

TRACE/SNAP User Workshop  
March 26<sup>th</sup> - 29<sup>th</sup>, 2018

System Source Computer Training Center  
Columbia, Maryland

Introduction to SNAP

Applied Programming Technology, Inc.



# What is SNAP?

---

- Symbolic Nuclear Analysis Package
  - Suite of Applications
- Standard Graphical User Interface designed to simplify the use of analytical codes.
  - Construct, Maintain and Document Models
  - Run Cases
  - Analyze Results
- Platform Independent, Pure-Java, Plug-in Arch.
  - Can Be Adapted to Any Engineering Code
  - Highly Extensible and Flexible

# SNAP is a Suite of Applications

---

- Primary SNAP Apps:
  - Model Editor
  - Configuration Tool
  - Job Status
  - Calculation Server
- Associated Software
  - AptPlot, PyPost & Demuxers
    - Plotting & Post-Processing Tools
  - ACAP - Automated Code Assessment Program
  - TSA - Test Suite Analyzer
  - DAKOTA - Uncertainty Quantification
  - MATLAB / Mathcad
  - MS Office eXcel & Word
  - OpenOffice / LibreOffice

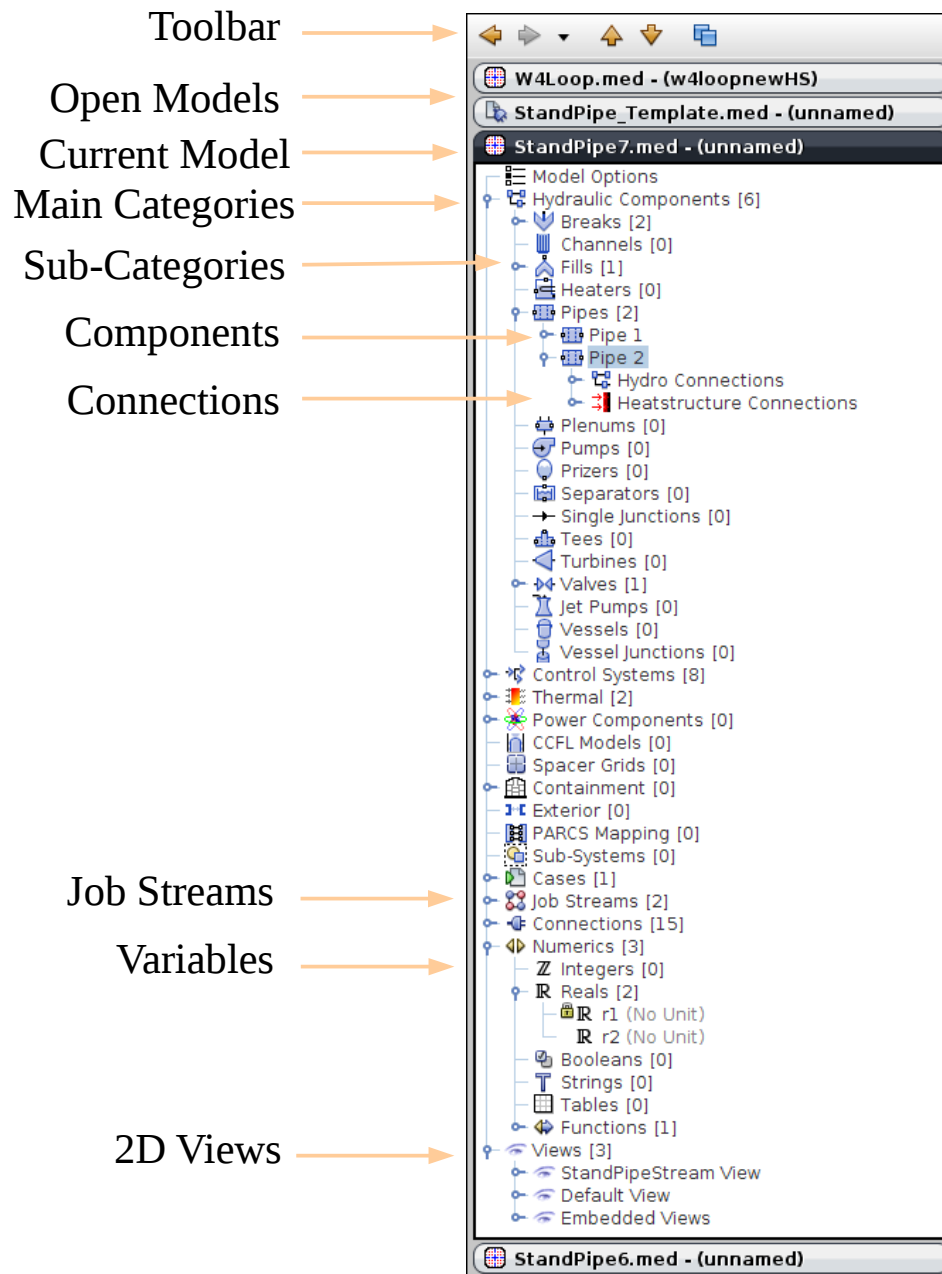
# Why Use SNAP?

---

- Common Functionality Across Different Codes
  - Minimize Learning Curve for new Codes
- Logical Organization of Model Components and Interconnections
- Advanced Model Editing Tools
  - Elevation Loop checking
  - Renodalization
- Input Checking and Model Validation Tests
- Manage Model Documentation
- Submit and Monitor Calculations
  - Single execution or sequence of calculations
- Automate Post-processing of Results
  - 2D/3D Animation
  - Plot generation
  - Report Generation



# Component Navigator



- Multiple Open Models
  - Accordion Layout
- Organizes all Model Components
- Categories and Root Components
- Interconnections
- Copy/Paste
- Add/Remove
- Drag onto 2D View

# Property Views

- Main Property View
  - Data Follows Selection in 2D Views or the Navigator
  - Multi-Select Editing
  - Tab Traversal
- Component Property View
  - Separate Window
  - Single Component Only
- Custom Editors
- Attribute Documentation
- Pop-up Help

**Pipe 1**

☐ Show Disabled

Component Name	unnamed	
Component Number	1	
Description	<none>	
Component Geometry	Cells: 20	
Initial Conditions	[ Valid Conditions ]	
Friction	Kfac ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	
Fluid Power Options	Not Modeled	
Wall Roughness	0.0 (ft)	
Inlet	Fill 11 Cell 1 outlet	
Outlet	Pipe 2 Cell 1 inlet	
Cross Flow Connections	[1] Connections	
Pipe Type	[0] No Accumulator	
Number of Pipes	1	
Leak Paths	[0] Leak Paths	
<b>Pipe Wall</b>		
Use Pipewall	<input type="radio"/> True <input checked="" type="radio"/> False	
▶ Wall Power		
▶ Trace Species		

