Job Stream Sequences Tutorial

1 Introduction

This document describes how to couple multiple Job Streams together into Sequences and execute them. By working through this tutorial the user will learn how to create new Sequences, initialize variable values, perform a steering iteration, and use the output of one Job Stream as input to a second. For more detailed information on the specific features covered in this tutorial, refer to the SNAP User's Manual.

This tutorial consists of two exercises. The first exercise walks through the process of building a two-stream job sequence consisting of an iterative TRACE steady-state stream followed by a TRACE restart transient stream. The first stream involves a simple TRACE standpipe model which is run iteratively, with an incrementing pipe wall heat flux until a desired output condition (in this case outlet void fraction) is met. The second stream will then restart from the final iteration of the steady state stream. After completing the first exercise the user will be familiar with creating and editing sequences, defining an iteration script, and transferring the output of one job stream to the next in a sequence.

The second exercise builds a two-stream sequence where the second stream is an iterative Uncertainty Quantification stream that restarts a steady state job stream. The model for the second exercise is an Engineering Template model that contains a coupled TRACE/PARCS main-line steam break model. The UQ stream will be set to iterate through a range of boundary condition values, with a separate report generated for each case. After completing the second exercise the user will be familiar with sequences in an Engineering Template model, and iterating a parametric job stream.

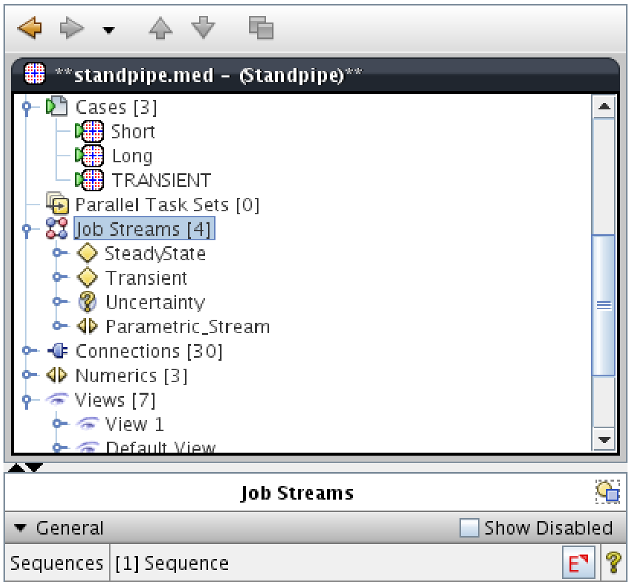
1.1 Background

Job Stream Sequences were added in SNAP Version 2.6.0 to support analyses that require looping, convergence, or optimization. A Job Stream Sequence consists of a set of Job Streams that will be executed in order. Each entry (Job Stream) in the Sequence will be executed in turn until all of the entries have been completed. If looping behavior is required for a Job Stream entry in a Sequence then it can be executed multiple times as separate iterations controlled by a user-defined Python Script. The script can be used to change variable values and determine (based on keywords) whether the loop is finished.

The Python scripts can also be used to set the file locations in External Files and File Sets by using Dynamic File Replacement. This allows Job Streams to be chained together using the output files of one stream as inputs for subsequent streams. Dynamic File Replacement allows the available files to be searched using a range of criteria including file type, keyword values, and iteration number.

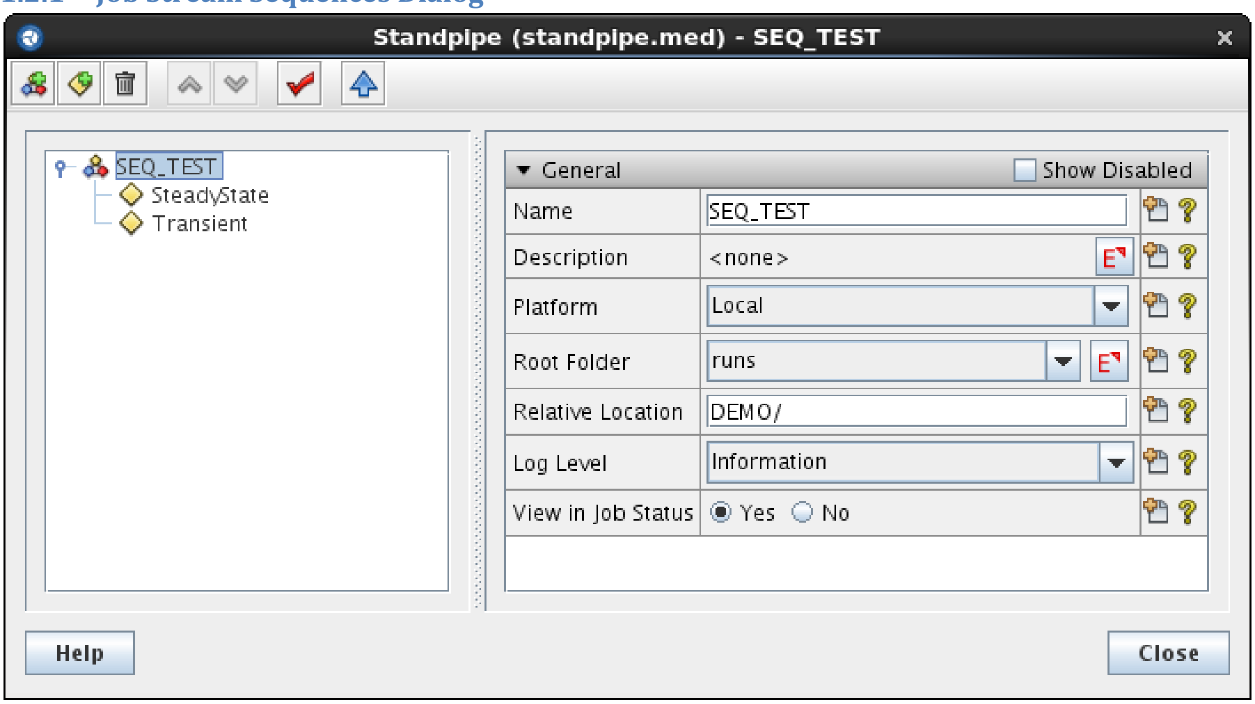
Sequence entries (i.e. Job Streams) are executed in the order in which they are defined in the Sequence. If an entry will be executed only once (i.e. no looping) then it will be placed in the Sequence folder directly. If an entry will be executed multiple times then the set of executions will be placed in a sub-folder. The Python scripts for setup (Entry_setup.py) and iteration (Entry_iterate.py), and their output (Entry.py_screen) will also be placed in this sub-folder.

