Job Management

Bordeaux, 28-29/01/2015

Overview

DIRAC JDL

DIRAC Commands

Tutorial Exercises

What do you have learned?

Job Description Language (JDL)

JDL is the language used to specify the characteristics and resources required by the job.

For example:

- Executable to run and its parameters.
- Files to be moved to and from the Worker Node
  - Input files
  - Grid files
- and any requirements.

- JDL is a *de facto* Grid standard for job description

Differences with gLite JDLs

DIRAC is not using JDLs internally, DIRAC transforms the user JDL to DIRAC internal conventions.

- Using DIRAC specific matching mechanism for job to resources mapping

Not all the JDLs attributes are allowed in DIRAC.

- In particular, there is no Requirements JDL attribute.

Some attributes are interpreted as requirements

- E.g. Site, CPUTime

JDLs

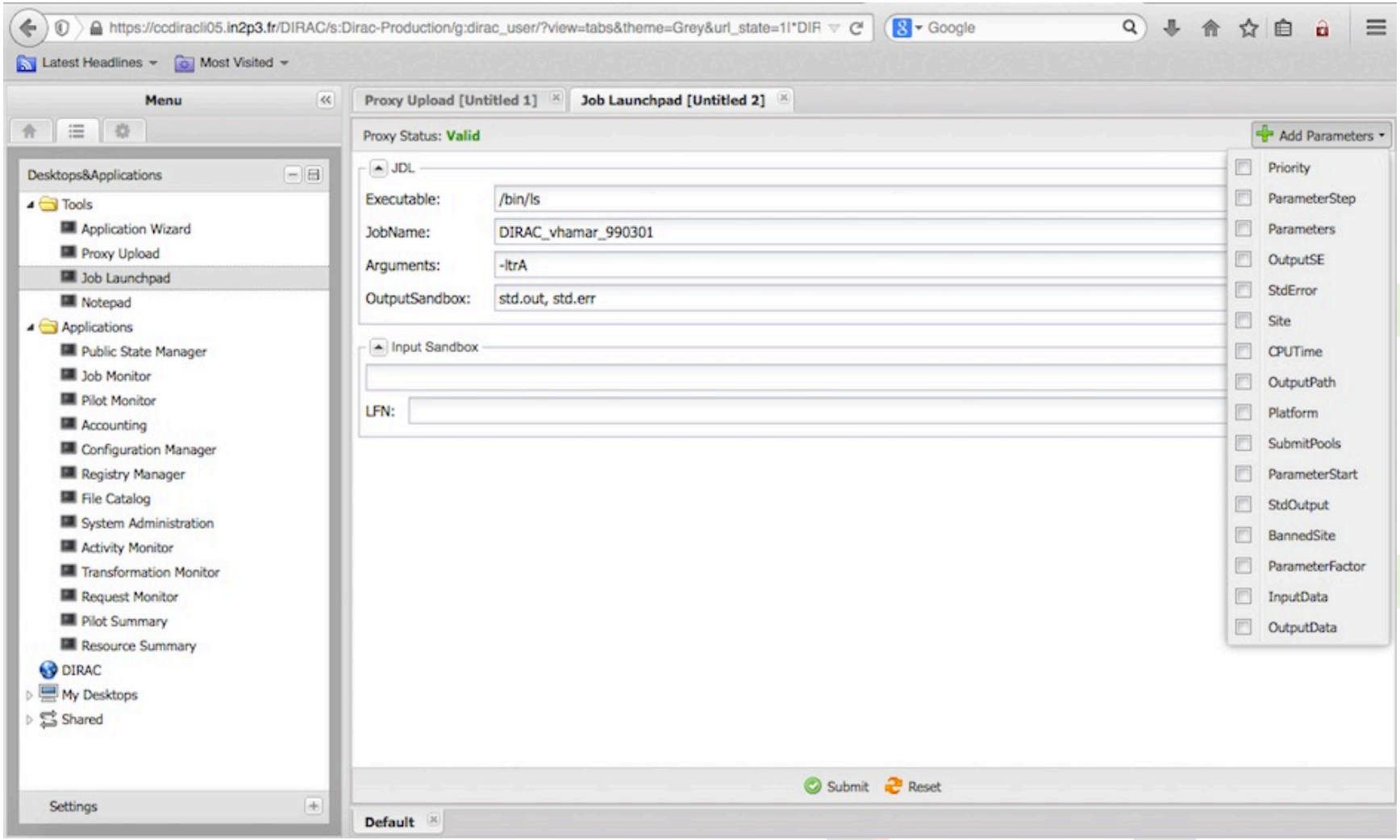
The JDL syntax consists of statements ended by a semicolon, like:

```
attribute = value;
```

