

Petrel TIPS&TRICKS from SCM

Knowledge Worth Sharing

Command Line Execution of Workflows

Petrel has become a standard tool for field management and well planning. Existing reservoir models are constantly being updated as new data are added and interpretations refined. The most efficient way to update a model is to use a workflow or series of workflows. SCM uses workflows as a standard part of the model building process on all projects. These workflows do several things:

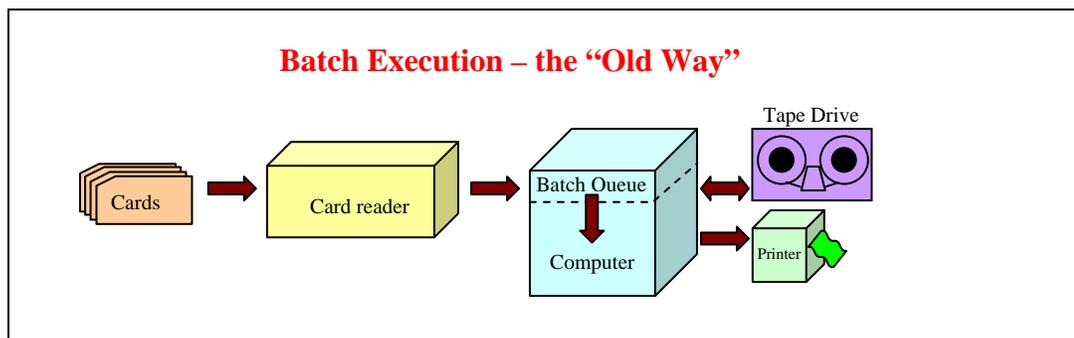
- Allow steps to be run many times, enabling you to fine tune them until the desired result is achieved.
- Forces clear organization of the entire modeling process.
- Documents the steps that were run and parameters used. This is absolutely required for the person who built the model and others who will use or update it, as no one can remember 1000s of parameter settings.
- Allows the entire modeling process to be rerun in hours (it possibly took days to do the first time) when new data are added.

For those new to Petrel’s Process Manager (the tool used to build workflows), you will probably want to take a course that introduces you to syntax and workflow building techniques. Building a workflow is not intuitive and a little coaching will save you days. The classes are usually one day long and if you really want to speed the learning curve, mentors can be hired to help you create the first set of workflows for your project.

For those who are regular users of the Process Manager, the following discussion describes execution of workflows from the command line, a technique that is not commonly applied and that we have found useful. SCM trainers learned these tricks by calling the SLB help line, browsing the documentation, and a bit of brute-force experimentation.

Batch Execution

Old-timers who started their computer careers using punch cards to input commands to large mainframe computers and tape drives to store and retrieve data will know what is truly meant by the term Batch Execution. In that environment, computer runs (batch jobs) were entered using a card reader. That job was queued up to run behind other jobs that had been entered previously. If your job was extremely important or you knew the person running the computer you could get it kicked to a higher priority level and jump to the head of the queue. When complete, the results would be printed or plotted or tapes written. Problems were fixed (new cards punched) and the job resubmitted for execution.



Petrel is a mark of Schlumberger

This seems a bit archaic by today's standards; however, the ability to kick off a job and then move on to working another part of the project while the job executes is appealing and an efficient use of time. The appeal goes up exponentially as the time it takes the job to execute goes up. Unfortunately, Petrel does not have batch execution capability. Once a workflow is submitted for execution the entire Petrel project is tied up and nothing else can be done. Petrel really has a run-while-you-wait capability and you can't do anything until the workflow finishes. This is the way the program architecture works and it will not change.

So why are we still talking about "batch like" execution for Petrel? There are several reasons:

- You want to pass specific text or numeric variables into a workflow without having to edit the workflow.
- You want to set off many workflows, perhaps even dozens of workflows, in succession and you don't want the hassle of having to babysit the submission of each.
- You use the same project often and want a command line short cut that will start Petrel and open that project.

Required Input

To submit a workflow for "batch" execution requires:

- The directory path and name of the Petrel project
 - The project contains the workflow and it has been successfully executed
 - The project contains files needed by the workflow
 - The files needed by the workflow are linked to the workflow
- The instructions for starting Petrel on your computer
- The name of the workflow that is to be run
- A file containing instructions to start Petrel, open the project, and execute the workflow

Command Parameters for Running a Workflow

There are a number of instructions (parameters) required to run a workflow in "batch". You can either type the instructions into a DOS command window or put them into a command file and execute that file. Since directory paths are part of the instructions and they can be long and complex, it is best to put the instructions into a file. Use the standard PC Text editor to create a file of type .txt. When you have the file ready to go, save the file as a type .cmd, which stands for command file. A file of type .cmd can be executed by double clicking on its name.

