

SVN Plug-in Users Manual

Symbolic Nuclear Analysis Package (SNAP)



Version 2.0.0 - August 02, 2021

Applied Programming Technology, Inc.

**240 Market St., Suite 301
Bloomsburg PA 17815-1951**

Subversion Plug-in Users Manual

Applied Programming Technology, Inc.

by Ken Jones, Bill Dunsford, and Jake Kelly

Copyright © 2019-2020

***** Disclaimer of Liability Notice *****

The Nuclear Regulatory Commission and Applied Programming Technology, Inc. provide no express warranties and/or guarantees and further disclaims all other warranties of any kind whether statutory, written, oral, or implied as to the quality, character, or description of products and services, its merchantability, or its fitness for any use or purpose. Further, no warranties are given that products and services shall be error free or that they shall operate on specific hardware configurations. In no event shall the US Nuclear Regulatory Commission or Applied Programming Technology, Inc. be liable, whether foreseeable or unforeseeable, for direct, incidental, indirect, special, or consequential damages, including but not limited to loss of use, loss of profit, loss of data, data being rendered inaccurate, liabilities or penalties incurred by any party, or losses sustained by third parties even if the Nuclear Regulatory Commission or Applied Programming Technology, Inc. have been advised of the possibilities of such damages or losses.

Table of Contents

| | |
|--|----|
| 1. Introduction | 1 |
| 2. Revision Control Toolbar | 3 |
| 2.1. Update Button | 3 |
| 2.2. Commit / Add Button | 3 |
| 2.2.1. Commit Button | 3 |
| 2.2.2. Add Button | 4 |
| 2.3. Lock / Unlock Button | 4 |
| 2.4. Status Button | 4 |
| 3. Subversion Menu | 7 |
| 3.1. Add Model To Repository | 7 |
| 3.2. Rename Model in Repository | 7 |
| 3.2.1. Rename Dialog | 7 |
| 3.3. Lock / Unlock Model in Repository | 7 |
| 3.4. Resolve Conflicts | 8 |
| 3.5. Accept Local Changes | 8 |
| 3.6. Commit Local Changes | 8 |
| 3.7. Compare to Base | 8 |
| 3.8. Compare to Head | 8 |
| 3.9. Compare to Revision | 9 |
| 3.10. Revert to Base | 9 |
| 3.11. Update to Head | 9 |
| 3.12. Update To Revision | 9 |
| 3.13. Query File Status | 9 |
| 3.14. Show Commit Log | 10 |
| 3.15. Operation Availability | 11 |

Chapter 1. Introduction

The Symbolic Nuclear Analysis Package (SNAP) consists of a suite of integrated applications designed to simplify the process of performing thermal-hydraulic analysis. SNAP provides a highly flexible framework for creating and editing input for engineering analysis codes as well as extensive functionality for submitting, monitoring and interacting with the analysis codes. A single analysis model is stored in a platform independent binary format (MED).

Revision control systems are software packages that are used to maintain changes to documents. These systems allow multiple users to create, and maintain documents while preserving a history of what those changes are, and when those changes were made. The most recent version of a document is called the "head" revision. Applying local changes to a document into a repository is called committing. Retrieving changes made remotely to a document is called "updating" the document. If a document has both local and remote changes, the document is said to contain "conflicts".

This plug-in provides a user interface for interacting with the Subversion (SVN) revision control system. Models may be added to a SVN repository if they have been saved to a directory that is managed by Subversion. There are two user interfaces for accessing the SVN commands: the revision control toolbar, and the Tools menu. The toolbar contains the four most used features of revision control: update, commit, lock and status. The Tools menu contains the full list of SVN capabilities supported by SNAP.


Chapter 2. Revision Control Toolbar

The primary user interface for the Subversion plug-in is the revision control toolbar. The buttons in this toolbar provide quick access to revision controls for the currently selected model. Changing the currently selected model will automatically update the toolbar to reflect the new model. When a model is graphically editing a restart case the toolbar will be disabled.



Figure 2.1. Revision Control Toolbar

2.1. Update Button

 This button will apply remote modifications to the current model. This will attempt to retrieve the current head of the repository. If the operation succeeds, the current copy of the model will close, and the new model will be opened. A warning dialog will be displayed if the working model has unsaved changes or has been locally modified. Updating the file will discard local changes.

The update button is a shortcut for updating to the head revision of the repository. If the current file does not exist in the head revision (ie: renamed, or removed), the update operation will fail and a warning will be displayed. The Update To Revision menu item will show the full commit log history for the current model, and allow the renamed file to be selected.

2.2. Commit / Add Button

This button either adds the current model to a repository or commits the current model as the new head if it already exists in the repository. The icon that is displayed in the button indicates the behavior that will occur when it is pressed.

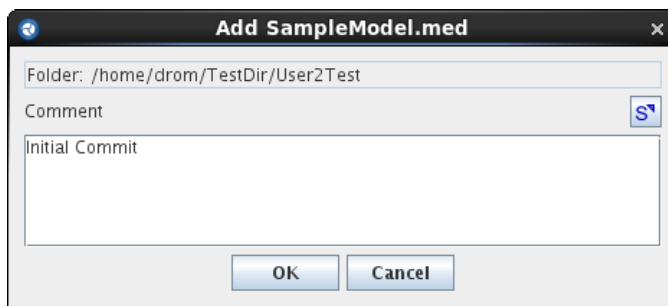



Figure 2.2. Commit Model Dialog

2.2.1. Commit Button

 This button commits local changes to the head of a SVN repository. A commit message dialog, shown in [Figure 2.2, “Commit Model Dialog”](#) will prompt the user for a log message, after which the model will be added as the new head of the repository. This button is only enabled when the current model has local modifications, and there are no remote changes on the server.

