



# ALICE Offline Tutorial

## Using the AliEn Grid Client

andrei.gheata@cern.ch  
GSI, 4<sup>th</sup> Mar. 2010



# Contents I

- Prerequisites
- Installation of the AliEn Grid client
- Connection and Login/Authentication
- General description of the shell
  - Functionality and orientation
  - Basic commands
  - View, edit and copy files within the catalogue



# Contents II

- Grid Jobs:
  - Job submission, status and control
  - Overview of the JDL files (Job Description Language)
- Working with the file catalogue
  - Copying files from/to the catalogue.
  - Creating collections of files
- Working with the AliEn plug-in
  - Configure your own plug-in
  - Run the analysis in grid via the plug-in



# Prerequisites

**Did you follow ALL the steps for the user registration ?**

**Do you have valid usercert.pem and userkey.pem files ?**

If not, you will only be able to watch this tutorial ...

The registration was supposed to be done at:

*<http://alien.cern.ch/twiki/bin/view/Alice/UserRegistration>*



# Installing the AliEn client

Get the client installer (alien-installer)

```
wget http://alien.cern.ch/alien-installer
```

Make the file executable

```
chmod +x alien-installer
```

Just start the installer and wait:

```
[pcepalice10] /home/alienmaster > ./alien-installer
##### Automated AliEn user installer #####
Installing in the default directory: /home/alienmaster/alien
If you wish to install in a specific directory do: ./alien-installer -install-dir <path>
Platform set as: x86_64-unknown-linux-gnu
Waiting 10 seconds. Press 'Ctrl+c' to exit the installation.....Starting installation
Downloading user distribution..Done
Extracting the files.....Done
Relocating the user installation.Done
Installation finished!
[pcepalice10] /home/alienmaster > ls -l
total 28
lrwxrwxrwx  1 alienmaster alienmaster   32 Feb 16 01:17 alien -> /home/alienmaster/alien.v2-17.84
-rwxr-xr-x  1 alienmaster alienmaster 17586 Feb 15 15:53 alien-installer
drwxr-xr-x 11 alienmaster alienmaster  4096 Feb 16 01:17 alien.v2-17.84
drwxr-xr-x  2 alienmaster alienmaster  4096 Feb 16 01:17 bin
[pcepalice10] /home/alienmaster > █
```



# Installing AliEn client (cont.)

... or if you want to specify an alternate installation location

```
[pcepalice10] /home/alienmaster > ./alien-installer -install-dir MyAliEn
##### Automated AliEn user installer #####
Installing in the user specified directory: /home/alienmaster/MyAliEn
Platform set as: x86_64-unknown-linux-gnu
Waiting 10 seconds. Press 'Ctrl+c' to exit the installation.....Starting installation
Downloading user distribution..Done
Extracting the files.....Done
Relocating the user installation.Done
Installation finished!
[pcepalice10] /home/alienmaster > ls -l
total 28
-rwxr-xr-x  1 alienmaster alienmaster 17586 Feb 15 15:53 alien-installer
drwxr-xr-x 11 alienmaster alienmaster  4096 Feb 16 01:25 alien.v2-17.84
drwxr-xr-x  2 alienmaster alienmaster  4096 Feb 16 01:25 bin
lrwxrwxrwx  1 alienmaster alienmaster   32 Feb 16 01:25 MyAliEn -> /home/alienmaster/alien.v2-17.84
[pcepalice10] /home/alienmaster > █
```



# Installing AliEn client (cont.)

If you didn't have the folder `~/bin`, the installer created it for you

But you have ensure it is set in your **PATH** Shell Environment Variable

**Set it in the appropriate configuration file for your shell !**

Or you'll have to set it each time you open a new shell to login:

```
export PATH=$PATH:~/bin
```



# Installation – Try it out

- Copy grid certificates to the computer in front of you

```
mkdir .globus
```

```
scp <username>@lxplus:.globus/*.pem .globus
```

- Verify location of your certificate+key and their permissions

```
~/.globus - 750 userkey.pem - 400 usercert.pem - 640
```

- Download the alien installer: ***<http://alien.cern.ch/alien-installer>***

- Make the file executable

- Run the installer (specify alternative installation location)

- Check the installation went fine and you see the directories

- Do ***export PATH=\$PATH:~/bin***







# Authenticating at the AliEn Shell (I)

To access the AliEn Shell you have to authenticate every 24h by creating your access token with *alien-token-init <grid-cert-username>*

```
[pcalice22] /home/alienmaster > alien-token-init
-----
Setting central config:
=====
export alien_API_SERVER_LIST="pcapiserv01.cern.ch:10000|pcapiserv02.cern.ch:10000|pcapiserv03.cern
export alien_API_PORT=10000
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$GSHELL_ROOT/lib
export TERMINFO=/usr/share/terminfo
=====
*****
Attention: You don't have a valid grid proxy - doing grid-proxy-init for you ...
*****
Your identity: /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=alienmaster/CN=677029/CN=alienmaster
Enter GRID pass phrase for this identity: █
```