The parameters for the command line are listed below along with descriptions and syntax. The Petrel help manual contains brief descriptions.

Starting Petrel

This parameter must be the first entry on the command line, the other parameters may occur in any order. This is the full path and command used to start Petrel on your machine. To find the command you either right click on the Petrel shortcut on your desk top and select properties or you navigate to the Petrel program using the *Start → All Programs →* approach and then right click and select properties. Highlight the command on the *Target* line and copy it to your file. The quotes will already be on the command.

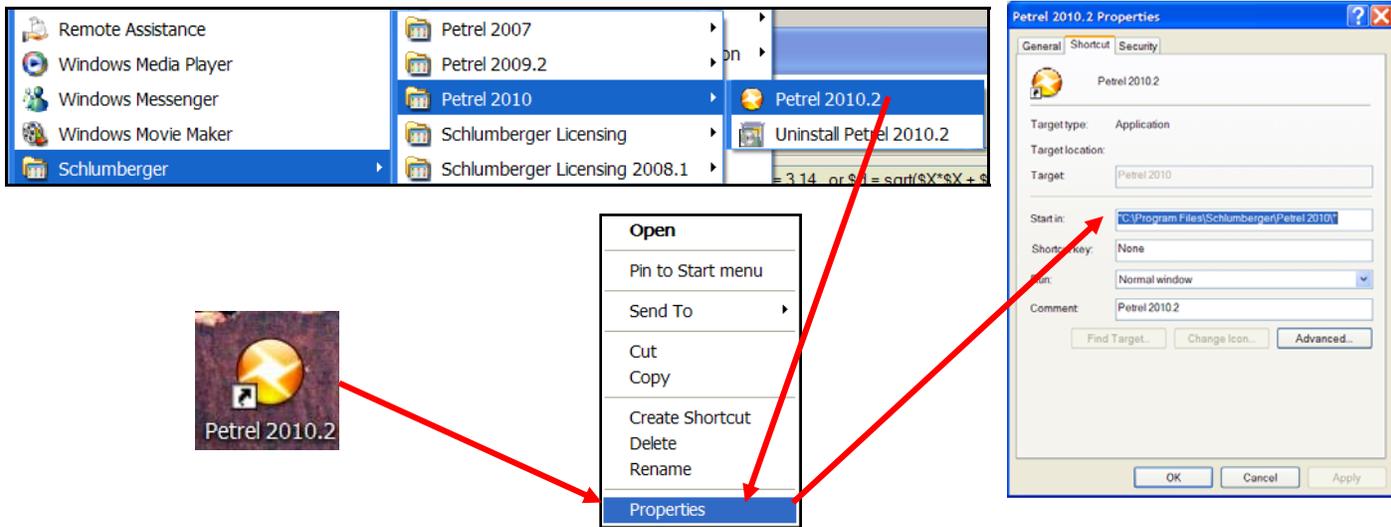
Petrel is a mark of Schlumberger



4801 Woodway Drive, Suite 150W • Houston, TX 77056 • www.scminc.com • info@scminc.com

© 2011 SCM E&P Solutions, Inc.

Figure: Right click on either the shortcut or the program and select *Properties* to see the Target (instructions for starting Petrel).



A typical start command might be:

```
"C:\Program Files\Schlumberger\Petrel 2010\Petrel.exe"
```

Usually it is enclosed in double quotes since it often contains spaces.

License Package

Just as when starting Petrel interactively you must tell it which license package you want to use. License packages are named and you must know and enter that name in the command line. The syntax to use is:

```
/licensePackage Package2
```

Where */licensePackage* (entered exactly) is the name of the parameter and *Package2* is the name of the license that is to be used. The names can be found on the *Select license package* window:



As when you select a license package interactively, you must select a package here that has the options necessary to run the commands in the workflow. If you want Petrel to bring up the entire *Select license package* window so you can interactively select a license, use the parameter */SelectLicense* instead of */licensePackage*. The syntax for this is:

```
/SelectLicense
```



Workflow or All Workflows

You can enter the name of one workflow or you can ask Petrel to execute, in a top-down order, all workflows found in the Workflow tab. Usually you will only execute one workflow per command line and you refer to that workflow by name.