# Property Editors and Selectors

- Property Editors – **E**

- Detailed custom dialogs
- Display/edit sets of values

**Pipe 11 (\$11\$ int-loop st-gen primary)**

▼ General ☐ Show Disabled

Component Name: \$11\$ int-loop st-gen primar

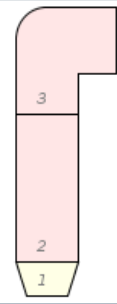
Component Number: 11

Description: <none>

Component Geometry: Cells: 3

Initial Conditions: [ Valid Conditions ]

Geometry - Pipe 11 (\$11\$ int-loop st-gen primary)



Cell Number	Volume (m³)	Length (m)	Vol. Avg. Flow Area	DZ (m)	2D Drawing Pivot
1	16.77	1.0	16.77	1.0	<input type="checkbox"/>
2	14.153	4.35	3.2535632	4.35	<input type="checkbox"/>
3	14.153	4.35	3.2535632	2.175	<input type="checkbox"/>
<b>Total</b>	<b>45.076</b>	<b>9.7</b>	<b>23.277126</b>	<b>7.525</b>	

**Calculate**

☐ Volume ☐ Length ☒ Area

**Cells** **Edges** **Orientation**

Close

- Component Selectors – **S**

- Select from the available components
- Sort by category, number, or component

**Fill 25 (\$25\$ int-loop hpis & lpiis)**

Timestep Holdover: 0.0 (-)

Max Acceleration: 1.0E5 (m/s²)

Fill Table: Rows: 17 [0.0,32.8],[2.0E5,33....]

Rate Factor Table: Rows: 0 []

Table Signal: Pressure 2

► Contan Coupling

► Scale Factors

► State Controllers

Select from Control Blocks and Signal Variables

**Available Components**

Category	Number	Component
Signal Variables	1	Problem Time 1
Signal Variables	2	Pressure 2
Signal Variables	3	Pressure 3
Signal Variables	4	Pressure 4
Signal Variables	17	Collapsed Water Level 17
Signal Variables	18	Collapsed Water Level 18
Signal Variables	19	Collapsed Water Level 19
Signal Variables	21	Collapsed Water Level 21

None

OK Cancel



# Component Differencing

- Compare the ASCII input of two component selections
- Single Component to Entire Models
- Left Side Implements Component Listener Interface
  - Monitor Changes to a component
- Can Export Results to a File.

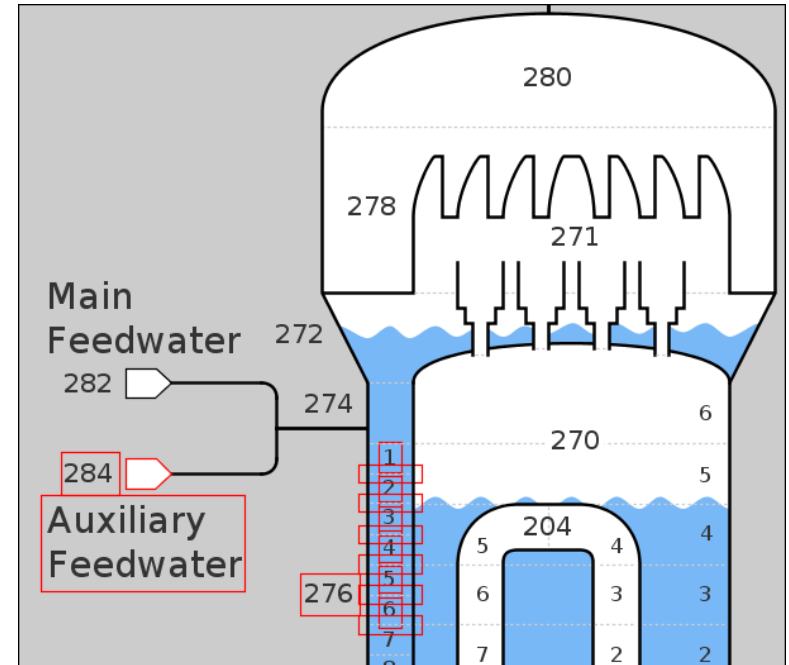
Tee 17 (\$17\$ bkn-loop sec-side downcomer)					Tee 27 (\$27\$ brk-loop sec boiler/stdome)				
* alp1	*	1.0	0.0	0.0	* pa1	*	4.85E6	4.85E6	4.85E6
* v11	*	0.0	0.0	0.0	* dx2	*	2.175e	0.0	0.0e
* vv1	*	0.0	0.0	0.0	* vo12	*	6.525e		
* vv1	*	0.0e			* fa2	*	5.0	1.5e	
* t11	*	535.14	535.14	535.14	* kfac2	*	1.0E-10	3.0E-3e	
* tv1	*	535.14	535.14	535.14	* grav2	*	-0.55758	-1.0e	
* p1	*	4.85E6	4.85E6	4.85E6					
* pa1	*	0.0	0.0	0.0					
* dx2	*	1.0e							
* vo12	*	0.5e							
* fa2	*	0.5	0.5e						
* kfac2	*	1.0E-10	0.0e						
* grav2	*	0.0	0.0e						
* hd2	*	0.1	0.1e		* hd2	*	0.1	0.1e	
* nff2	*	1	1e		* nff2	*	1	1e	
* alp2	*	0.0e			* alp2	*	1.0e		
* v12	*	0.0	0.0e		* v12	*	0.0	0.0e	
* vv2	*	0.0	0.0e		* vv2	*	0.0	0.0e	
* t12	*	440.0e			* t12	*	535.14e		
* tv2	*	440.0e			* tv2	*	535.14e		
* p2	*	4.85E6e			* p2	*	4.85E6e		
* pa2	*	0.0e			* pa2	*	0.0e		