# Authenticating at the AliEn Shell (II)

```
[pcalice22] /home/alienmaster > alien-token-init
-----
Setting central config:
=====
export alien_API_SERVER_LIST="pcapiserv01.cern.ch:10000|pcapiserv02.cern.ch:10000|pcapiserv03.cern
export alien_API_PORT=10000
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$GSHELL_ROOT/lib
export TERMINFO=/usr/share/terminfo
=====
*****
Attention: You don't have a valid grid proxy - doing grid-proxy-init for you ...
*****
Your identity: /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=alienmaster/CN=677029/CN=alienmaster
Enter GRID pass phrase for this identity:
Creating proxy ..... Done
Your proxy is valid until: Tue Jan 26 06:02:45 2010
=> Trying to connect to Server [1] http://pcapiserv02.cern.ch:10000 as User alienmaster
Your identity: alienmaster
Creating token ..... Done
Your token is valid until: Tue Jan 26 18:02:45 2010
[pcalice22] /home/alienmaster > █
```



# Authentication – Problems

- Permissions on `~/.globus/userkey.pem` are not private to your user  
`chmod 400 userkey.pem`
- Your certificate authority is exotic and not known to the server
- Your certificate has expired
- You have not given the AliEn user name as an argument to the `token init` command and your local user name is not identical to the AliEn user name
- Clock skew - your local computer time is out of the validity time of your certificate



# Authentication – Try it out

- Do *alien-token-init* *<your-grid-cert-username>*
- If asked about compiling the gapi and xrootd libs say "no"
- Later, to install on your own machine and do analysis you'll have to say "yes".
- At the end you should have a valid token.





# AliEn Shell

Doing *aliensh* ...

```
[pcepalice10] /home/alienmaster > aliensh  
[ aliensh 1.0.10x (C) ARDA/Alice: Andreas.Joachim.Peters@cern.ch/Derek.Feichtinger@cern.ch]  
aliensh:[alice] [1] /alice/cern.ch/user/a/alienmas/ >
```

- Standard bash shell with grid commands
- Main shell features are available
- Command / file / path completion



# AliEn Shell - Basic commands

```
[pcepalice10] /home/alienmaster > aliensh  
[ aliensh 1.0.10x (C) ARDA/Alice: Andreas.Joachim.Peters@cern.ch/Derek.Feichtinger@cern.ch]  
aliensh:[alice] [1] /alice/cern.ch/user/a/alienmas/ >
```

- Standard Unix Shell commands work as usual:

*ls, cd, mkdir/rmdir, cat, more, pwd, whoami ...*

- There's a **help** command to list all known commands
- Get a complete command list by typing **<tab><tab>**
- Commands have **'-h'** flag to print out a short help message



# Shell – editing files

```
aliensh:[alice] [14] /alice/cern.ch/user/a/alienmas/test/ >edit tutorial.test
```

All old versions of the edited file are saved in a hidden folder

```
aliensh:[alice] [15] /alice/cern.ch/user/a/alienmas/test/ >ls
.tutorial.test
tutorial.test
aliensh:[alice] [16] /alice/cern.ch/user/a/alienmas/test/ >cd .tutorial.test/
aliensh:[alice] [17] /alice/cern.ch/user/a/alienmas/test/.tutorial.test/ >ls -a
.
..
v1.0
v1.1
```

To delete all the old versions (NOT the file itself with your last changes )

```
aliensh:[alice] [33] /alice/cern.ch/user/a/alienmas/test/ >purge tutorial.test
Jan 26 22:49:42 info  purge: =====> purging file /alice/cern.ch/user/a/alienmas/te
st/tutorial.test

Jan 26 22:49:42 info  purge: cleaning v1.0 for /alice/cern.ch/user/a/alienmas/test/tutorial.test
Jan 26 22:49:42 info  Making the envelope ourselves: -s delete /alice/cern.ch/user/a/alienmas/test/.tutori
al.test/v1.0 ALICE::CCIN2P3::SE
Jan 26 22:49:42 info  Checking the size in ALICE::CCIN2P3::SE (1073741824)
```

You can choose your editor in the file `~/.alienshrc` :

`export EDITOR='your-choice' : emacs | emacs -nw | xemacs | xemacs -nw | pico | vi | vim` ("vi" is default)



# Shell – Copying files from/to the Catl.

- The **cp** command works just like the Unix Shell command but is operating on the files in the Grid Catalogue.
- To specify files on your local disk, use "**file:**" as a location prefix
- You may need to define the environment variable **alien\_CLOSE\_SE** pointing to a SE that is accessible and close to your location
  - **export alien\_CLOSE\_SE="ALICE::GSI::SE"**
- You can always specify the SE location in the copy commands
  - cp <source> <alien\_destination>@someSE**

```
aliensh:[alice] [47] /alice/cern.ch/user/a/alienmas/test/ >ls -a
.
..
aliensh:[alice] [48] /alice/cern.ch/user/a/alienmas/test/ >cp file:tutorial.test tutorial.test.grid
[xrootd] Total 0.00 MB |=====| 100.00 % [inf MB/s]
aliensh:[alice] [49] /alice/cern.ch/user/a/alienmas/test/ >cp tutorial.test.grid tutorial.test.grid.copy
[xrootd] Total 0.00 MB |=====| 100.00 % [0.3 MB/s]
[xrootd] Total 0.00 MB |=====| 100.00 % [inf MB/s]
aliensh:[alice] [50] /alice/cern.ch/user/a/alienmas/test/ >cp tutorial.test.grid.copy file:backup.local
[xrootd] Total 0.00 MB |=====| 100.00 % [0.3 MB/s]
aliensh:[alice] [51] /alice/cern.ch/user/a/alienmas/test/ >ls
tutorial.test.grid
tutorial.test.grid.copy
aliensh:[alice] [52] /alice/cern.ch/user/a/alienmas/test/ >
```