The syntax to use is:

```
/runWorkflow "Calculate Isochore"
```

Where `/runWorkflow` (entered exactly) is the name of the parameter and `Calculate Isochore` is the name of the workflow to be executed. Put the workflow name in double quotes if it contains internal spaces.

If you are going to run all workflows in the Workflow tab, use the syntax:

```
/runAllWorkflows
```

Where `/runAllWorkflows` (entered exactly) is the name of the parameter and nothing follows it.

Numeric Variables

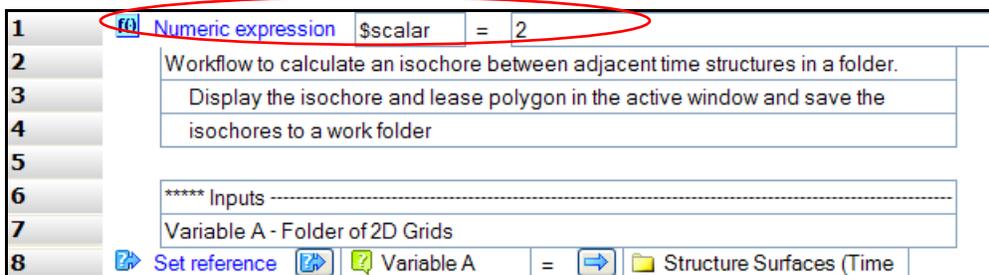
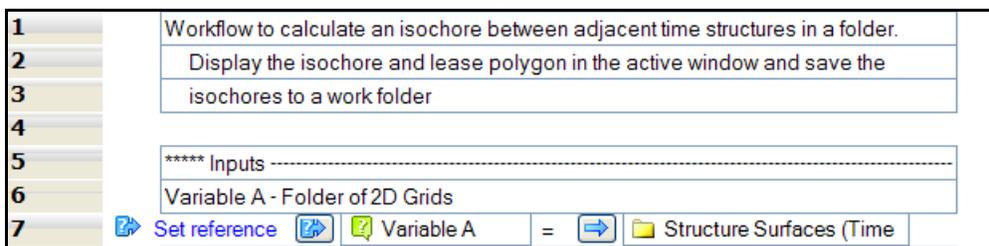
It is possible to define a numeric variable and its value on the command line. The program will copy the workflow and insert a *Numeric Expression* at the top of the copy with the variable and value you defined. You must be careful when using this approach as a workflow that accepts a numeric parameter from the command line will fail if executed interactively from within Petrel (because there is no *Numeric Expression* defining the variable). This is easily remedied by using a *Numeric Expression* in the workflow to define the variable but *Disabling* that line (gray it out) when using the command line approach to run the workflow.

The syntax to use is:

```
/nparm num=555
```

Where `/nparm` (entered exactly) is the name of the parameter, `num` is the variable in the workflow (note there is no dollar sign "\$") and `555` is the value to be used for the variable `num`.

Figure: Top of workflow before executing a command (top) and top of copied version of the workflow after executing from a command line containing `/nparm` (bottom). Note the addition of the *Numeric Expression* line. The line `/nparm scalar=2` was used, so the *Numeric Expression* added contains `$scalar`.



Petrel is a mark of Schlumberger



4801 Woodway Drive, Suite 150W • Houston, TX 77056 • www.scminc.com • info@scminc.com

© 2011 SCM E&P Solutions, Inc.

If you have more than one numeric variable to enter, you can make several inserts of the /nparm parameter. The syntax would be:

```
/nparm num=555 /nparm pie=3.14 /nparm num2=666
```

An alternate approach that is slightly error prone is to use only one /nparm followed by several variable=value sets each separated by a comma with no spaces before or after the comma. The syntax would be:

```
/nparm num=555,pie=3.1415,num2=666
```

As you can see this is hard to read and if a space is added before or after the comma, an error will be generated.

String Variables

It is possible to define a string variable and its value on the command line. Like the numeric variable a String Expression line containing this information is inserted at the top of a copied workflow before execution.

The syntax to use is:

```
/sparm text=aaa
```

Where /sparm (entered exactly) is the name of the parameter and text=aaa is what you entered for the variable and value.