Left [46] \* alp2 \* 0.0e  
Right [39] \* alp2 \* 1.0e



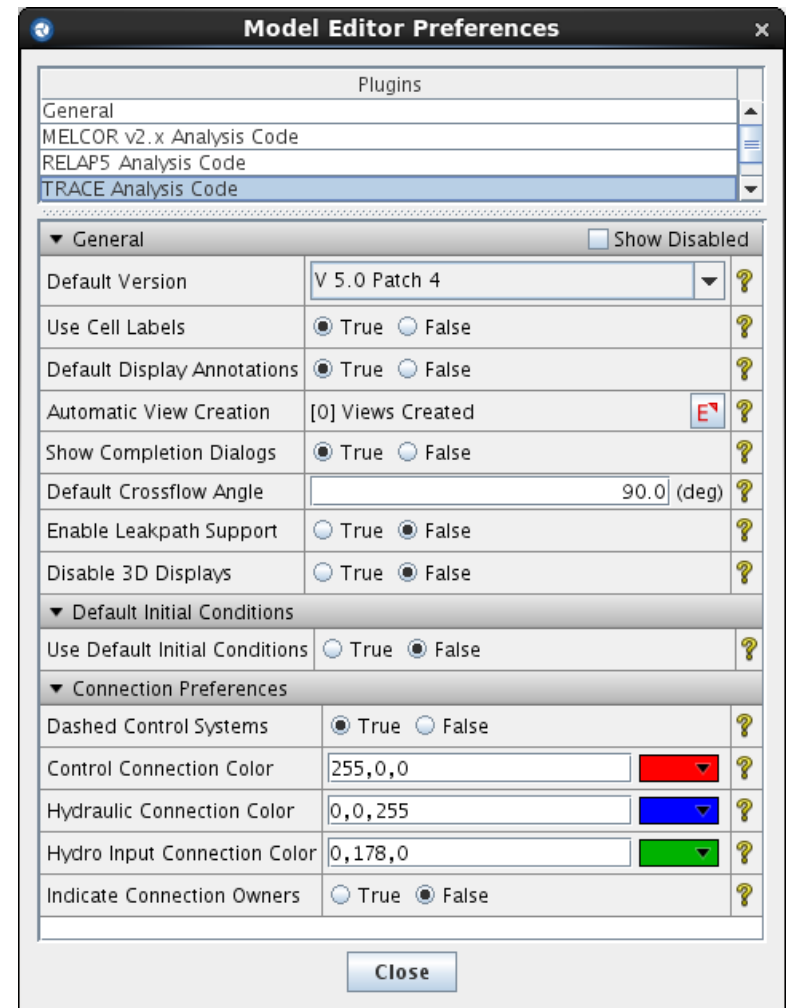
# 2D View Templates

- Exported From 2D Views
- Imported Into Any Similar Model
- Contains Annotations, Display Beans, Component Locations, etc.
- Complex Views Duplicated Between Models
- Recreate Views When Re-Importing a Modified Model



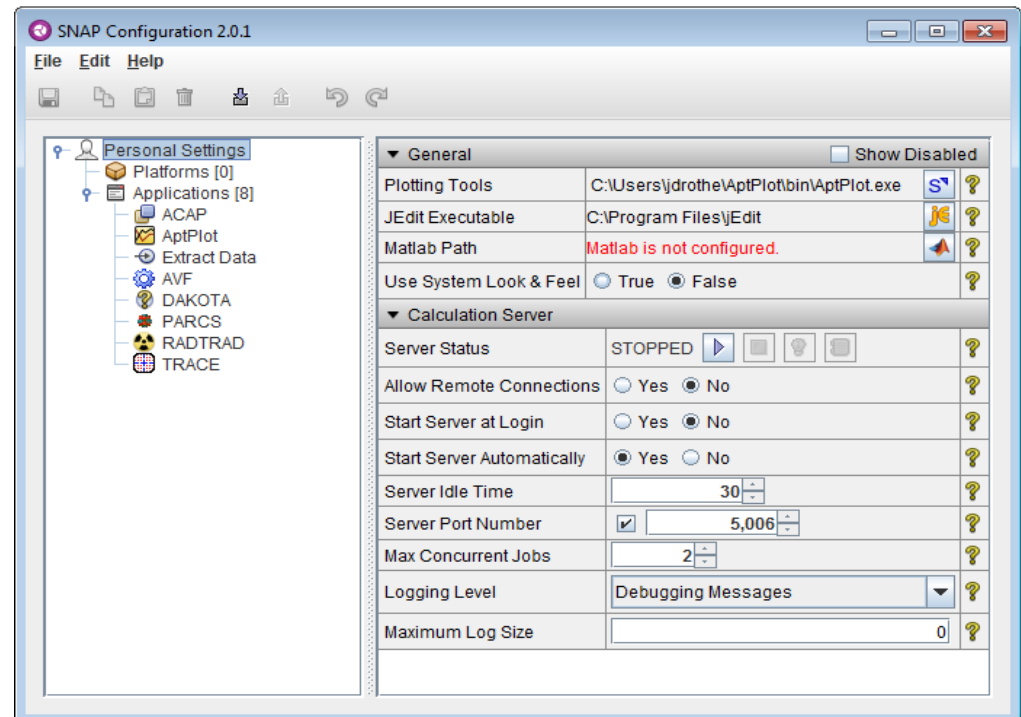
# Model Editor Preferences

- Located under Edit menu
- General Preferences
  - Default Units
  - Session
  - Selection Modes
- Plug-in Preferences
  - Completion dialogs
  - User Manual Locations
  - Drawing Options