# Shell – “whereis” command

Where is the file *tutorial.test* actually stored ?

```
aliensh:[alice] [15] /alice/cern.ch/user/a/alienmas/test/ >whereis tutorial.test
Jan 26 20:20:32 info The file test/tutorial.test is in
SE => ALICE::CCIN2P3::SE pfn =>root://ccxrdsn038.in2p3.fr:1094//02/55499/78ed59
24-0a9f-11df-8cf4-001e0bd3f44c
SE => ALICE::Clermont::SE pfn =>root://clralicexrd.in2p3.fr:1094//02/55499/78ed
5924-0a9f-11df-8cf4-001e0bd3f44c
SE => ALICE::Cyfronet::SE pfn =>root://alice-se.grid.cyf-kr.edu.pl:1094//02/554
99/78ed5924-0a9f-11df-8cf4-001e0bd3f44c
aliensh:[alice] [16] /alice/cern.ch/user/a/alienmas/test/ >
```



# Shell – Try it out

- Access the alien shell
- Check your user name by typing *whoami*
- List the contents of your home directory
- Do the following in your AliEn space:
  - Create the directories `~/bin` , `~/tutorial/` and `~/tutorial/output`
  - `cp /alice/cern.ch/user/s/sschrein/tutorial/tutorial_textfile ~/tutorial`
  - `cp /alice/cern.ch/user/s/sschrein/tutorial/grid_tutorial.pdf ~/tutorial`
  - Now copy the `grid_tutorial.pdf` to your local machine
- Get the information of the file (*whereis*) **tutorial\_textfile**
- Edit the file and append a comment of yourself
- Copy it to your local machines home directory and  
check it's there and you can open it





# Grid Jobs – JDL Files I

- **Executable:** Compulsory field where we give the name of the executable (must be stored in /bin or \$V0/bin or ~/bin)

```
Executable = "alienroot";
```

- **Arguments:** Will be passed to the executable

```
Arguments = "-q -b";
```

- **Packages:** Type *packages* in the shell to see what packages are installed

```
Packages = { "APICONFIG::V2.2" , "ROOT::v5-13" };
```

- **InputFile:** The files that will be transported to the node where the job will run

```
InputFile = { "LF:/alice/cern.ch/user/a/alip/macros/bAnalysis.C" };
```

- **Validationcommand:** Specifies the script to be used as a validation script

```
Validationcommand = "/alice/cern.ch/user/a/alienmas/validation.sh";
```



# Grid Jobs – JDL Files II

- **InputData:** It will require that the job will be executed in a site close to the files specified here

```
InputData = { "LF:/alice/cern.ch/data/AliESDs.root,nodownload" };
```

- **InputDataCollection:** The filename of the collection of the input data

```
InputDataCollection = "LF:/alice/cern.ch/data/002.xml,nodownload" ;
```

- **InputDataList:** The filename in which the job will get the input data collection

```
InputDataList = "collection.xml" ;
```

- **InputDataListFormat:** The format of the InputData list

```
InputDataListFormat = "xml-single" ;
```

- **Email:** Receive a mail when the job finishes

```
Email = "alienmaster@cern.ch" ;
```



# Grid Jobs – JDL Files III

- **TTL:** The maximum run time of your job in seconds

```
TTL = 7200 ;
```

- **Split:** Split the jobs in several sub jobs

```
split = "se" ;
```

- **MasterResubmitThreshold:** Resubmit sub jobs, if less than are successful

```
MasterResubmitThreshold= "99%" ; # or give a absolute Job number
```

- **SplitMaxInputFileNumber:** Max input file count of each sub job

```
SplitMaxInputFileNumber= "100" ;
```

- **OutputDir:** Where the output files+archives will be stored

```
OutputDir = "/alice/cern.ch/user/a/aliprod/analysis/output101" ;
```



# Grid Jobs – JDL Files IV

- **OutputFile:** The files that will be registered in the catalogue once the job finishes

```
OutputFile = { "myFilename" };           # default 2 copies (disk=2)
```

```
OutputFile = { "myFilename@disk=3" };    # give me 3 copies
```

- **OutputArchive:** What files will be archived in a zip file

```
OutputArchive = { "myArchivename:*.*root" };    # analogue above
```

We have a Storage Element discovery and failover mechanism, storing your output files by default always in the two topmost locations. To get more (up to 9 copies), you can specify the count.

