

# Bulk job submission

DIRAC Project



A parametric job consists in submission of a set of jobs where only one parameter make the difference between the jobs.

Parametric jobs share the same Input Sandbox

The parametric job must define a JDL attribute **"Parameters"**. It can take the following values:

A list (strings or numbers).

▶ `Parameters = {"value1", "value2", "value3"};`

Or, an integer specifying the number of parameters to generate, in this case the JDL attributes *ParameterStart* and *ParameterStep/ParameterFactor* must be defined in order to create the sequence of values:

$$P_0 = \text{ParameterStart}$$

$$P_i = P_{i-1} * \text{ParameterFactor} + \text{ParameterStep}$$

The **%s** placeholder is substituted by the parameter value and can be put in any JDL attributes.

Not all the attributes can be parameterized this way, e.g. Input Sandbox files can not be parameterized.

The **%n** placeholder is replaced by the parameter sequential number

Useful to be used in job and file names

Left padded with zeros to make names alphabetically sortable

The **%j** placeholder is replaced by the DIRAC job ID

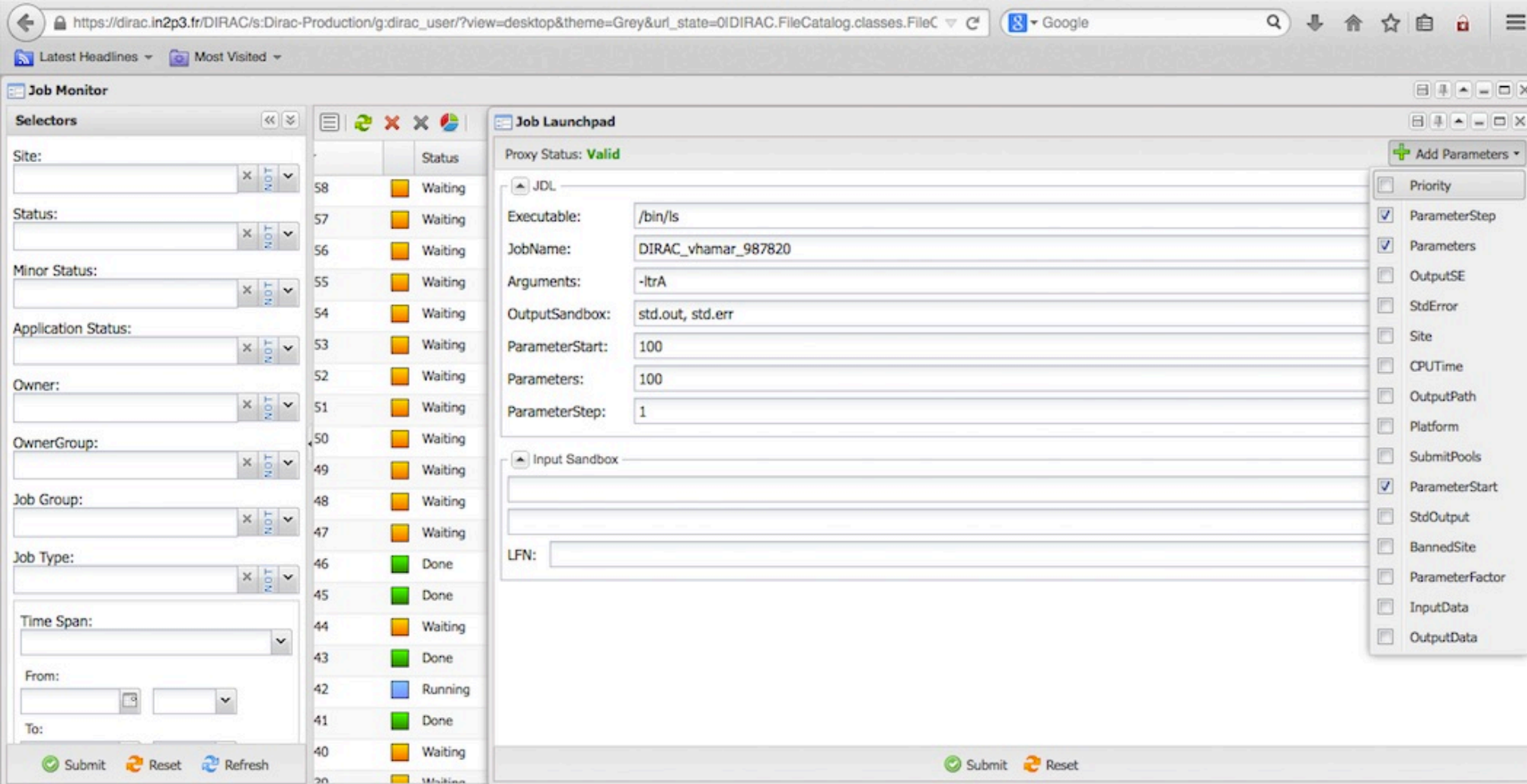
Can be useful in the OutputPath JDL attribute

```
Executable = "testParametricJob.sh";  
JobName = "Parametric_ %n";  
Arguments = "%s";  
Parameters = 20;  
ParameterStart = 0;  
ParameterStep = 0.02;  
ParameterFactor = 1;  
StdOutput = "StdOut_ %j";  
StdError = "StdErr_ %j";  
InputSandbox = {"testJob.sh"};  
OutputSandbox = {"StdOut_ %j", "StdErr_ %j"};
```