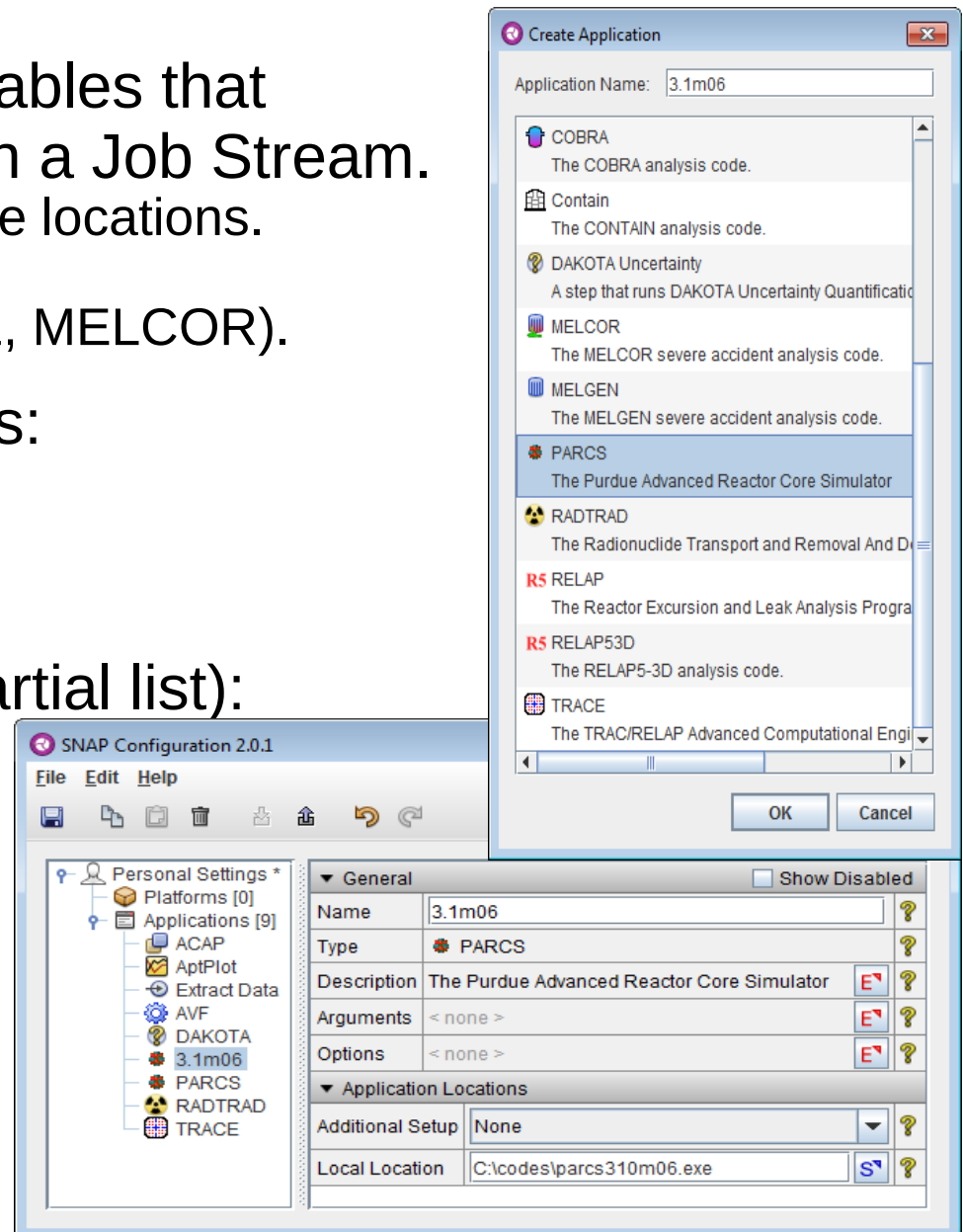
# Configuration Tool

- Personal Settings are stored in the User's home directory.
- Application paths for jEdit, AptPlot, and MATLAB.
- Local Calculation Server Settings
  - Status, Server Port, Logging Level, etc.
- Usually no need to start/stop the Calculation Server manually.
- Manages the available set of Platforms & Applications.
- Global Settings are setup by your System Admin.



# Configuration Tool – Applications

- Applications define executables that can be included as steps in a Job Stream.
  - Local and remote executable locations.
  - Command line arguments.
  - Application version (TRACE, MELCOR).
- Pre-configured applications:
  - ACAP
  - AptPlot
  - DAKOTA Uncertainty
- Supported applications (partial list):
  - CONTAIN
  - MELCOR
  - PARCS
  - RELAP5 / RELAP5-3D
  - TRACE
  - RADTRAD
  - FAST / FRAPTRAN
  - SCALE

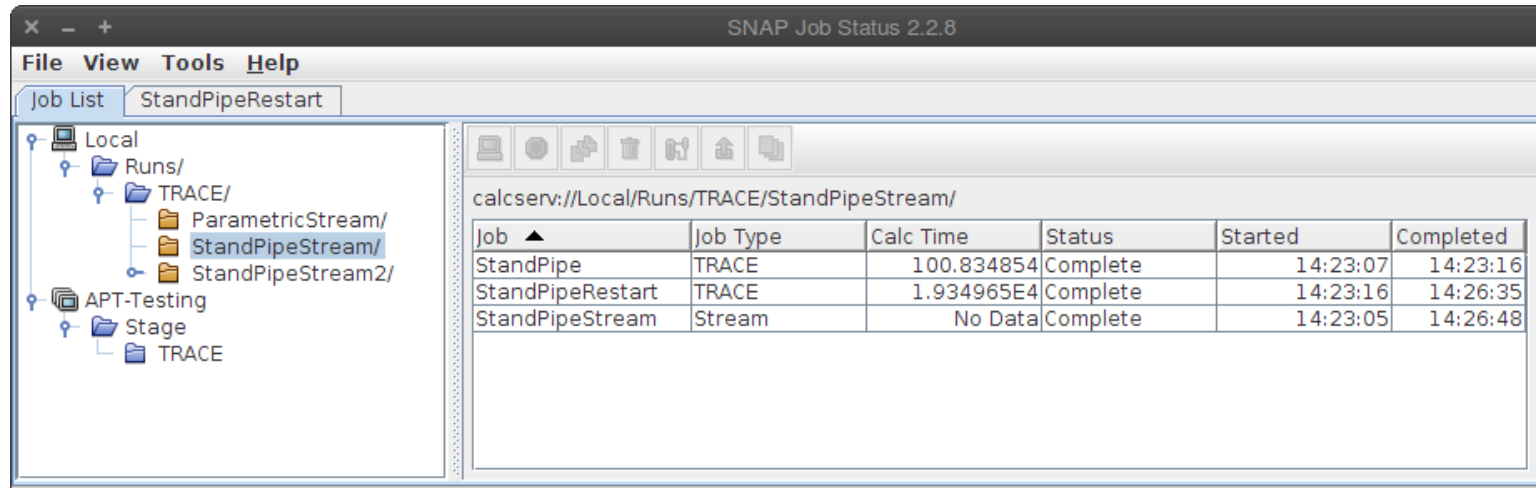


# Calculation Server

---

- Manages job stream execution.
- Required to Animate results. Interactively or in Replay mode.
- Provides access to files output by a calculation.
- Local server starts & stops automatically as needed.
- Use the Configuration Tool to Start/Stop remote servers.
- Completed Jobs can be imported using Job Status.