Literal strings (for values) are enclosed in double quotes.

```
Site = "LCG.CPPM.fr";
```

JDLs

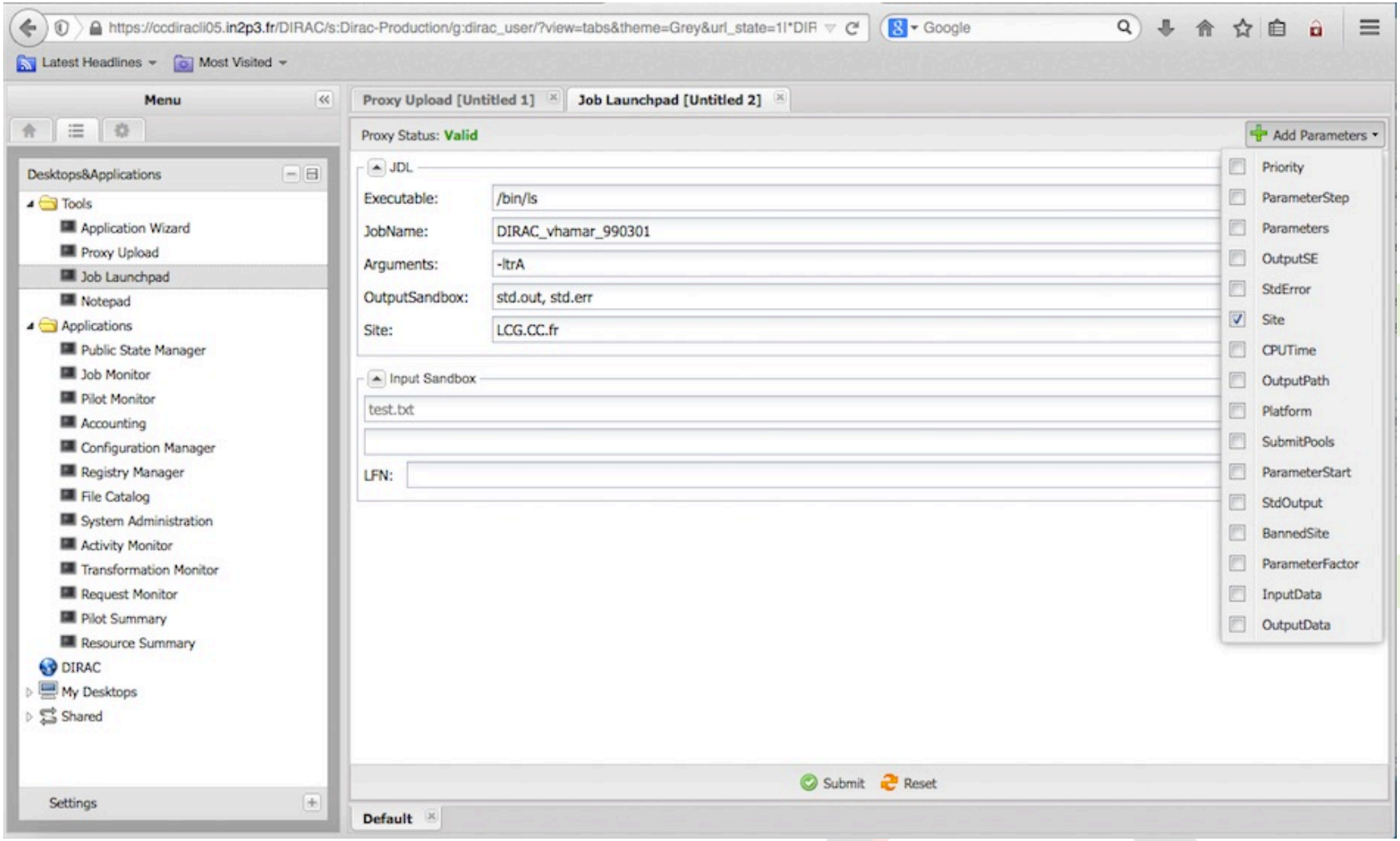

https://ccdiracli05.in2p3.fr/DIRAC/s:Dirac-Production/g:dirac_user/?view=tabs&theme=Grey&url_state=1|*DIR
Google
Latest Headlines
Most Visited
Menu
Desktops&Applications
Tools
Application Wizard
Proxy Upload
Job Launchpad
Notepad
Applications
Public State Manager
Job Monitor
Pilot Monitor
Accounting
Configuration Manager
Registry Manager
File Catalog
System Administration
Activity Monitor
Transformation Monitor
Request Monitor
Pilot Summary
Resource Summary
DIRAC
My Desktops
Shared
Settings