1.2 User Interface



Selecting the Job Streams node in the Navigator will show the Sequences property editor in the main Property View. This editor shows the number of sequences currently defined in the model.

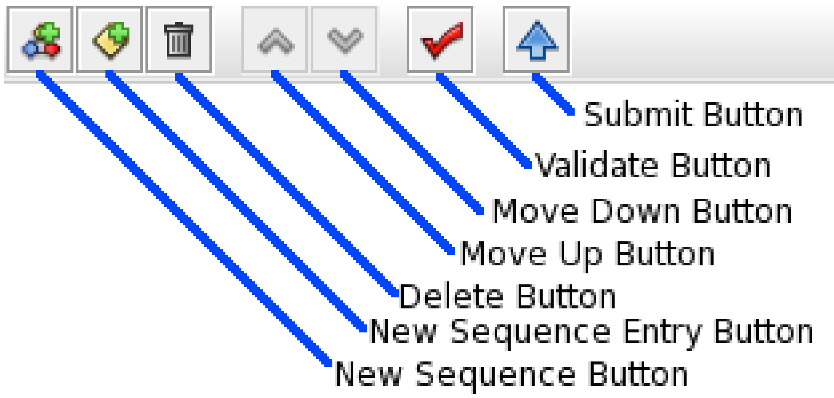
1.2.1 Job Stream Sequences Dialog



Pressing the Edit button in the Sequences property editor will open the sequences dialog. The sequences dialog is the primary user interface for creating, editing and submitting, Job Stream Sequences. This dialog has a tree of Sequences on the left with each Sequence's Entries as child nodes.

The right side of the dialog displays the properties of the selected node (Sequence or Sequence Entry) on the left. The main toolbar is at the top of the dialog and contains various buttons for creating and editing Sequences.

1.2.2 Job Sequence Main Toolbar



The main toolbar has buttons to create and reorder Sequences and Entries, validate a Sequence, and submit a Sequence.

- **New Sequence**: This button adds a new Job Stream Sequence to the current model.
- **New Sequence Entry**: This button adds a new Entry (Job Stream) to the currently selected Sequence. This button will only be enabled when a Sequence is selected in the tree and the model contains a Job Stream that is not already included in the selected sequence.
- **Delete**: This button removes whatever is selected in the tree. If a Sequence is removed all of its Entries will also be removed.
- **Move Up**: This moves the selected element in the tree one node higher.
- **Move Down**: This moves the selected element in the tree one node lower.
- **Validate**: This button validates the selected Sequence. If the Sequence contains any errors, an error report dialog will be displayed. Sequences with errors cannot be submitted.
- **Submit**: This button submits the current Sequence to the selected platform.

2 Exercise 1 - Sequence Basics

The purpose of this tutorial is to introduce the basic functionality of Job Stream Sequences. After completing this tutorial the user will be able to create a new Sequence, define an iteration script, and select the output of one stream in a Sequence as input for another stream.

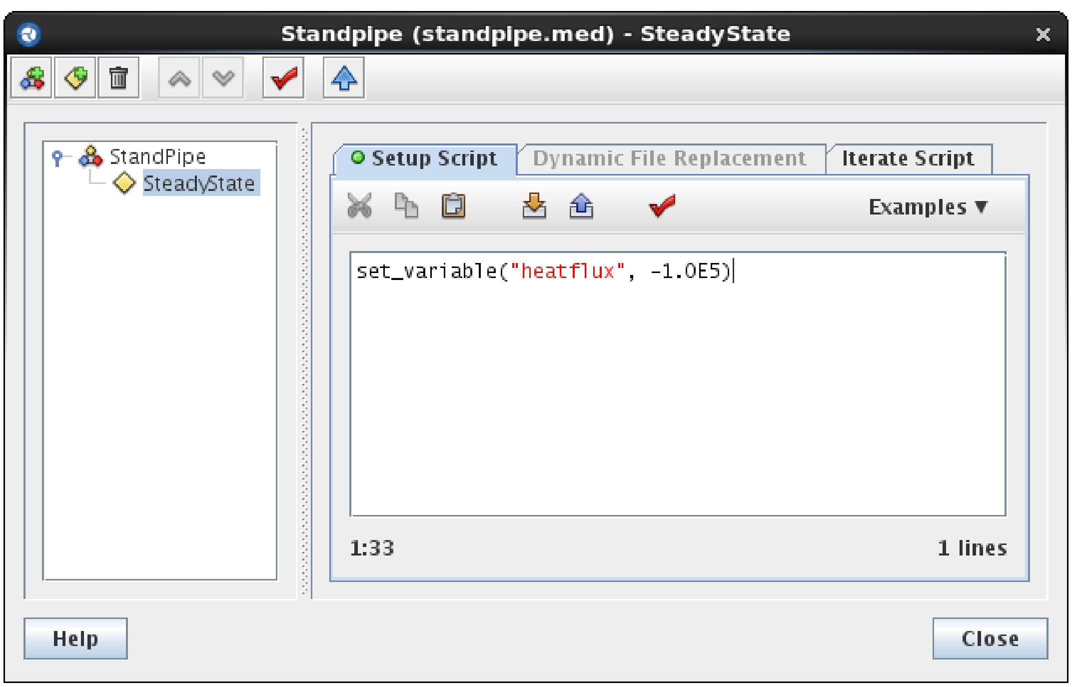
1. Open the included StandPipe/standpipe.med model.
2. Select the **Job Streams** node in the Navigator.
3. Open the Sequences dialog by pressing the **Edit** button on the Sequences property.