→ DON'T USE THE EXPLICIT FORMAT ( you may find it old JDLs ):

```
as e.g. OutputFile = {"filename@ALICE::CERN::SE"};
```



# Grid Jobs – JDL File Example

```
Packages = { "VO_ALICE@AliRoot::v4-18-16-AN", "VO_ALICE@ROOT::v5-25-04-3", "VO_ALICE@APISCONFIG::V1.1x" };
Executable = "/alice/cern.ch/user/a/alienmas/bin/PPQMixexe.sh";

InputFile = {
"LF:/alice/cern.ch/user/a/alienmas/PPQMix/runAnalysis.C",
"LF:/alice/cern.ch/user/a/alienmas/PPQMix/PPQMixexe.root",
"LF:/alice/cern.ch/user/a/alienmas/PPQMix/ConfigureCuts.C",
"LF:/alice/cern.ch/user/a/alienmas/PPQMix/PPQMixTask.h",
"LF:/alice/cern.ch/user/a/alienmas/PPQMix/PPQMixTask.cxx" };

InputDataListFormat = "xml-single";
InputDataList = "wn.xml";
InputDataCollection = { "LF:/alice/cern.ch/user/a/alienmas/PPQMix/0001048,nodownload" };
MasterResubmitThreshold = "99%";
Split = "se";
SplitMaxInputFileNumber = "100";

OutputArchive = { "log_archive.zip:stdout,stderr" };
OutputFile = { "output1.root" };
OutputDir = "/alice/cern.ch/user/a/alienmas/PPQMix/output/003";

TTL = 30000;
Validationcommand = "/alice/cern.ch/user/a/alienmas/PPQMix/PPQMixexe_validation.sh";
Jobtag = { "comment:My analysis Job" };
```



# Job Validation

You're supposed to have an error validation script for your jobs!

```
#!/bin/bash

#... Some missing content here

echo "*****" >> stdout
echo "* Time:    $validatetime " >> stdout

segFault=`grep -Ei "Segmentation fault" stderr`
if [ "$segFault" != "" ] ; then
    error=1
    echo "* ##### Job not validated - Segment. fault   ###" >> stdout
    echo "$segFault" >> stdout
    echo "Error = $error" >> stdout
fi
if ! [ -f *.file ] ; then
    error=1
    echo "Output file(s) not found. Job FAILED !" >> stdout
    echo "Output file(s) not found. Job FAILED !" >> stderr
fi
if [ $error = 0 ] ; then
    echo "* ----- Job Validated -----*" >> stdout
fi
cd -
exit $error
```