# Job Status



- Display the Status of Submitted Jobs
- Connect to One or More Platforms
  - Calculation Server or Tracking Server
- Delete, Plot, Animate and Terminate jobs
- Mount "Root Folders"
- View ASCII Input and Output Files
- Import Completed Jobs



# Additional SNAP Functionality

---

- SNAP Variables - Use variables & functions to define input values and control execution.
- Model Notes - Attach notes to components & individual attributes
- Attribute Ownership & Reviewer
- Model Notebook Generation – Generate a model documentation report in Microsoft Word (docx) or Open Office (odt).
- Component & Group Renodalization
- Initial Condition Import & Management
- Restart Cases
- Job Streams
- Uncertainty Quantification Analysis (Dakota)
- Master Integrator - Split complex models into sub-models.
- Engineering Templates - Provides support for multiple models.
- Kiosk - Create Input templates to constrain execution of predefined engineering templates.
- Integration with Revision Control System - Subversion

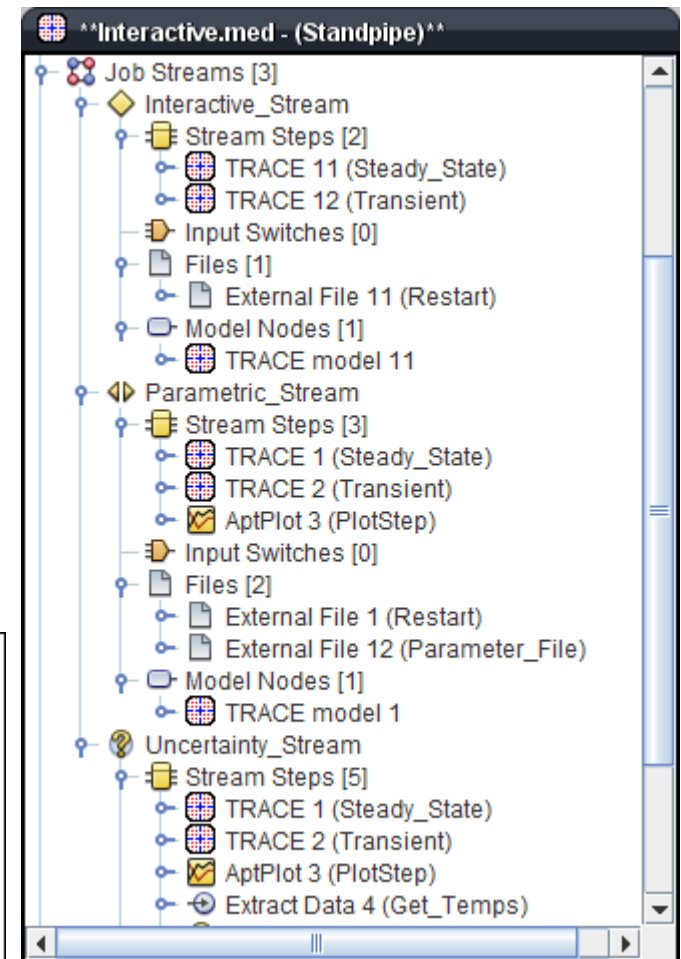
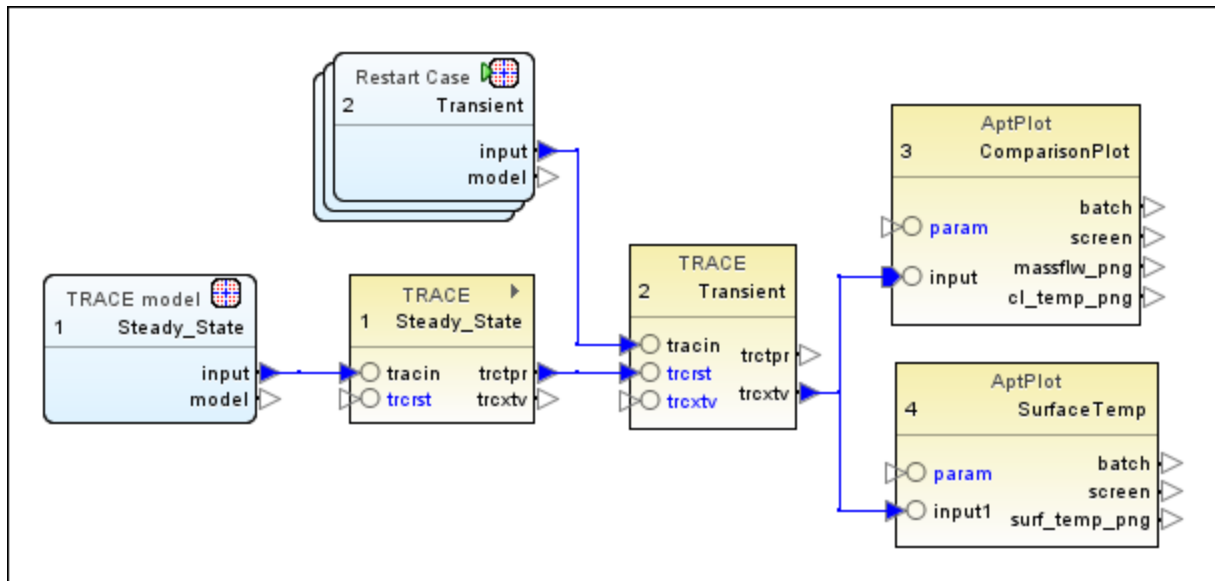
# Exercises

---

- Exercise 1. Configuring SNAP
- Exercise 2. Editing an Existing Model
- Exercise 3. Working with Model Views

# Job Streams

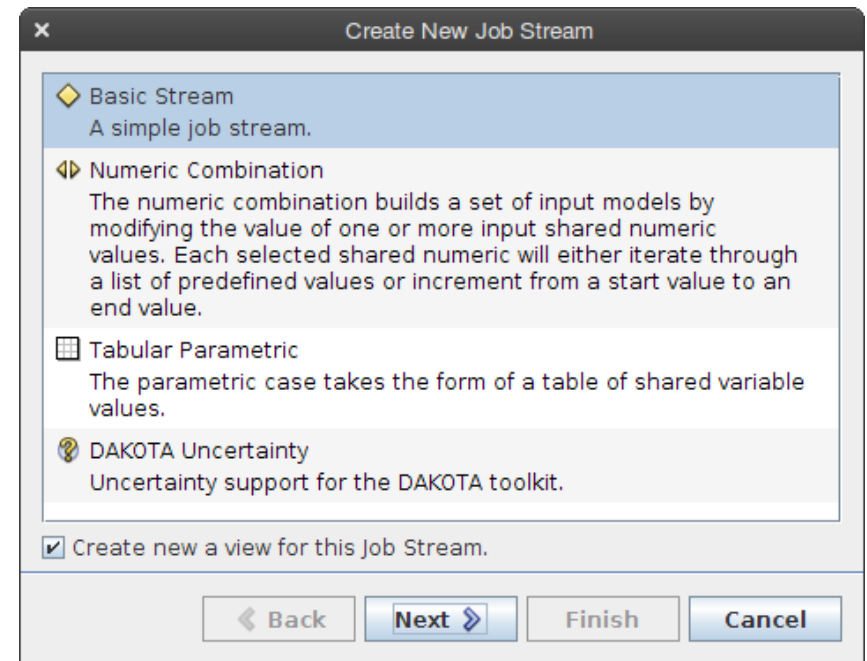
- Job Streams define how a series of applications should be executed.
- Graphically construct sequences of Job steps.
- Job Stream inputs can be model nodes or external files.
- A Model can contain any number of Job Streams.