Placeholder replaced by Parameter number value for each job

Placeholder replaced by Parameter value for each job

Placeholder replaced by Parameter value for each job



The screenshot shows the DIRAC Job Monitor and Job Launchpad interface. The Job Monitor on the left displays a list of jobs with their status (Waiting, Done, Running) and a 'Submit' button. The Job Launchpad on the right shows the configuration for a parametric job, including the executable, job name, arguments, output sandbox, and parameter start/step values. A 'Submit' button is also present at the bottom of the Job Launchpad.

**Job Monitor**

Job ID	Status
58	Waiting
57	Waiting
56	Waiting
55	Waiting
54	Waiting
53	Waiting
52	Waiting
51	Waiting
50	Waiting
49	Waiting
48	Waiting
47	Waiting
46	Done
45	Done
44	Waiting
43	Done
42	Running
41	Done
40	Waiting
39	Waiting

**Job Launchpad**

Proxy Status: **Valid**

**JDL**

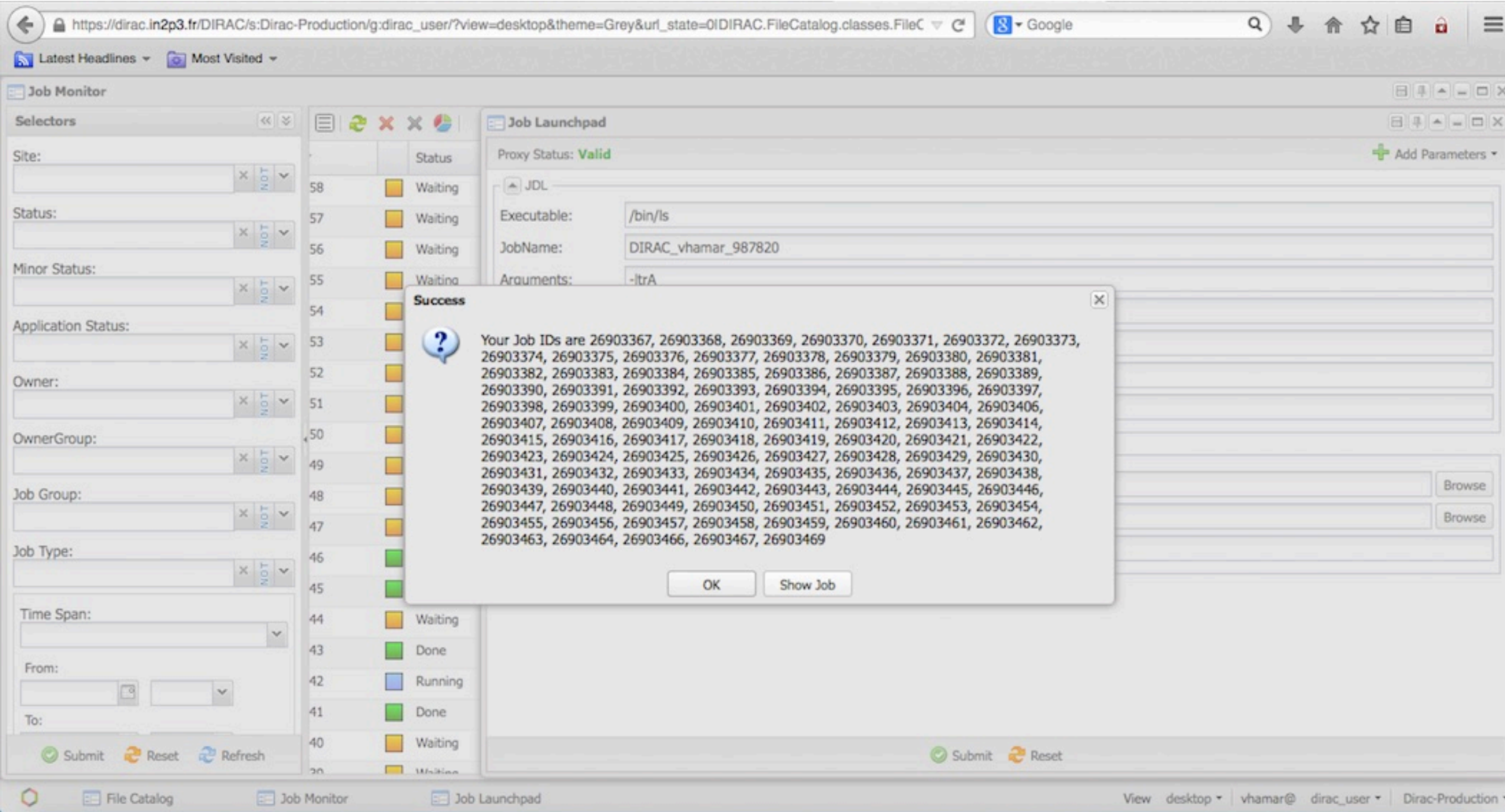
Executable: /bin/ls  
JobName: DIRAC\_vhamar\_987820  
Arguments: -ltrA  
OutputSandbox: std.out, std.err  
ParameterStart: 100  
Parameters: 100  
ParameterStep: 1