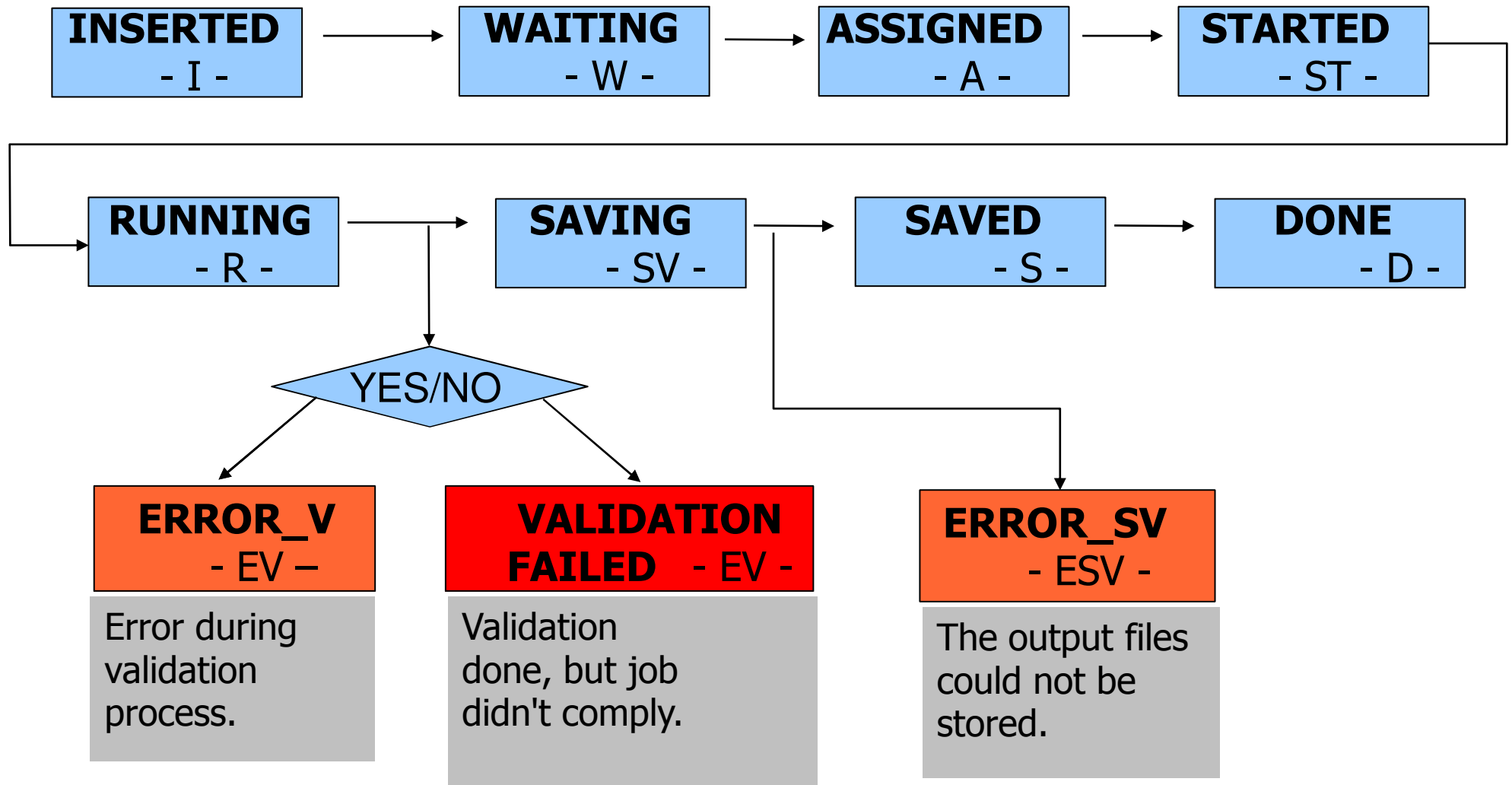
# Submitting Jobs

- In order to submit a job, call ***submit*** together with the JDL file you've created
- Your Job will be send to the AliEn Task Queue
- Thereafter, it will be picked up by an AliEn Site somewhere in the world

```
aliensh:[alice] [55] /alice/cern.ch/user/a/alienmas/ >submit TestJob.jdl
Submit submit TestJob.jdl
submit: Your new job ID is 41220350
aliensh:[alice] [56] /alice/cern.ch/user/a/alienmas/ >
```



# Job Status / Lifecycle (simplified)



All details: <http://pcalimonitor.cern.ch/show?page=jobStatus.html>



# Job Status I

To check about the Grid jobs you got two commands, both have a lot of additional parameters.

*ps* will give you a list of your jobs:

```
aliensh:[alice] [82] /alice/cern.ch/user/a/alienmas/ >ps
alienmas 41226863  _  D  command.example
alienmas 41226890  _  D  command.example
alienmas 41226891  _  SV command.example
aliensh:[alice] [83] /alice/cern.ch/user/a/alienmas/ >
```

*top* is more verbose than *ps* and will give you by default a list of ALL jobs in the queue. Attention, this can be a long list, better use parameters:

```
aliensh:[alice] [90] /alice/cern.ch/user/a/alienmas/ >top -status DONE -user alienmas
JobId  Status  Command name  SubmitHost
41149015  DONE  /alice/cern.ch/user/a/alienmas/bin/echo.bash  aliensm@pcapiserv06.cern.ch
41226863  DONE  /alice/cern.ch/user/a/alienmas/bin/date  aliensm@pcepalice10.cern.ch
```



# Job Status II – A job's JDL

*ps -jdl <Job-ID>* displays the job's JDL during or after the job's runtime.

```
aliensh:[alice] [7] /alice/cern.ch/user/a/aliemas/ >ps -jdl 41458752

[
  Packages =
  {
    "VO_ALICE@AliRoot::v4-17-Rev-24",
    "VO_ALICE@GEANT3::v1-11-7",
    "VO_ALICE@ROOT::v5-25-04-1",
    "VO_ALICE@APISCONFIG::V1.1x"
  };
  Jobtag =
  {
    "comment:EPOS, 900 GeV, Residual misalignment, ID #125"
  };
  Arguments = " -q -b -l simrun.C --run 125023 --event 816";
  OutputDir = "/alice/sim/LHC10a6/125023/816";
  Price = 1;
  MasterJobId = "41426232";
  Executable = "/alice/bin/aliproduct_new";
  InputDownload =
  {
    "/proc/aliproduct/41458752/tag.C->/alice/cern.ch/user/a/aliproduct/LHC10a6/tag.C",
    "/proc/aliproduct/41458752/simrun.C->/alice/cern.ch/user/a/aliproduct/LHC10a6/simrun.C",
    "/proc/aliproduct/41458752/sim.C->/alice/cern.ch/user/a/aliproduct/LHC10a6/sim.C",
    "/proc/aliproduct/41458752/rec.C->/alice/cern.ch/user/a/aliproduct/LHC10a6/rec.C",
    "/proc/aliproduct/41458752/EPOS.input.tar.gz->/alice/cern.ch/user/a/aliproduct/LHC10a6/EPOS.input.ta
r.gz",
    "/proc/aliproduct/41458752/CreateAODfromESD.C->/alice/cern.ch/user/a/aliproduct/LHC10a6/CreateAODfro
mESD.C",
    "/proc/aliproduct/41458752/Config.C->/alice/cern.ch/user/a/aliproduct/LHC10a6/Config.C",
    "/proc/aliproduct/41458752/CheckESD.C->/alice/cern.ch/user/a/aliproduct/LHC10a6/CheckESD.C"
  }
]
```

Be aware, during/after runtime the JDL contains more information.