The next step is to create a new Sequence and define its submission properties. The submission properties will apply to all of the Job Streams in the Sequence.

4. Press the **New Sequence** ( ) button.
5. Set the **Name** field to "StandPipe".
6. Set the **Relative Location** to "DEMO/".

Note that the initial platform is set to Local. Sequences are currently only supported on the Local Calculation Server. In the next few steps we will add a Job Stream to the Sequence and define the initial value for a numeric variable.

7. Press the **New Sequence Entry** ( ) button.
8. Select the SteadyState Job Stream and press the **OK** button.
9. Ensure that the Setup Script panel is shown.
10. Click on the **Examples** button and select **Set A Variable Value** from the menu.
11. Change the referenced variable from “V1” to "heatflux".
12. Change the initial value from 1.0 to -1.0E5.

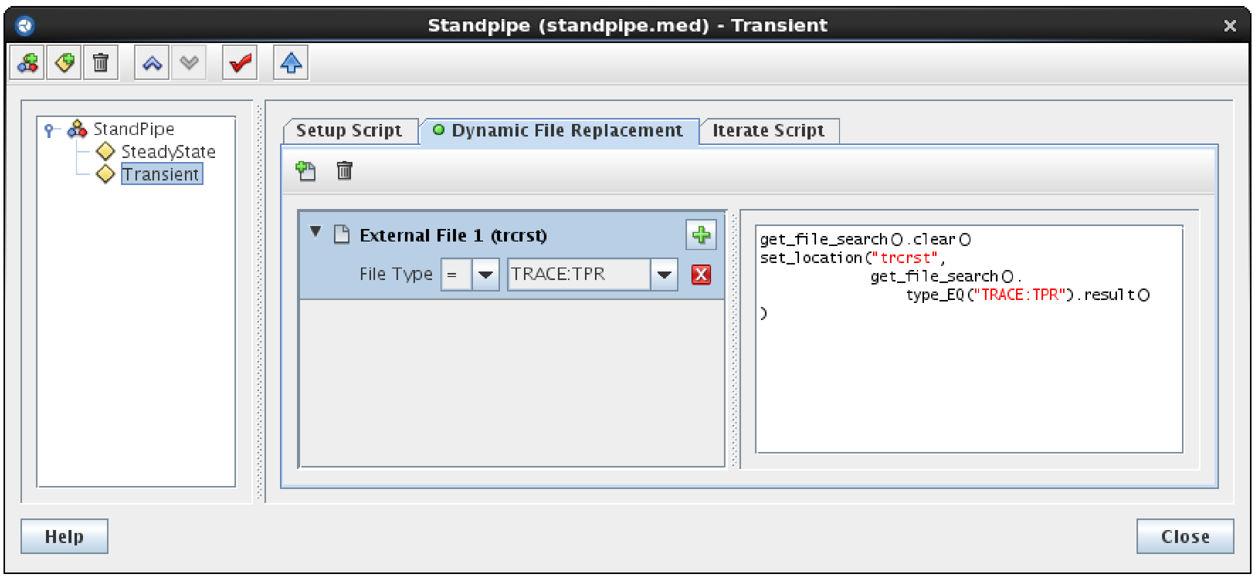


This will set the value of the heatflux variable to -1.0E5 before the stream is executed. Now we will add an additional Sequence Entry that will use the output generated from the SteadyState stream as input.

13. Press the **New Sequence Entry** ( ) button.
14. Select the Transient Job Stream and press the **OK** button.

Notice that the Dynamic File Replacement tab is now enabled. This means that the selected Sequence Entry contains either an External File or File Set and that the Entry is not the first Entry in the Sequence.

15. Select the **Dynamic File Replacement** tab.
16. Press the **New File Replacement** ( ) button.
17. Select **External File 1** (trcrst) in the selection dialog and press **OK**.



Notice the Python script that appears in the right panel of the dialog. This script will be executed after the setup python script and before executing the Job Stream. The first line clears any previous file searches. The next line indicates that the file location for File Set 'trcrst' is being defined. The following lines define the criteria for the file search. The last line shows that the file type for the file must be 'TRACE:TPR'. Currently, in the SteadyState Job Stream, there is only one output file with the file type 'TRACE:TPR, so this provides enough information to return the TPR output file from the steady state case.

18. Press the **Check Sequence** ( ) button.

This will open an Error Report dialog for the currently selected Sequence. This report contains the results of validating the Sequence as well as each of the Job Streams referenced by Sequence Entries. In this case, there is a warning stating that the "Transient" job stream contains an error. This is due to the fact that External File 1 does not have a location defined. The Dynamic File Replacement added above will correct this problem before the stream is executed.

19. Press the **Close** button to close the Error Report.

20. Press the **Submit Sequence** () button.
21. Press **OK** to submit the sequence without correcting the warning.
22. Wait for the submitted Sequence to complete.