Proxy Upload [Untitled 1]
Job Launchpad [Untitled 2]
Proxy Status: Valid
Add Parameters
JDL
Executable: /bin/ls
JobName: DIRAC_vhamar_990301
Arguments: -ltrA
OutputSandbox: std.out, std.err
Input Sandbox
LFN:
Priority
ParameterStep
Parameters
OutputSE
StdError
Site
CPUTime
OutputPath
Platform
SubmitPools
ParameterStart
StdOutput
BannedSite
ParameterFactor
InputData
OutputData
Submit
Reset
Default

Example
Simple JDL

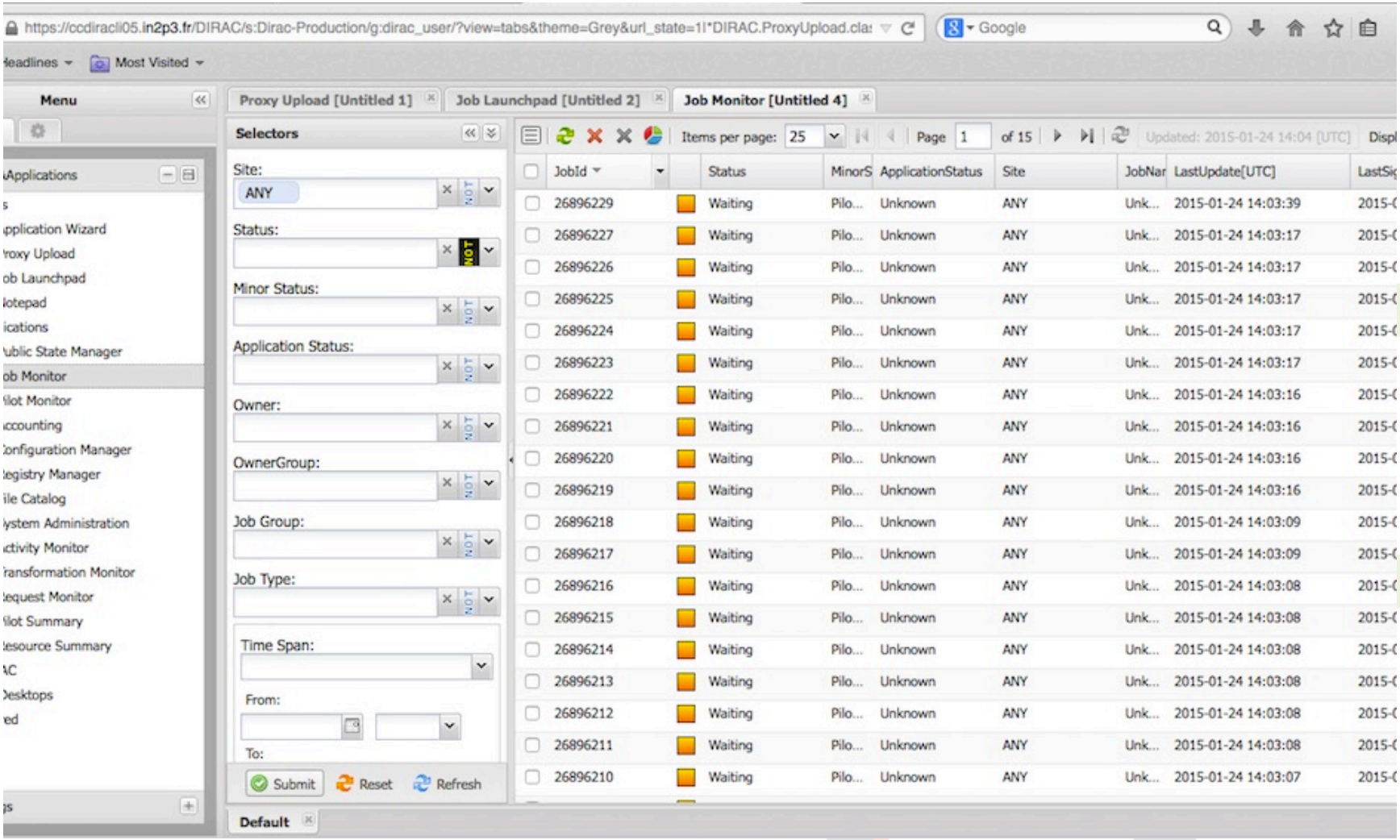
```
[
JobName = "Simple_Job";
Executable = "/bin/ls"; *
Arguments = "-ltr";
StdOutput = "StdOut";   *
StdError = "StdErr";    *
OutputSandbox = {"StdOut","StdErr"}; *
]
```