# Job Status III – A job's tracelog

*ps -trace <Job-ID> all* prints out the complete job trace log during or after the job's runtime.

```
aliensh:[alice] [71] /alice/cern.ch/user/a/alienmas/ >ps -trace 41226890 all
Wed Jan 27 00:39:54 2010 [state ]: Job 41226890 inserted from alienmas@pcepalice10.cern.ch
Wed Jan 27 00:40:28 2010 [state ]: Job state transition to STARTED |=| procinfotime: 1264549228 site: ALICE:
:LLNL::PBS started: 1264549228 node: glcc1.ucllnl.org
Wed Jan 27 00:40:28 2010 [trace ]: The job has been taken by the jobagent 8139_74
Wed Jan 27 00:40:28 2010 [trace ]: The job needs 21600 seconds
Wed Jan 27 00:40:28 2010 [trace ]: Creating the working directory /home/alien/alien-job-41226890
Wed Jan 27 00:40:31 2010 [state ]: Job state transition from INSERTING to WAITING
Wed Jan 27 00:40:33 2010 [state ]: Job state transition from STARTED to RUNNING |=| procinfotime: 1264549233
site: ALICE::LLNL::PBS started: 1264549233 spyurl: glcc1.ucllnl.org:8089 node: glcc1.ucllnl.org
Wed Jan 27 00:40:34 2010 [state ]: Job state transition from RUNNING to SAVING |=| procinfotime: 1264549234
site: ALICE::LLNL::PBS error:
Wed Jan 27 00:41:14 2010 [state ]: Job state transition from WAITING to ASSIGNED (glcc0.ucllnl.org)
Wed Jan 27 00:41:28 2010 [trace ]: Downloading input file: /alice/cern.ch/user/a/alienmas/bin/date
Wed Jan 27 00:41:28 2010 [proc ]: 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Wed Jan 27 00:41:59 2010 [state ]: Job state transition to SAVED |=| procinfotime: 1264549319 site: ALICE::L
LLNL::PBS spyurl: jdl:
Wed Jan 27 00:42:29 2010 [trace ]: Finally, processing archives: myArchiveName.zip:stdout,stderr,resources@d
isk=3
Wed Jan 27 00:42:29 2010 [trace ]: Finally, processing files:
Wed Jan 27 00:42:29 2010 [trace ]: Registering myArchiveName.zip.
Wed Jan 27 00:42:29 2010 [trace ]: Successfully stored the file /home/alien/alien-job-41226890/myArchiveName
.zip on 3 SEs
Wed Jan 27 00:42:29 2010 [trace ]: OK. All files and archives for this job where uploaded as specified. Supe
rb!
Wed Jan 27 00:42:29 2010 [state ]: The job finished on the worker node with status SAVED
Wed Jan 27 00:42:35 2010 [state ]: Job state transition from SAVED to DONE
aliensh:[alice] [72] /alice/cern.ch/user/a/alienmas/ >
```



# Job Status IV – Spy on Job Output

With *spy* you can check the output of a job while it is still running

```
aliensh:[alice] [83] /alice/cern.ch/user/a/alienmas/ >spy 41226891 workdir
total 88
drwxr-x--- 2 alien alien 4096 Jan 26 15:51 .
drwx----- 7 alien alien 4096 Jan 26 15:51 ..
-rwxr-x--- 1 alien alien 68192 Jan 26 15:51 command
-rw-r----- 1 alien alien 225 Jan 26 15:51 resources
-rw-r----- 1 alien alien 0 Jan 26 15:51 stderr
-rw-r----- 1 alien alien 115 Jan 26 15:51 stdout
aliensh:[alice] [84] /alice/cern.ch/user/a/alienmas/ >
```

```
aliensh:[alice] [85] /alice/cern.ch/user/a/alienmas/ >spy 41226891 stdout
Test: ClusterMonitor is at glcc0.ucllnl.org:8084
Execution machine: glcc1.ucllnl.org
"an example of a program output"
aliensh:[alice] [86] /alice/cern.ch/user/a/alienmas/ >
```

**BUT:** Never spy on large files, e.g. never spy on a \*.root file



# Job Status V – Master Jobs

*masterjob* **<JOB-Id>** will print a status of the sub jobs of the specified master job

```
aliensh:[alice] [2] /alice/cern.ch/user/a/alienmas/ >masterjob 42080407
Feb 17 22:32:40 info Checking the masterjob of 42080407
Feb 17 22:32:40 info The job 42080407 is in status: SPLIT
It has the following subjobs:
                Subjobs in DONE: 100
                Subjobs in WAITING: 1

In total, there are 101 subjobs
aliensh:[alice] [3] /alice/cern.ch/user/a/alienmas/ >█
```

... this is only interesting/working, if you have a job that splits



# Job Control – Kill a Job

You can also kill a job while it is running with: *kill <Job-ID>*

```
aliensh:[alice] [4] /alice/cern.ch/user/a/alienmas/ >submit TestJob.jdl
Submit submit TestJob.jdl
submit: Your new job ID is 42138635
aliensh:[alice] [5] /alice/cern.ch/user/a/alienmas/ >ps
  alienmas 42138635  _  SV          testjob
aliensh:[alice] [6] /alice/cern.ch/user/a/alienmas/ >kill 42138635
Process 42138635 killed!!
aliensh:[alice] [7] /alice/cern.ch/user/a/alienmas/ >
```





# Submitting/Running Jobs - Problems

If everything is ok with your JDL then your job is submitted and a <JOBID> is assigned to it.