Job Status will open to show the Sequence Manager's log output while the Sequence is running. The location of the Python output files are listed directly in the log to make them easier to find.

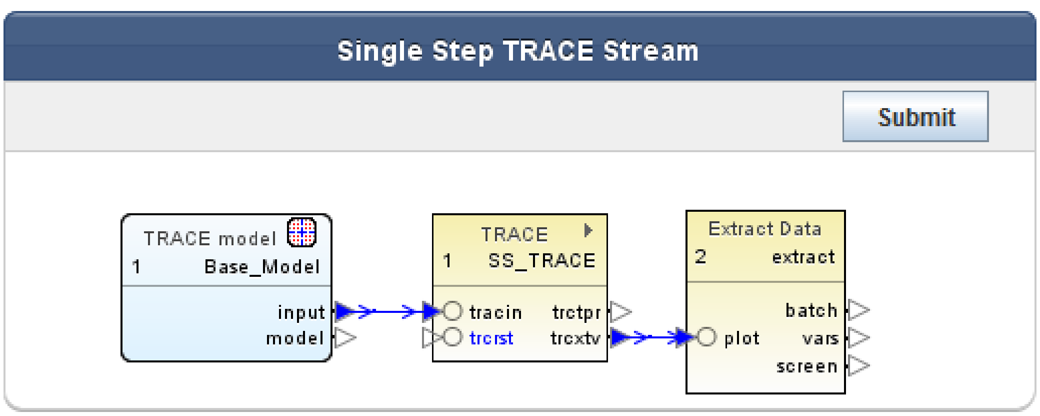
```
[<Date>] - Writing Python stderr/stdout for Transient iteration 1 to: <Location>
```

Notice that the streams are executed in the DEMO/StandPipe/ folder.

23. Close Job Status.

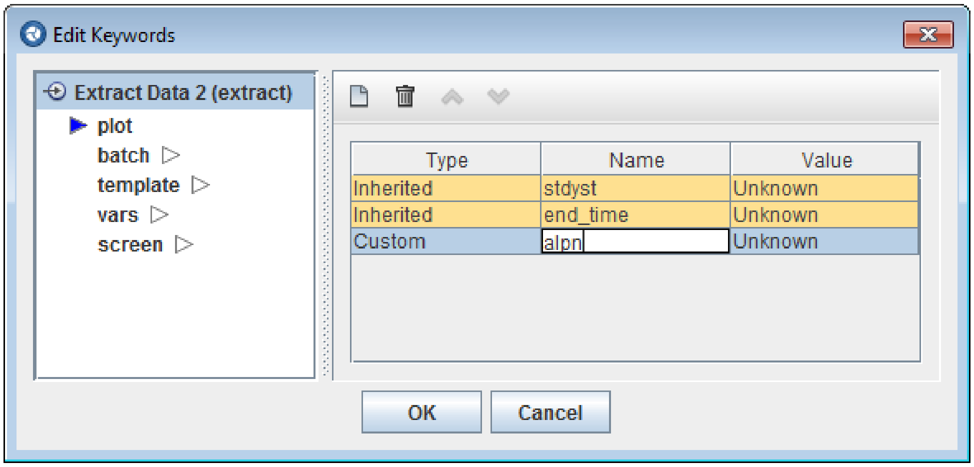
Next we will modify the Sequence to iterate on the steady state case, changing the value of the heat flux variable until the void fraction at the end of the pipe is greater than 0.5. This will be accomplished using an Extract Data step to retrieve the value from the plot file, and a post processor Python method to write the keyword value. The first step is to add an Extract Data step to the stream that will write the void fraction of the last hydraulic cell to a new keyword.

24. Press the **Close** button to close the Sequences dialog.
25. Select the **Steady State Stream** view in the view tabs.
26. Use the **Insert Tool** to add a new Extract Data Job Step to the right of the SS_TRACE Step.



27. Set the new Job Step's **Stream** to "SteadyState".
28. Set the new Job Step's **Name** to "extract".
29. Set the **Plot File Type** to TRACE.
30. Press the **Edit** button for the **Input Files** property.
31. Click in the table cell at row 1 ("plot") column 4 ("source").
32. Press the **Edit** button to open the **Select Input Source** dialog.
33. Select "TRACE 1 (SS_TRACE)" and press **OK**.
34. Press the **OK** button to close the **Define Input Files for Extract Data** dialog.

Now we ***mu****st define the keyword* ***that will hold the plot steering criteria. To s****et the value of a keyword on a step, or file, the keyword must already exist. The next steps create a new keyword for the Extract Data step that will be used to steer the iteration.*



35. Press the **Edit** button for the **Keywords** property.
36. Select "Extract Data 2 (extract)" in the list on the left hand side of the dialog.
37. Press the **New Keyword** ( ) button.
38. Set the **Name** of the new Keyword to "alpn".
39. Press the **OK** to close the **Edit Keywords** dialog.

Now the AptPlot script will be updated to extract the desired value to the var *output file. Any number of variables may be written to this file, each included with the "SAVEVAR" method. For this example a single SCALAR variable is saved, which will then be read-in by a post-processor python method.*

40. Press the **Edit** button for the **AptPlot Script** property.
41. Enter the following script and press **OK**.

```
CALC "ALPN = yAtMaxX(T0_alpn-21A20)"
SAVEVAR "ALPN"
```

42. Press the **Edit** button for the **Custom Processing** property.
43. Select the **Post-Execution Python** tab at the top of the dialog
44. Enter the following script and press **OK**.

```
f = open( "variables.dat", 'r')
for line in f:
  if line.startswith("SCALAR"):
    values = line.split("=")
    set_keyword('alpn', values[1].strip() )
```

The extract step now defines a keyword with the void fraction from node 20 of pipe 21. The Sequence iteration can use this keyword value to determine the result of each SteadyState iteration. The next steps will add an iteration script to the SteadyState Sequence Entry. The Iteration Script will increase the constant heat flux of the pipewall heat structure until the void fraction of the pipe hits 0.5.