**Input Sandbox**

LFN:

**Submit Parameters**

- Priority
- ParameterStep
- Parameters
- OutputSE
- StdError
- Site
- CPUTime
- OutputPath
- Platform
- SubmitPools
- ParameterStart
- StdOutput
- BannedSite
- ParameterFactor
- InputData
- OutputData



The screenshot displays the DIRAC Job Monitor web interface. A 'Success' dialog box is open, displaying a list of Job IDs. The background interface includes a 'Job Launchpad' with a 'JDL' section containing fields for Executable, JobName, and Arguments. The 'Job Monitor' panel on the left shows a list of jobs with their status (Waiting, Done, Running) and a 'Time Span' section. The 'Job Launchpad' also has 'Submit' and 'Reset' buttons at the bottom.

**Success**

Your Job IDs are 26903367, 26903368, 26903369, 26903370, 26903371, 26903372, 26903373, 26903374, 26903375, 26903376, 26903377, 26903378, 26903379, 26903380, 26903381, 26903382, 26903383, 26903384, 26903385, 26903386, 26903387, 26903388, 26903389, 26903390, 26903391, 26903392, 26903393, 26903394, 26903395, 26903396, 26903397, 26903398, 26903399, 26903400, 26903401, 26903402, 26903403, 26903404, 26903406, 26903407, 26903408, 26903409, 26903410, 26903411, 26903412, 26903413, 26903414, 26903415, 26903416, 26903417, 26903418, 26903419, 26903420, 26903421, 26903422, 26903423, 26903424, 26903425, 26903426, 26903427, 26903428, 26903429, 26903430, 26903431, 26903432, 26903433, 26903434, 26903435, 26903436, 26903437, 26903438, 26903439, 26903440, 26903441, 26903442, 26903443, 26903444, 26903445, 26903446, 26903447, 26903448, 26903449, 26903450, 26903451, 26903452, 26903453, 26903454, 26903455, 26903456, 26903457, 26903458, 26903459, 26903460, 26903461, 26903462, 26903463, 26903464, 26903466, 26903467, 26903469

OK Show Job

It is convenient to give a meaningful unique JobGroup parameter to a group of jobs submitted in one go, e.g. *MyJobGroup*:

Look up the status of the jobs in the group

```
dstat -g MyJobGroup
```

Store the output of all the jobs in the group in one directory

```
doutput -g MyJobGroup-n
```

# Mandelbrot quest in 3 commands

- ▶ JDL for bulk job submission with JobGroup defined:

```
Executable = "mandelbrot";  
...  
JobGroup = "MyMandelbrotQuest";  
Parameters = 100;  
...
```

- ▶ Submit
  - ▶ **dsub my\_mandelbrot.jdl**
- ▶ Get results
  - ▶ **doutput -g MyMandelbrotQuest -D mydir -n -t**
- ▶ Make the movie
  - ▶ **cd mydir**
  - ▶ **convert \*.bmp movie.gif**



## Submit series of “mandelbrot” jobs with Web Launchpad

Submit multiple jobs to explore the C seeds

Submit multiple jobs with increasing precision to make a series of frames for the Mandelbrot movie

Submit jobs to multiple sites, one per site

▶ Monitor jobs

Try to use COMDIRAC commands also

Store results in the grid SE

## Goals

Understand bulk job description

Understanding job input/output data