* Mandatory attributes

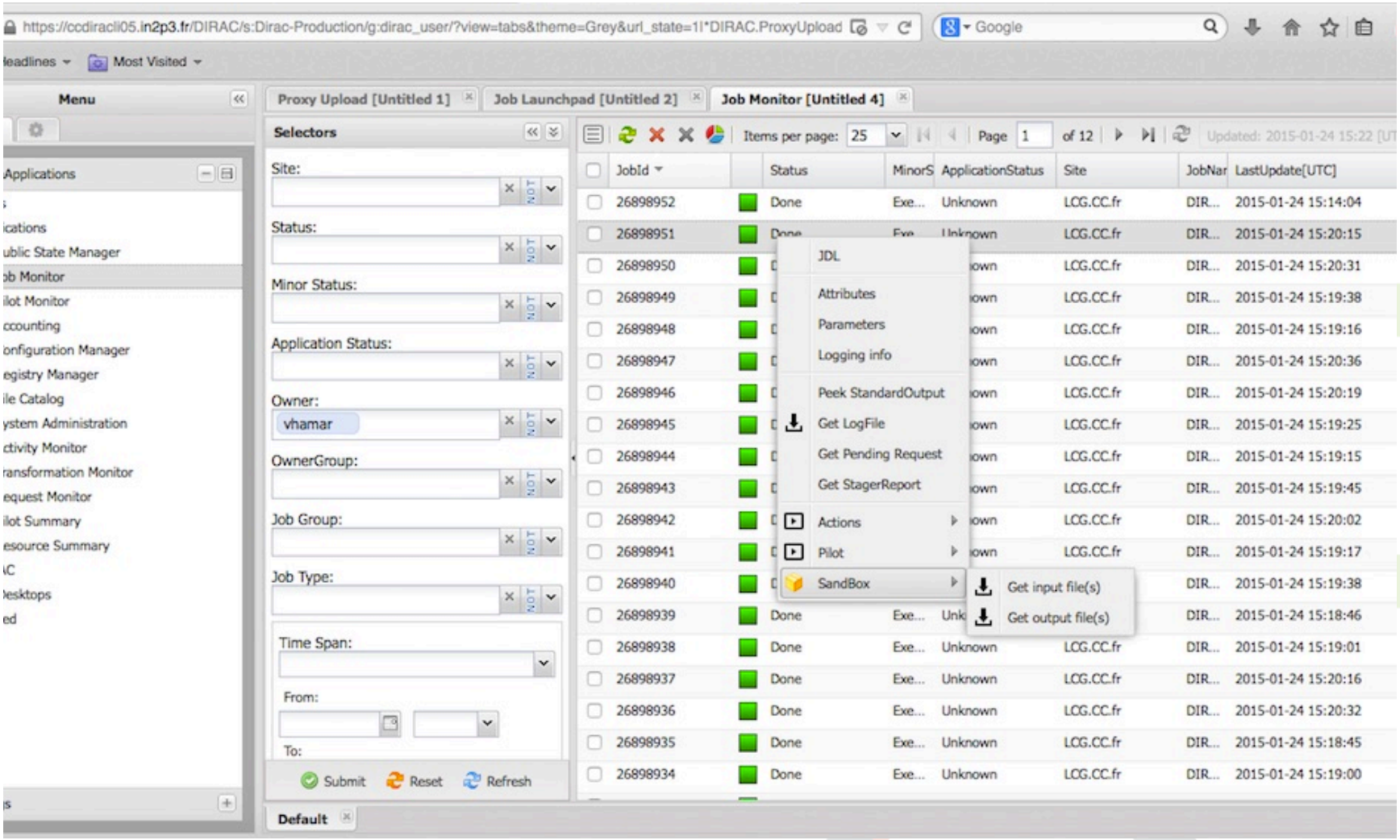
Example
Simple JDL



Job Monitoring



Job Retrieval



DIRAC Commands

In the directory *scripts* of your DIRAC installation there are a lot of commands that can be used for different purposes:

- Job Management
- Data Management
- Proxy Management
- Administration of the system

DIRAC Commands

DIRAC WMS	DIRAC ADMIN
dirac-wms-job-attributes	dirac-admin-add-group
dirac-wms-job-delete	dirac-admin-add-host
dirac-wms-job-get-input	dirac-admin-add-site
dirac-wms-job-get-jdl	dirac-admin-add-user
dirac-wms-job-get-output	dirac-admin-allow-catalog
dirac-wms-job-get-output-data	dirac-admin-allow-se
dirac-wms-job-kill	dirac-admin-allow-site
dirac-wms-job-logging-info	dirac-admin-ban-catalog
dirac-wms-job-parameters	dirac-admin-ban-se
dirac-wms-job-peek	dirac-admin-ban-site
dirac-wms-job-reschedule	dirac-admin-bdii-ce
dirac-wms-job-status	dirac-admin-bdii-site
dirac-wms-job-submit	dirac-admin-ce-info
dirac-wms-jobs-select-output-search	dirac-admin-delete-user
dirac-wms-select-jobs	dirac-admin-externals-versions

DIRAC Commands

DIRAC DMS

dirac-dms-catalog-metadata
dirac-dms-change-replica-status
dirac-dms-check-file-integrity
dirac-dms-clean-directory
dirac-dms-data-size
dirac-dms-filecatalog-cli
dirac-dms-fts-monitor
dirac-dms-fts-submit
dirac-dms-get-file
dirac-dms-get-storage-directories
dirac-dms-lfn-accessURL
dirac-dms-lfn-logging-info
dirac-dms-lfn-metadata
dirac-dms-lfn-replicas
dirac-dms-pfn-accessURL