# Job Stream Types

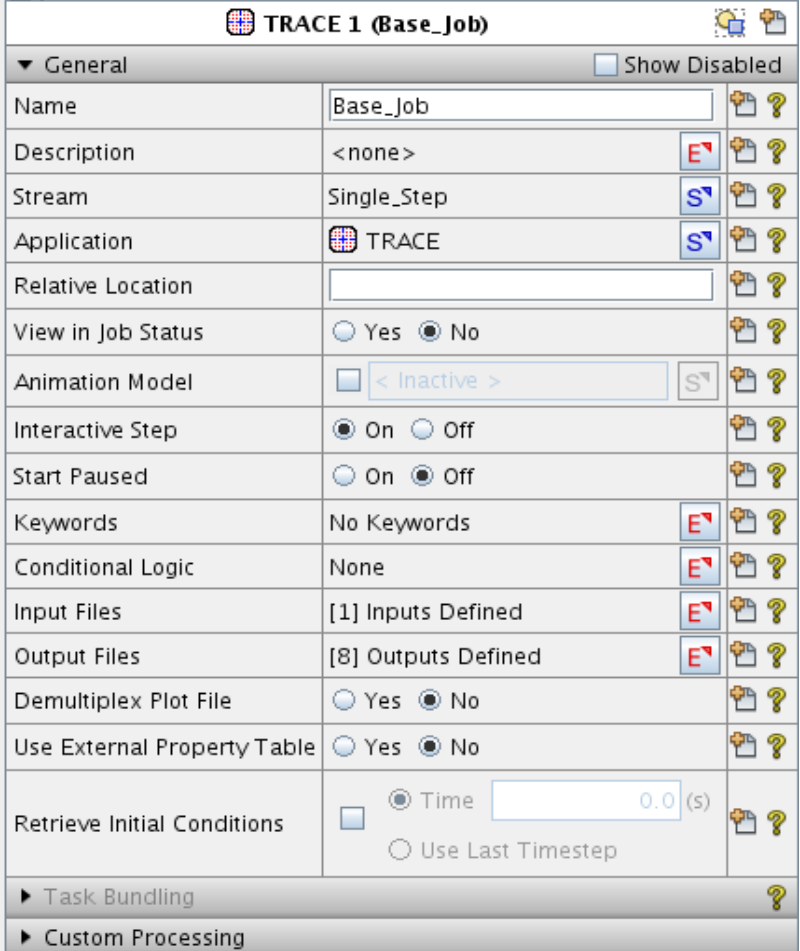
---

- Basic Job Stream
  - Model node generates a single file
  - Can contain multiple jobs steps
- Numeric Combination
  - Generate a set of parametric cases by specifying one or more interactive variables.
  - A set of values is specified for each variable.
  - Each combination of values corresponds to a parametric case.
- Tabular Parametric
  - Create a table of SNAP variables
  - Each row corresponds to a parametric case.
- DAKOTA Uncertainty



# Job Streams – Job Steps

- Job Steps can include:
  - Input Pre-processing (IC retrieval, Renodalize, etc.)
  - Analysis Code Execution (TRACE, PARCS, etc...)
  - Post-Processing (Plot generation, Data Extraction, etc...)
- Application Outputs can be archived after execution.
- Conditional Logic can be used to determine when a step should be included in the stream submission.



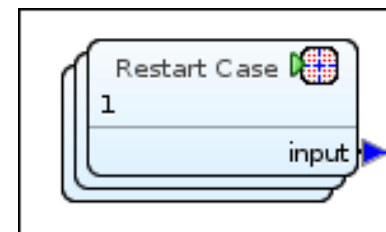
The screenshot shows the 'TRACE 1 (Base\_Job)' configuration window. It has a 'General' tab and a 'Show Disabled' checkbox. The window contains a list of properties and their values, with icons for help and search on the right of each row.

TRACE 1 (Base_Job)	
▼ General <input type="checkbox"/> Show Disabled	
Name	Base_Job
Description	<none>
Stream	Single_Step
Application	TRACE
Relative Location	
View in Job Status	<input type="radio"/> Yes <input checked="" type="radio"/> No
Animation Model	<input type="checkbox"/> < Inactive >
Interactive Step	<input checked="" type="radio"/> On <input type="radio"/> Off
Start Paused	<input type="radio"/> On <input checked="" type="radio"/> Off
Keywords	No Keywords
Conditional Logic	None
Input Files	[1] Inputs Defined
Output Files	[8] Outputs Defined
Demultiplex Plot File	<input type="radio"/> Yes <input checked="" type="radio"/> No
Use External Property Table	<input type="radio"/> Yes <input checked="" type="radio"/> No
Retrieve Initial Conditions	<input type="checkbox"/> <input checked="" type="radio"/> Time 0.0 (s) <input type="radio"/> Use Last Timestep
► Task Bundling	
► Custom Processing	

# Job Streams – Model Nodes

- Model Nodes represent a model in a Job Stream.
- Optionally select a Restart Case.
- A Job Stream can contain multiple Model Nodes.
- Can be Parametric if the Stream Type allows.
  - Define job stream locations where parametric values are applied.