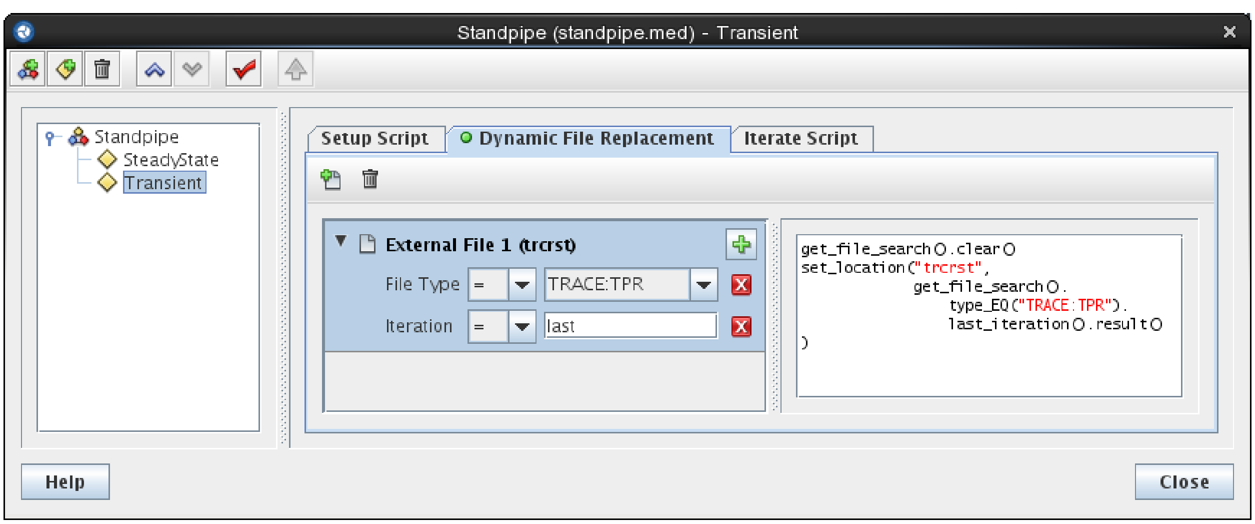
45. Select the Job Streams node in the Navigator.
46. Press the **Edit** button for the **Sequences** property.
47. Select the **SteadyState** Sequence Entry.
48. Select the **Iterate Script** tab.
49. Enter the following script into the panel.

```
print "Calculating Input For Iteration " + str(get_iteration_number())

# Get the keyword set on the extract step
alpn = get_keyword("extract/alpn")
print "ALPN Keyword = " + str(alpn)

# The end value is 0.5 void fraction
if alpn > 0.5:
    print "End Reached"
    set_finished()
else:
    # Keep increasing the heatflux by -1.0E5 until the desired
    # value is reached.
    heatflux = get_variable("heatflux") - 1e5
    print "Setting Heat Flux to: " + str(heatflux)
    set_variable( "heatflux", heatflux )
```

The SteadyState Sequence Entry will now execute multiple times, creating a new TPR file for each iteration. The Dynamic File Replacements in the Transient Entry will now need to be udpated to select just one of the TPR files.