dirac-dms-pfn-metadata
dirac-dms-remove-catalog-files

DIRAC PROXY

dirac-proxy-get-uploaded-info
dirac-proxy-info
dirac-proxy-init
dirac-proxy-upload

DIRAC UTILITIES

dirac-platform
dirac-rss-reassign-token
dirac-rss-renew-token
dirac-service

COMDIRAC Job Submission

We will be using COMDIRAC style commands

```
dsub /bin/hostname
```

```
26897474
```

Job ID

```
dsub –J Simple.jdl
```

```
26897538
```

Job ID

```
Options:
 -J  --JDL <value>          : JDL file or inline
 -N  --JobName <value>      : job name
 -E  --StdError <value>     : job standard error file
 -O  --StdOutput <value>    : job standard output file
 -   --OutputSandbox=       : job output sandbox
 -   --InputSandbox=        : job input sandbox
 -   --OutputData=          : job output data
 -   --InputData=           : job input data
 -   --OutputPath=          : job output data path prefix
 -   --OutputSE=            : job output data SE
 -   --CPUTime=             : job CPU time limit (in seconds)
 -   --Site=                : job Site list
 -   --BannedSite=          : job Site exclusion list
 -   --Platform=            : job Platform list
 -   --Priority=            : job priority
 -   --JobGroup=            : job JobGroup
 -   --Parametric=          : comma separated list or named parameters or integer
                              loops (in the form<start>:<stop>[:<step>])
 -   --ForceExecUpload      : Force upload of executable with InputSandbox
 -v  --verbose              : verbose output
```

