrid.hpc2n.umu.se:2811/j  
Jobs processed: 1, successfully downloaded: 1
```

## More information

- ▶ This document on <http://www.theochem.kth.se/docs/swegrid.pdf>
- ▶ [www.nordugrid.org](http://www.nordugrid.org)
- ▶ [www.swegrid.se](http://www.swegrid.se)
- ▶ All support questions to [support@swegrid.se](mailto:support@swegrid.se)