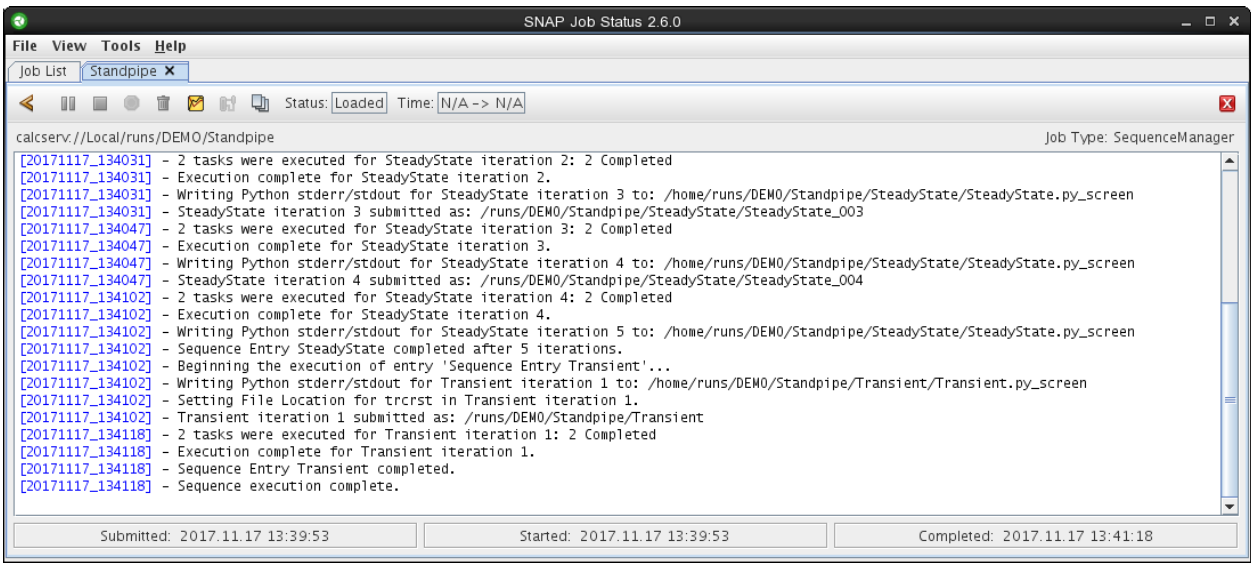


50. Select the **Transient** Sequence Entry on the left side of the dialog.

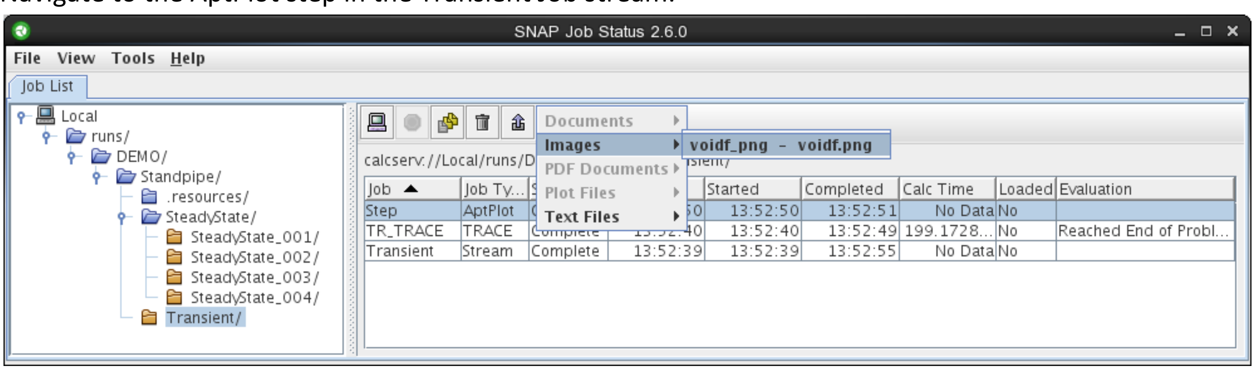
51. Select the **Dynamic File Replacement** tab.
52. Press the **New Search Criteria** ( ) button to for **External File 1 (trcrst)**.
53. Select **Iteration** and press **OK**.

Notice that the value of the iteration criteria is set to "last". By default the iteration criteria is used to select the most recent iteration. This ensures that the TPR file for the transient case comes from the last executed steady state iteration.

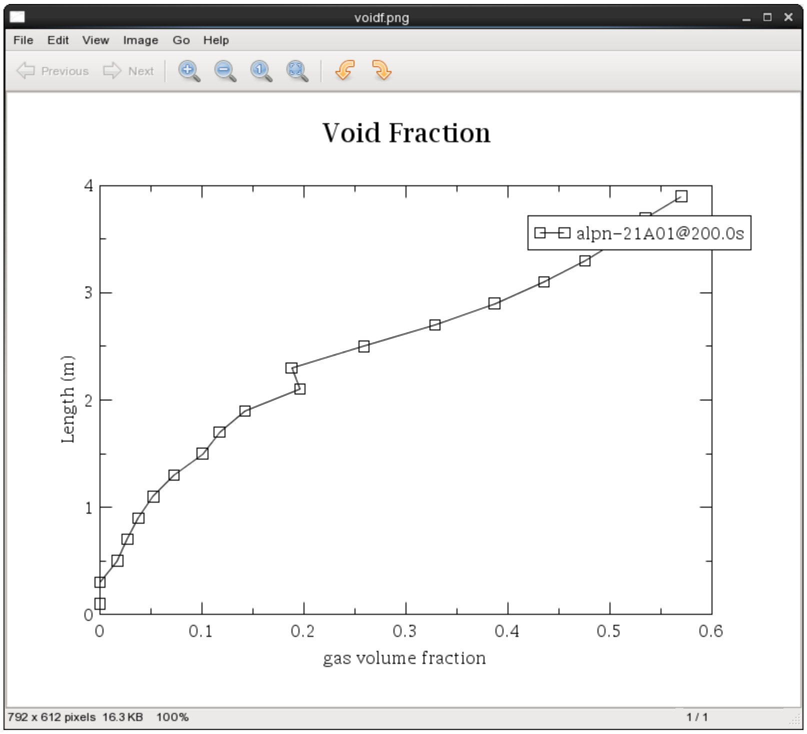
54. Select the **StandPipe** node in the **Sequence Tree** on the left.
55. Press the **Submit Sequence** ( ) button and wait for the sequence to complete.



56. In Job Status, close the **Standpipe** tab.
57. Navigate to the AptPlot step in the Transient Job stream.



58. Open the voidf.png output file.



This completes the Sequence Basics exercise.

3 Exercise 2 – Advanced Sequence Concepts

The purpose of this exercise is to demonstrate the advanced concepts of Job Stream Sequences. The exercise includes the addition of a Sequence to an Engineering Template model that includes two coupled TRACE/PARCS Job Streams. The second Job Stream will be converted into a DAKOTA UQ stream, varying a TRACE sensitivity coefficient. Finally, the Sequence will be updated to iterate the UQ stream by incrementing an underlying model variable

The tutorial begins by adding a new Sequence to the model.

1. Open the included MSLB/MSLB_TEMPLATE.med model.
2. Select the **Job Streams** node in the Navigator.
3. Press the **Edit** button for the **Sequences** property.
4. Press the **New Sequence** ( ) button.
5. Set the **Name** field to "Coupled".
6. Set the **Relative Location** to "DEMO/".
7. Press the **New Sequence Entry** ( ) button and select the “CP_SS” stream.
8. Press the **New Sequence Entry** ( ) button and select the “CP_TR” stream.
9. Select the **Dynamic File Replacement** tab.
10. Press the **New File Replacement** ( ) button and select “parcs_rsti”.
11. Press the **New File Replacement** ( ) button and select “trcrst”.