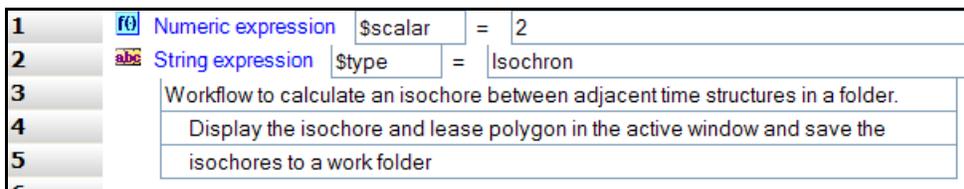
If the string contains internal blanks use double quotes around the entire expression. The syntax would be:

```
/sparm "text=aa a"
```

If you have more than one string variable to enter, you can make several inserts of the /sparm parameter. The syntax would be:

```
/sparm text=aaa /sparm word=bbb /sparm "text2=hi how are you"
```

Figure: String and numeric expressions added at the top of the workflow for /sparm and /nparm entries.



As with numeric variables, you can enter several string variable definitions separated by commas and no spaces as long as they are not quoted. If using quotes we have found that this comma style input produces errors. The syntax would be:

```
/sparm text=aaa,word=bbb,text2=ccc
```

Again, this is hard to read and error prone.

Plug-in Parameters

New to Petrel is the ability to add parameters for plug-ins. You may add either numeric or string parameters. For numeric parameters, instead of using the /nparm parameter, you use the /noption parameter. The syntax to use is:

```
/noption "plug=pnum=500"
```

This sets the parameter *pnum* to a value of 500 when running the plug-in named *plug*.

Similarly, to use a string parameter, use the /soption parameter instead of /sparm. The syntax is:

`/soption "plug=pname=My Data"`

This sets the parameter *pname* to a value of "My Data" when running the plug-in named *plug*.

No quotes are needed in the names if there are no internal spaces in either the parameter names or values.

Quiet

The *quiet* parameter allows Petrel to execute a workflow without opening the workflow's dialog. It still displays the primary Petrel window. The syntax is:

`/quiet`

Where */quiet* (entered exactly) is the name of the parameter and nothing follows it.

No splash screen

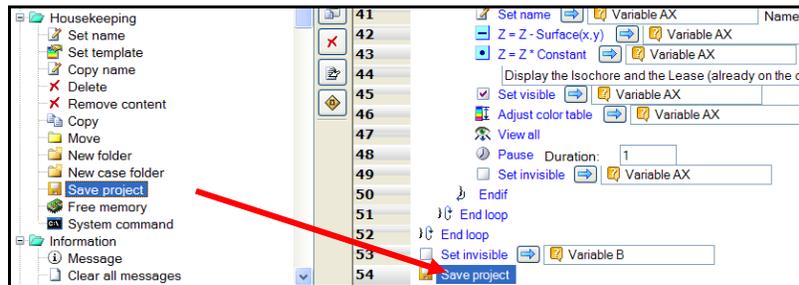
The *nosplashscreen* parameter doesn't allow Petrel to bring up the Petrel splashscreen when starting batch execution. If this parameter is not used, the splashscreen will appear even if /quiet or /exit is in the command line. To disable the splash screen, the syntax is:

`/nosplashscreen`

Where */nosplashscreen* (entered exactly) is the name of the parameter and nothing follows it.

Exit

The *exit* parameter will cause Petrel to close after the workflow executes. If you use this parameter, remember that it will NOT save any changes made by the workflow to the project. If you want to save changes made by the workflow you MUST add a *Save Project* line (*Utilities* button → *Housekeeping* folder) to the bottom of the workflow. This will save all actions done by the workflow before exiting.



If you are going to insert multiple batch execution commands, each executing one workflow, you will need an */exit* parameter in each command line.

The syntax for this parameter is:

`/exit`

Where */exit* (entered exactly) is the name of the parameter and nothing follows it.

Not saving before exiting is useful when the workflow builds data (spreadsheet) and exports that data to an external file. This gives you the opportunity to run without altering your project and test many scenarios.

Petrel is a mark of Schlumberger



4801 Woodway Drive, Suite 150W • Houston, TX 77056 • www.scminc.com • info@scminc.com

© 2011 SCM E&P Solutions, Inc.

Project

You must tell Petrel which project to open. This requires the full path name up to and including the .pet file name. The syntax to use is:

```
"T:\Tips and Tricks\2010_02\Batch_Tests.pet"
```

Note that there is no parameter name before the path and project name. Enclose this in quotes if there are any spaces in the name. This is usually the last parameter on the command line, but this isn't required.

Executing the Command Line

As mentioned above, the command instructions are:

- typed into a file of type .txt,
- that file is converted to a command file of type .cmd,
- and that cmd file is double clicked on to execute the workflow.