Restart Case 1	
▼ General <input type="checkbox"/> Show Disabled	
Label	unnamed
Stream	ParametricStream
Description	<none>
Parametric	<input checked="" type="radio"/> True <input type="radio"/> False
Restart Case	<input checked="" type="checkbox"/> Restart Case



# Job Streams – External Files

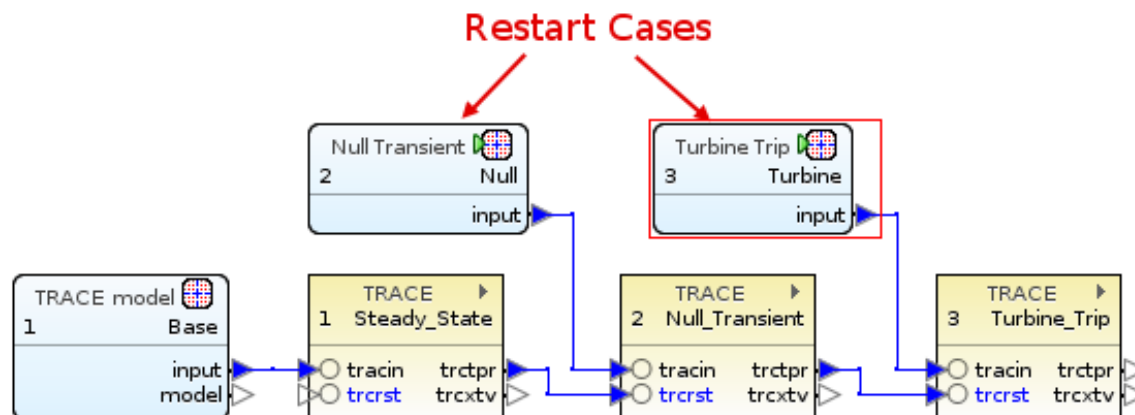
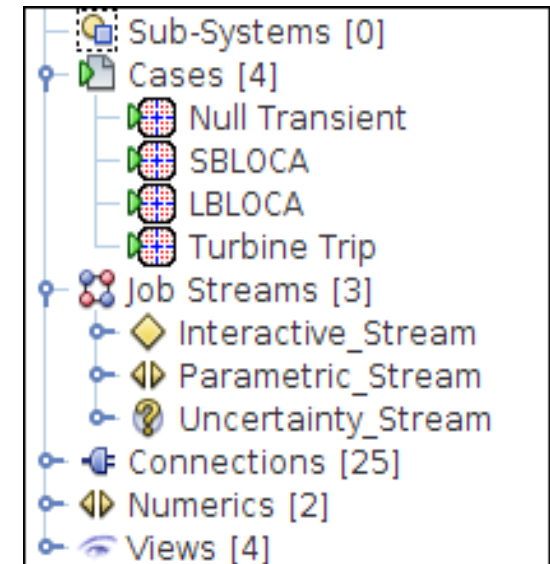
- Represent one or more files that will be included in the Stream.
- Each file has an explicit file type that determines where it can be used in a Job Stream. e.g. TRACE:XTV, PARCS:Output
- Files can be bundled with the stream, retrieved as they are needed during execution, or directly referenced.

External File 1 (trcrst)	
▼ General <input type="checkbox"/> Show Disabled	
Name	trcrst
Description	<none>
Stream	TRACE_FileTest
File Mode	Bundle With Stream
File Type	TRACE:TPR
File	file:///C:/Runs/PreviousRun/trctpr

File Set 2 (Plot_Files)	
▼ General <input type="checkbox"/> Show Disabled	
Name	Plot_Files
Description	<none>
Stream	TRACE_FileTest
File Type	TRACE:XTV
Files	[5] Selected Files
File Mode	Reference From Tasks

# Restart Cases

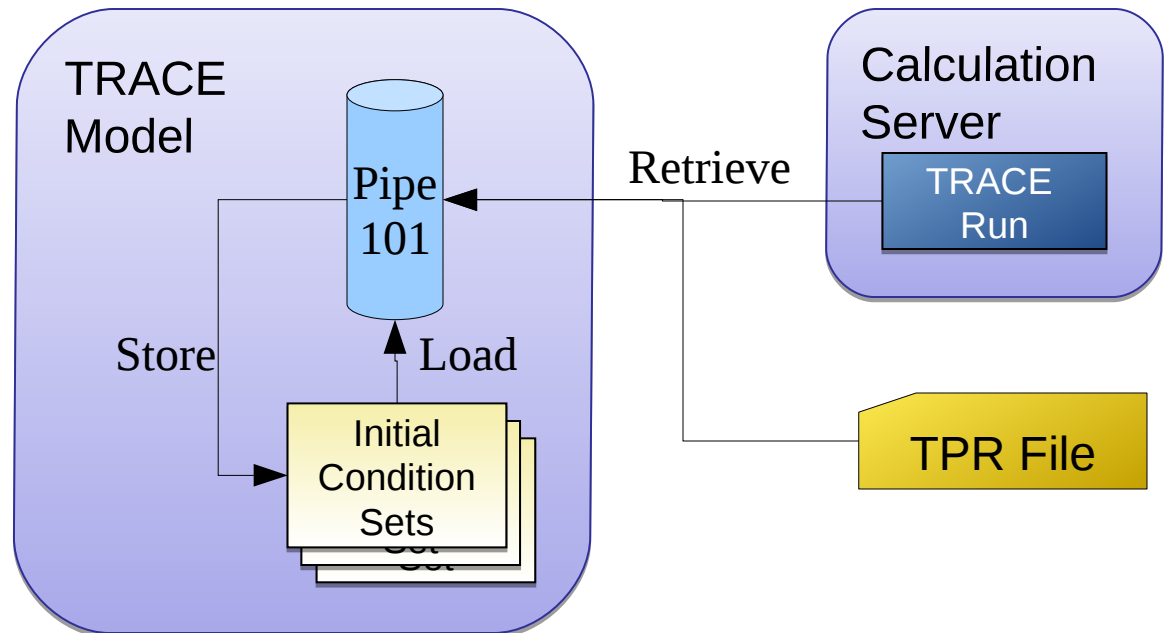
- A model can have any number of Restart Cases.
- Cases may be edited either graphically or by directly modifying the ASCII input.
- Base Model remains unaffected.
- Cases can be imported from and exported to ASCII files.