You get a submission error message, e.g. if

- Your JDL contains errors , e.g. the syntax is not correct
- A file listed in the JDL is missing
- A package defined in the JDL is not listed in the packman



# Creating File Collections

With find you can create XML collections of files:

***find -x <Coll-Name> <Path-To-Search> <Search-Tag> > <Local-Outp.-File>***

Don't forget the output file is local on you machine, you need to upload it.

```
aliensh:[alice] [9] /alice/cern.ch/user/a/alienmas/ >find -x testColl ~/ *.Test > testcollection.xml
aliensh:[alice] [10] /alice/cern.ch/user/a/alienmas/ >cp file:testcollection.xml testcollection.xml
[xrootd] Total 0.00 MB |=====| 100.00 % [inf MB/s]
aliensh:[alice] [11] /alice/cern.ch/user/a/alienmas/ >cat testcollection.xml
<?xml version="1.0"?>
<alien>
  <collection name="testColl">
    <event name="1">
      <file name="123.Test" aclId="" broken="0" ctime="2010-02-18 08:29:59" dir="2693" entryId="2969" expiretime="
" gowner="alienmas" guid="21AC510C-1C5F-11DF-9B2E-0025B3E7369C" guidtime="11DF1C5F" lfn="/alice/cern.ch/user/a/ali
enmas/123.Test" md5="78a55619a8576bca2d203e078486f2de" owner="sschrein" perm="755" replicated="0" size="169" turl=
"alien:///alice/cern.ch/user/a/alienmas/123.Test" type="f" />
    </event>
    <event name="2">
      <file name="one.Test" aclId="" broken="0" ctime="2010-02-18 08:29:43" dir="2693" entryId="2968" expiretime="
" gowner="alienmas" guid="2630421A-1C5F-11DF-A1D9-0025B3E7369C" guidtime="11DF1C5F" lfn="/alice/cern.ch/user/a/ali
enmas/one.Test" md5="78a55619a8576bca2d203e078486f2de" owner="sschrein" perm="755" replicated="0" size="169" turl=
```



# Files and Grid Jobs – Try it out

- Do the following ...

```
cp /alice/cern.ch/user/s/sschrein/bin/tut_testjob.sh ~/bin
```

```
cp /alice/cern.ch/user/s/sschrein/tutorial/tutorial.jdl ~/tutorial
```

```
cp /alice/cern.ch/user/s/sschrein/tutorial/tut_validation.sh ~/tutorial
```

Fix the file locations inside tutorial.jdl ( each "s/sschrein" to your user's folder)

- Now submit the job

```
submit ~/tutorial/tutorial.jdl
```

- You should get an error, the JDL file has syntax problems

-> find the errors and correct them !

- Submit the job again
- Follow the job's stages





# Files and Grid Jobs – Try it out

- Once the job has finished with <DONE>
- Go to  

```
cd ~/tutorial/output
```
- Check what the job did, where it was running, the files and their content.

That's supposed to be it. Mission accomplished!