Following is an example of a command file for running a workflow once. It shows most of the parameters, including those that pass numbers and text into the workflow.

```
"C:\Program Files\Schlumberger\Petrel 2010\Petrel.exe" /licensePackage Package1 "T:\Tips and Tricks\2010_02\batch_tests.pet" /runWorkflow Batch_1 /nparm num=555 /sparm "text=aaaa aaaaaa" /quiet /exit
```

Following is a command file that is designed only to start Petrel and open a specific project. This would be used when the same project is always being opened, usually for viewing and well planning. A short cut would be created and used to run the command file.

```
"C:\Program Files\Schlumberger\Petrel 2010\Petrel.exe" /licensePackage Package1 "T:\Tips and Tricks\2010_02\batch_tests.pet"
```

Following is a command file that runs the same workflow three times. It passes a number and file name into the workflow and a file is output with that name. The workflow is run 3 times, each with different parameters.

```
"C:\Program Files\Schlumberger\Petrel 2010\Petrel.exe" /licensePackage Package1 "T:\Tips and Tricks\2010_02\batch_tests.pet" /runWorkflow Batch_1 /nparm OWC=-1950 /sparm "file=vols_for_owc_-1950" /exit
```

```
"C:\Program Files\Schlumberger\Petrel 2010\Petrel.exe" /licensePackage Package1 "T:\Tips and Tricks\2010_02\batch_tests.pet" /runWorkflow Batch_1 /nparm OWC=-1975 /sparm "file=vols_for_owc_-1975" /exit
```

```
"C:\Program Files\Schlumberger\Petrel 2010\Petrel.exe" /licensePackage Package1 "T:\Tips and Tricks\2010_02\batch_tests.pet" /runWorkflow Batch_1 /nparm OWC=-2000 /sparm "file=vols_for_owc_-2000" /exit
```

Petrel is a mark of Schlumberger

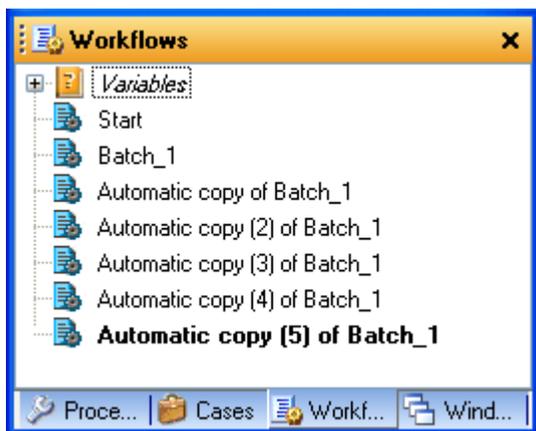


4801 Woodway Drive, Suite 150W • Houston, TX 77056 • www.scminc.com • info@scminc.com

© 2011 SCM E&P Solutions, Inc.

As mentioned when discussing the *nparm* and *sparm* parameters, a copy of the workflow is created each time petrel runs. These are saved if there is a *Save Project* line at the bottom of the workflow. These copies are easily deleted when no longer needed and provide an audit trail of what was executed. If the workflow does not have a *Save Project* line in it then these copies will not be saved. Not saving is useful when many runs are made and the exported results are needed but not the changes to the project.

Figure: Workflow tab showing the original workflow and several copies of it, each from a different command line execution.



Command Line continuation

A typical Petrel batch command can be very long. An example is:

```
"C:\Program Files\Schlumberger\Petrel 2010\Petrel.exe" /licensePackage Package2 /runWorkflow "Calculate Isochore" /nparm scalar=2 /sparm "type=Isochore" /quiet /nosplashscreen /exit "C:\Documents and Settings\DPM Documents\Petrel Projects\Batch_Tests.pet"
```

To keep it more readable, use the "carat" (^) character at the end of a line. This signals DOS that the command continues on the next line. Thus, the command:

```
"C:\Program Files\Schlumberger\Petrel 2010\Petrel.exe" /licensePackage Package2 ^  
/runWorkflow "Calculate Isochore" /nparm scalar=2 /sparm "type=Isochore" /quiet /nosplashscreen /exit ^  
"C:\Documents and Settings\DPM Documents\Petrel Projects\Batch_Tests.pet"
```

is equivalent to the above single line, works the same and is more readable. Use whichever method you prefer.



4801 Woodway Drive, Suite 150W • Houston, TX 77056 • www.scminc.com • info@scminc.com

© 2011 SCM E&P Solutions, Inc.

Petrel is a mark of Schlumberger