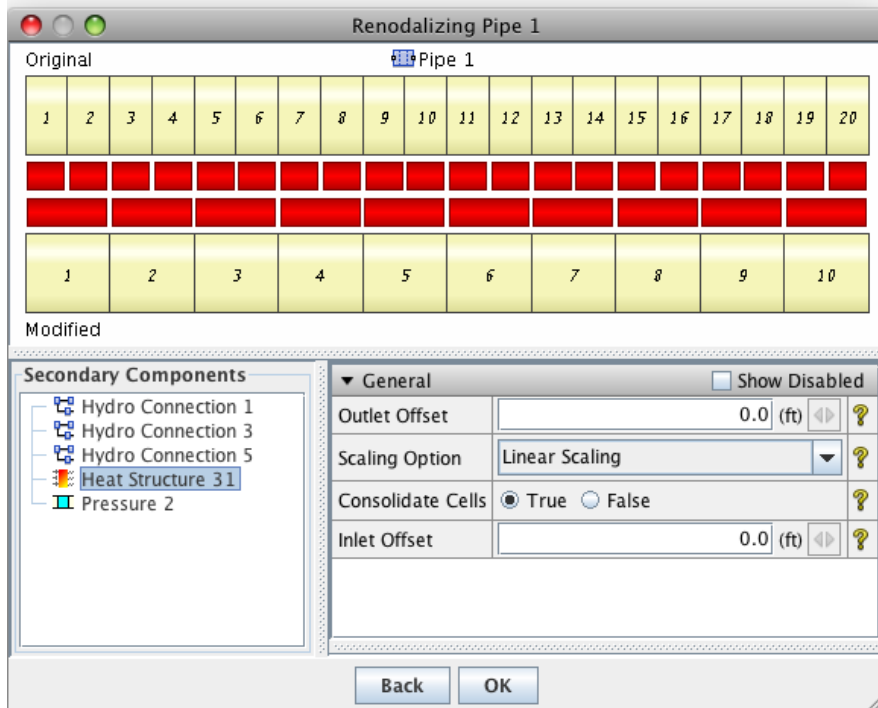


# Handling Initial Conditions

- Retrieve ICs from a Completed TRACE Calculation
  - Local TPR File or Calculation Server
  - Retrieved ICs Applied Directly to Components (Pipe 101)
- Manage Initial Condition Sets
  - Can Maintain Multiple Sets of ICs
  - Store the Current ICs as an “Initial Condition Set”
  - Load an Initial Condition Set into the Model
  - Initial Condition Sets are Saved with Model
- Automatic IC Retrieval
  - TRACE Job Step
  - Specified Time or Last Timestep
- Associate an IC Set with a Restart Case



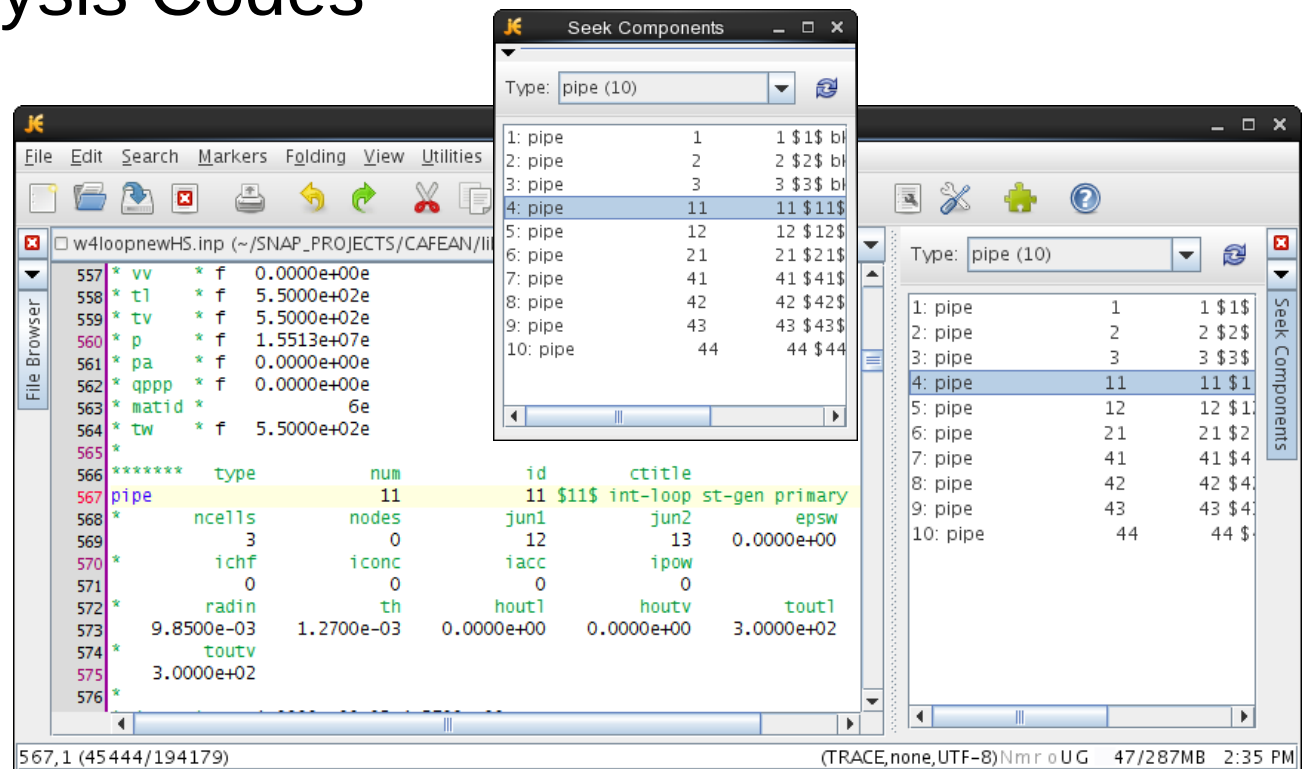
# 1D Renodalization



- Split and Merge individual cells
- Total axial length remains unaffected
- Displays changes in total elevation as well as to crossflow elevations
- Display compares the component before and after renodalization.
- Affected components are listed in a tree.
- Various options provided to control changes to components affected by the renodalization process.

# SNAP jEdit Integration

- Export to jEdit from the Model Editor
- SNAP Plug-in for jEdit
  - Input File Syntax Highlighting
  - Seek Components Window (Floating or Docked)
- Supported Analysis Codes
  - COBRA
  - MELCOR2x
  - PARCS
  - RELAP5
  - TRACE



# SNAP Diagnostic Output

---

- SNAP Applications write messages to screen and log files that can help diagnose issues.
  - Model Editor – me.screen
  - Job Status – js.screen
  - Calculation Server – cs.screen
- SNAP user preferences and screen files are stored in:
  - Windows 7 or Vista:  
C:\Users\<username>\.snap\2.0\
  - Windows XP:  
C:\Documents and Settings\<username>\.snap\2.0\
  - Unix/Linux:  
\$HOME/.snap/2.0/
- The Calculation Server also writes a log file to:  
<home>\.snap\2.0\log\calculation\_server.log
- Job Streams and Tasks write log files in their folders.
  - Job Stream – <stream name>.streamlog
  - Task – <task name>.tasklog

# Exercises

---

- Exercise 4. Introduction to Job Streams
- Exercise 5. Restart Editing
- Exercise 6. Model/Component Diff Viewer
- Exercise 7. 1D Component Renodalization

## Optional Exercises

- Exercise 8. Working with View Templates
- Exercise 9. Resource Bundle Import/Export
- Exercise 10. 2D Drawing Features
- Exercise 11. Using AptPlot and the ACS Plug-in