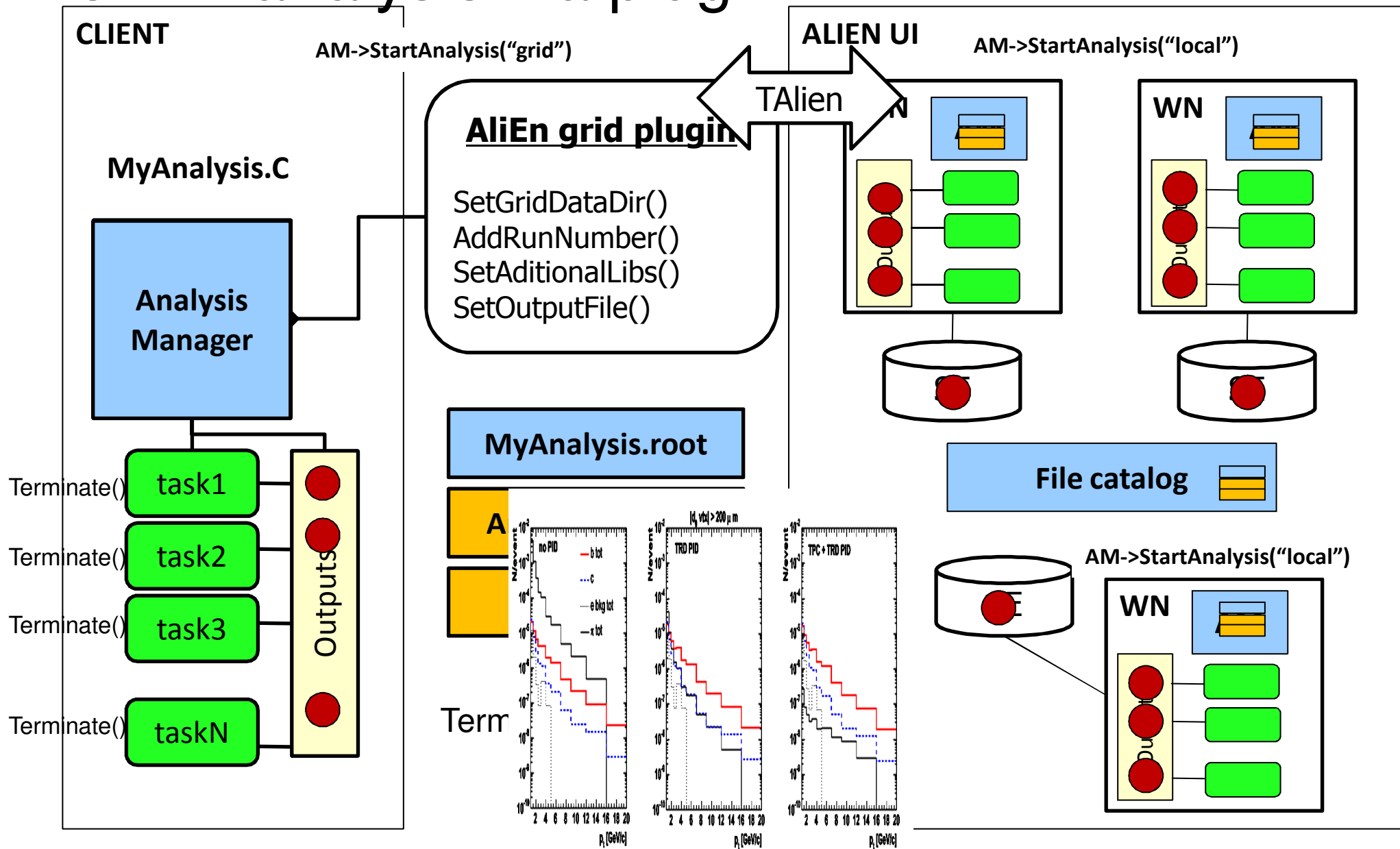


# A helper for AliEn analysis

- Works as a plugin for the analysis manager (as event handlers)
  - One has to create and configure a AliAnalysisAlien object
  - See:  
<http://aliceinfo.cern.ch/Offline/Activities/Analysis/AnalysisFramework/AlienPlugin.html>
- Creates dataset, JDL, analysis macro, execution+validation scripts
- Submits your job and merges the results



# GRID analysis via plugin





# Important plug-in settings

- ***plugin->SetRunMode(const char \*mode)***
  - “full”: generate files, copy in grid, submit, merge
  - “offline”: generate files, user can change them
  - “submit”: copy files in grid, submit, merge
  - “terminate”: merge available results
  - “test”: generate files + a small dataset, run locally as a remote job
    - `plugin->SetNtestFiles(Int_t nfiles)` – default 1
- ***plugin->SetROOTVersion(const char \*rootver)***
- ***plugin->SetAliRootVersion(const char \*alrootver)***
  - Change whenever needed
  - See command: `aliensh[] packages`



# Describing the input data

- ***plugin->SetGridDataDir(const char \*datadir)***
  - Put here the alien path before run numbers
  - See [pcalimonitor.cern.ch](http://pcalimonitor.cern.ch) for relevant data paths
- ***plugin->SetDataPattern(const char \*pattern)***
  - Use uniquely identifying patterns
    - i.e. ***\*/pass3/\*/AliESDs.root***
  - Plugin supports making datasets on ESD, ESD tags or AOD
- ***plugin->SetRunRange(Int\_t min, Int\_t max)***
  - Sets the run range to be analyzed
  - Enumeration of run numbers allowed
  - For existing data collections, use **AddDataFile()**
- ***Plugin->SetRunPrefix("000")***
  - To be used for real data





# Describing the output

- ***plugin->SetGridOutputDir(const char \*dir)***
  - Can be absolute AliEn FC path (/alien/cern.ch/...) or relative to work directory (no slashes)
- ***plugin->SetOutputFiles("file1 file2 ...");***
  - Allows a selection of files among the analysis outputs
- ***plugin->SetDefaultOutputs()***
  - Enables all outputs of the tasks connected to the analysis manager
- ***plugin->SetOutputArchive("log\_archive.zip:stderr,stdout root\_archive.zip:\*.root@disk=2");***
  - Will save the standard output/error in a zip and all root files in another zip replicated in 2 storage elements
  - **Note:** If archiving the output you may want to omit declaring the output files



# Other settings

- Using par files
  - ***plugin->EnablePackage("package.par")***
- Using other external libraries available in AliEn
  - ***plugin->AddExternalPackage("fastjet::v2.4.0")***
- Compiling single source files
  - ***plugin->SetAnalysisSource("mySource.cxx")***
  - But files have to be uploaded to AliEn from current directory
  - ***plugin->SetAdditionalLibs("libJETAN.so mySource.cxx mySource.h")***
    - Extra libraries to be loaded (besides AF ones) have to be enumerated in the same method.



# Optional settings

- Number of files per job
  - ***plugin->SetSplitMaxInputFileNumber(Int\_t n);***
- Number of runs per master job
  - ***plugin->SetNrunsPerMaster(Int\_t n)***
- Number of files to merge in a chunk
  - ***plugin->SetMaxMergeFiles(Int\_t n)***
- Resubmit threshold
  - ***plugin->SetMasterResubmitThreshold(Int\_t percentage)***
- Process a single run per job and output to a single directory
  - ***plugin->SetOutputSingleFolder(const char \*folder)***



# Configuring and running the AliEn plugin

- Open CreateAlienHandler.C
- Change working/output directories
- Modify number of files/worker
- Make sure the run mode is set to "full"
- Run macro runGrid.C
- Inspect the job status
- Modify the run mode to "terminate" once job finished
- Run again runGrid.C





# References I

- AliEn Website with further documentation:
  - <http://alien2.cern.ch>
- ALICE Analysis User Guide:
  - <http://project-arda-dev.web.cern.ch/project-arda-dev/alice/apiservice/AA-UserGuide-0.0m.pdf>