COMDIRAC
Job Submission

```
>dsub
echo Hello from `/bin/hostname -f`
<Crtl d>
26898185
```

COMDIRAC Job Status

>dstat

```
  JobID    Owner    JobName    OwnerGroup JobGroup      Site      Status       MinorStatus         SubmissionTime

-------- ------ ---------- ---------- -------- ------------ ------- ---------------------- -------------------

26897824 vhamar testJob.sh dirac_user NoGroup  LCG.IN2P3.fr Waiting Pilot Agent Submission 2015-01-24 14:48:45
```

>dlogging 26898185

Status	MinorStatus	ApplicationStatus	Time	Source
Received	Job accepted	Unknown	2015-01-24 14:59:12	JobManager
Checking	JobSanity	Unknown	2015-01-24 14:59:12	JobPath
Checking	JobScheduling	Unknown	2015-01-24 14:59:12	JobSanity
Waiting	Pilot Agent Submission	Unknown	2015-01-24 14:59:12	JobScheduling

COMDIRAC
Job Retrieval

```
>doutput 26898185
```

```
>cd 26898185/
```

- Options:

-D --OutputDir <value>: destination directory

--Data: donwload output data instead of output sandbox

--Sandbox: donwload output sandbox, even if data was required

-v --verbose: verbose output

-n --NoJobDir: do not create job directory

-g --JobGroup <value>: Get output for jobs in the given group

DIRAC Commands: Common options

All the DIRAC commands are following the UNIX notation for passing arguments to the command

Some options are common to all the commands:

-o to pass any configuration option, e.g.

```
-o /DIRAC/Setup=DIRAC-Production
```

-d, -dd, -ddd to choose various levels of the output verbosity

-h to show help message for the command

In the following there are some examples of commonly used commands

Tutorial

- Submit “Hello World” job with Web Launchpad
  - Monitor it
  - Get results in the Web browser
- Submit “Hello World” job with command line tools and COMDIRAC
  - Edit JDL
  - Monitor it
  - Get results with command line tools
- Goals
  - Understand job description
  - Understanding job Web monitoring and manipulation tools

Useful links

- DIRAC Web Portal

https://dirac.in2p3.fr

- Job Submission

http://diracgrid.org/files/docs/UserGuide/GettingStarted/UserJobs/index.html

- COMDIRAC Job Management

https://github.com/DIRACGrid/COMDIRAC/wiki/Job-Management

Back-up slides

DIRAC Commands: Sending Jobs

- The command used to submit jobs using DIRAC is:

```
dirac-wms-job-submit  <JDL>
```

```
$ dirac-wms-job-submit Simple.jdl
JobID = 11
```

Dirac Commands: Job Status

The command used to get the job status is:

dirac-wms-job-status <Job_ID>

```
$ dirac-wms-job-status 11
  JobID=11 Status=Waiting;
  MinorStatus=Pilot Agent Submission;
  Site=ANY;
```

Dirac Commands: Job Retrieval

To retrieve job outputs use the command:

```
dirac-wms-job-get-output [ --dir output_dir] <Job_ID> [<Job_ID>]
```

```
$ dirac-wms-job-get-output --dir this_dir 470 469 468
$ ls -la this_dir/
total 20
drwxr-xr-x 5 hamar dirac 4096 Oct 26 19:02 .
drwxr-xr-x 3 hamar dirac 4096 Oct 26 19:02 ..
drwxr-xr-x 2 hamar dirac 4096 Oct 26 19:02 468
drwxr-xr-x 2 hamar dirac 4096 Oct 26 19:02 469
drwxr-xr-x 2 hamar dirac 4096 Oct 26 19:02 470
```