Note that no additional file criteria are needed for either file replacement because the CP_SS Job Stream produces only one file of each file type.

12. Press the **Submit Sequence** (⇧) button and wait for the sequence to complete.
13. Close Job Status.
14. Press the **Close** button to close the Sequences Dialog.

This step in the processes is a test submission to ensure that all submission properties are executed correctly. The next steps convert the Job Stream into an DAKOTA UQ stream where the TRACE gap-conductance sensitivity coefficient will be varied. The outside temperature of a representative heat structure will be used as the figure of merit.

15. Select the **CP_TR** Job Stream in the Navigator.
16. Press the **Edit** button for the **Stream Type** property.
17. Select **DAKOTA Uncertainty** and press the **OK** button.
18. Press the **Edit** button for the **Parametric Properties** property.
19. Set the **Number of Samples** to 10.

*In a real case, the Number of Samples would be much higher. The **Calculate Button** can be used to determine the Number of Samples that would be required to meet the Probability and Confidence levels entered in this panel for the number of Figures of Merit defined. For this tutorial, 10 cases are used for brevity.*

20. Add a **Figure of Merit** by pressing the **New** ( ) button.
21. Set the **Name** of the new Figure of Merit to "FuelTemp".

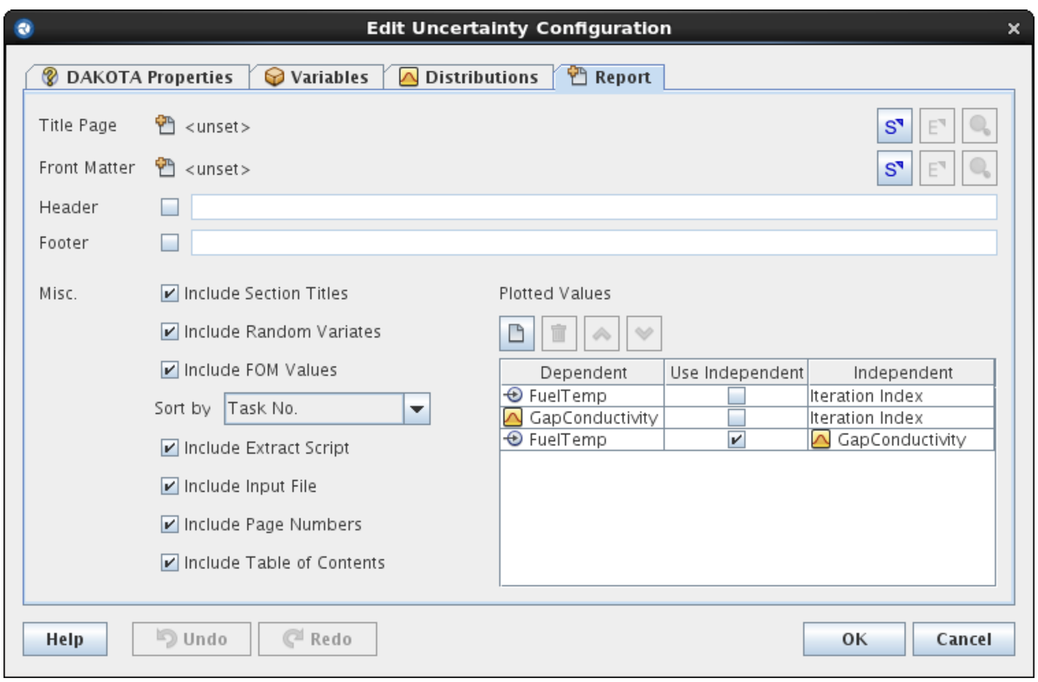
This value (FuelTemp) is the output value from the TRACE calculations. More specifically, this is the surface temperature of a representative heat structure at the end of the calculation.

The next steps will setup the UQ input variable. For this exercise, the gap conductivity sensitivity coefficient will be modified by a factor. The factor's value will be defined using a Normal distribution about 1.0 with a standard deviation of 0.05. The distribution names are used as the parametric keywords for this Job Stream.

22. Select the **Variables** tab.
23. Pressing the **New** ( ) button to add a **Variable**.
24. In the **Select a model:** drop-down list, select the "*TRACE.med – (mslb_trace_ss)"* item.
25. Select **Sensitivity Coefficients** in the left list.
26. Select "gapCondSV (Nounit)" from the list on the right.
27. Press the **Next** button.
28. Select the **Factor** radio button and press the **Finish** button.
29. Select the **Distributions** tab.
30. Set the **Name** to GapConductivity.
31. Set the **σ (STDV)** to 0.05.

The next steps define several graphs that will appear in the final DAKOTA UQ Report. Model Notes can can be selected in the Report tab to be used as the Title Page and Front Matter for the generated report. An optional header and footer text may also be included. Three plots will be added to the report: one for the figure of merit by iteration, one for the distribution by iteration, and one that shows the figure of merit by distribution.

32. Select the **Report** tab.
33. Press the **New** ( ) button to add a Plot.
34. Select **FuelTemp** and press the **OK** button.
35. Press the **New** ( ) button to add a Plot.
36. Select **GapConductivity** and press the **OK** button.
37. Press the **New** ( ) button to add a Plot.
38. Select **FuelTemp** and press the **OK** button.
39. For the second **FuelTemp** row, select the "Use Independent" column checkbox.
40. Select the **Independent** column for the second FuelTemp row and press the **Select** button
41. Select **GapConductity** and press the **OK** button.



42. Press the **OK** button and press **Yes** in the confirmation dialog.

The DAKOTA Uncertainty stream type is now set up. The next steps will add a Data extraction step after the TRACE step, and a DAKOTA report step after that.

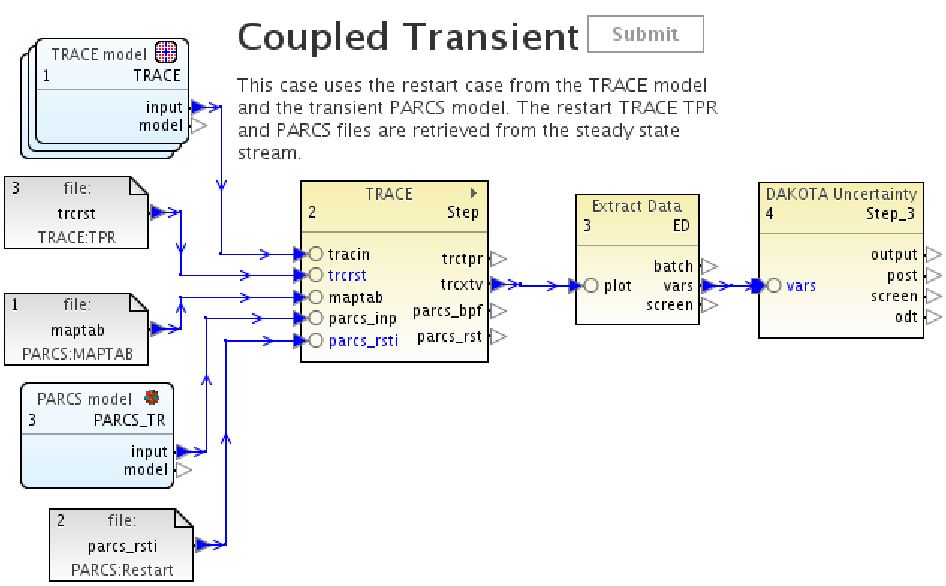
43. Select the **Transient Stream** view.
44. Select the **Insert Tool.**
45. Insert a new Job Stream Step to the right of the TRACE Step.

46. Select **Extract Data** and press the **OK** button.
47. Set the **Name** to “ED”
48. Set the **Plot File** Type to “TRACE”
49. Select the **Connect Tool.**
50. Connect the **trcxtv** output of the TRACE step to the **plot** input of the Extract Data step.
51. Press the **Edit** button for the **AptPlot Script** property.
52. Replace the text “< Enter AptPlot commands here >“ with the following line

```
CALC "FuelTemp = yAtMaxX(T0_tsurf0-840A03)"
```

This tells the extract data step to retrieve the y value at the end of the calculation for the provided expression. The expression “T0_tsurfo-840A03” tells AptPlot to extract from the first TRACE Plot file (T0) the channel “tsurfi-840A03” which is the outer surface temperature for heat structure 840, axial level 3.

53. Press the **OK** button.
54. Select the **Insert Tool.**
55. Insert a new Job Stream Step to the right of the TRACE Step.
56. Select **DAKOTA Uncertainty** and press the **OK** button.
57. Select the **Connect Tool.**
58. Connect the **vars** output of the Extract Data step to the **vars** input of the Dakota Uncertainty step.
59. Select the TRACE model node.
60. Set the **Parametric** property to “True”.



61. Re-open the **Job Stream Sequences** dialog.
62. Submit the Coupled Sequence.
63. Wait for the stream to complete.
64. Close Job Status.

This is testing to ensure that the Uncertainty Job Stream has been defined correctly. The DAKTOA report should demonstrate a linear relationship to the fuel center line temperature and the gap conductivity sensitivity coefficient. The next steps will modify the Sequence to iterate the uncertainty Job Stream by incrementing a variable in the underlying PARCS model.

65. Select the CP_TR node in the **Job Stream Sequence** dialog.
66. Enter the following line into the **Setup Script** text pane

```
set_variable(“m2.SIGNAL_DELAY”, 0.0)
```

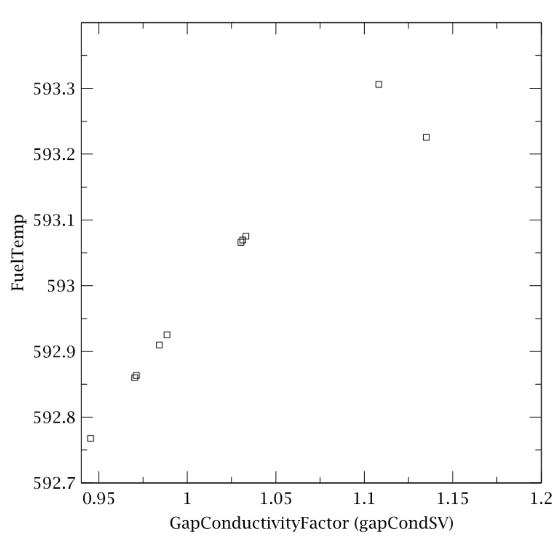
67. Select the **Iterate Script** tab.
68. Enter the following line into the **Iterate Script** text pane.

```
set_variable("m2.SIGNAL_DELAY", get_variable("m2.SIGNAL_DELAY") + 0.5 )
```

This encoded variable name includes a prefix that determines in which model the variable will be changed. To modify a variable in the TRACE model, a prefix of ‘tr’ would be used. This prefix is the ***Short Name*** *property of the Reference Model. The default name is ‘m’ followed by the creation index.*

69. Set the **Maximum Iterations** field to 5.
70. Submit the Coupled sequence.
71. Wait for the streams to complete.

The runs/DEMO/Coupled/CP_TR directory will contain all five iterations of the CP_TR stream. Each appears in a sub-directory CP_TR with the iteration number appended. The DAKOTA UQ report can be found in the “report.odt” file inside the UQ directory.



This completes the advanced sequence tutorial.