2.2.2. Add Button







This button adds the current model to a repository. This button only appears if the current model is not managed by SVN but is saved to a directory that is managed by SVN. A commit message dialog, shown in [Figure 2.2, “Commit Model Dialog”](#) will prompt the user for a log message, after which the model will be added to the repository and committed. The commit operation in this instance is performed automatically after adding the file.

2.3. Lock / Unlock Button







This button locks and unlocks the current model on the repository. A locked file prevents changes from being committed. The button icon displays the current lock state for the model. When the current model is locked, no user will be able to commit changes to the file. The different lock states, and the result of pressing the button in each state are described below.




-  *Unlocked* - The current model is not locked locally or on the server. Pressing the button will request a lock on the current model.
-  *Local Lock* - The current model is locked locally by the current user. Pressing the button will unlock the model on the repository.
-  *Remote Lock* - The current model has been locked remotely. Pressing the button will request that the current lock be broken.
-  *Broken Lock* - A local lock on the current model has been broken by a remote user. Pressing the button will request a lock on the current model.

2.4. Status Button



This button displays the current status of the file. Pressing this button will update the current status from the server and display the status dialog. The full list of possible status icons are displayed below.

-  *No Working Directory* - The current model either does not have a local save file or the save file is not in a directory managed by SVN.
-  *New File* - The current model is saved to a SVN managed directory but has not been added to the repository.
-  *Up To Date* - The current model is a clean copy of the head of the repository. No local or remote modifications have been made.
-  *Local Modifications* - The current model has local modifications, and no remote modifications have been made.

-  *Remote Modifications* - The current model has no local modifications but newer versions are available on the repository.
-  *Conflict (pre-merge)* - The current model has local modifications and new versions are available on the repository.
-  *Conflict (merging)* - Conflict resolution between local modifications and remote modifications has begun.

Chapter 3. Subversion Menu

The Tools menu on the main menu bar, and the Tools sub-menu on the right-click pop-up menu both include the Subversion menu. This menu contains the full list of Subversion operations supported by SNAP.

3.1. Add Model To Repository



This menu item adds the current model to a Subversion repository when it has been saved to a directory that is managed by SVN. When pressed, a commit message dialog, shown in [Figure 2.2](#), “Commit Model Dialog” will prompt the user for a log message. After entering the commit log message, the model will be added and committed to the repository. The commit operation in this instance is performed automatically.

3.2. Rename Model in Repository



This menu item renames the current file in the repository. This deletes the current file on the repository and adds the model under a new name. When pressed, the rename dialog described in [Section 3.2.1](#), “Rename Dialog” will prompt the user for a new file name and a commit message. The model is then renamed in the repository. This will also commit any local modifications, and change the local save file on disk.

3.2.1. Rename Dialog



Figure 3.1. Rename Dialog

This dialog allows the user to select a new file name for the current model and to enter a commit log entry for the SVN rename operation. This dialog opens when the Rename Model In Repository menu item is selected.

3.3. Lock / Unlock Model in Repository



SVN supports locking the head of the repository to prevent remote modification. This is extremely valuable when working with binary files that cannot be automatically merged by SVN. This menu item provides the capability to either lock, or unlock the current model in the repository depending on the current lock state. If the current model is unlocked on the server,

this menu item will lock the file. If the current model is locked on the server this menu item will unlock the file. If the file has been locked by a remote user, a confirmation dialog will be presented to allow the lock to be broken.

3.4. Resolve Conflicts



This menu item only appears if the working model has changes, remote changes have been made to the repository, and the model supports graphical value comparison. When pressed Subversion updates the local directory to contain the base version of the working model, and the current repository head. The multi-component comparison dialog opens showing the differences between the working model and the repository head. Updates may only be merged from the repository head to the local version.

Once initiated the conflict resolution state persists between model editor instances. The working model may be saved, and the model editor closed and the conflict resolution resumed at a later time. To complete resolving the conflict the user must press the Accept Local Changes subversion menu item.

3.5. Accept Local Changes



This menu item ends the conflict resolution state for the current model. Note that this does not commit the local changes to the repository. This calls the SVN command that identifies the local conflicts as having been resolved and allows the local changes to be committed as the new repository head. Note that any differences from the repository head not merged into the working model will be lost when the model is committed.

3.6. Commit Local Changes



This menu item commits the current model to the head of the repository. When pressed, a commit message dialog, shown in [Figure 2.2, “Commit Model Dialog”](#), will prompt the user for a log message. After entering the commit log message the model will be committed as the new head of the repository. This button is only enabled when the current model has local modifications, and there are no remote changes on the server.

3.7. Compare to Base



This menu item is enabled only when the model supports graphical value comparison and the model is not new, in a directory not managed by SVN, or un-changed from the base. When pressed the multi-component comparison dialog opens showing the changes that have been made locally from the repository. Updates may only be merged from the repository head to the local version.

3.8. Compare to Head



This menu item is enabled only when the model supports graphical value comparison and the model is not new or in a directory not managed by SVN. When pressed the multi-component

comparison dialog opens showing the changes between the local model and the repository head. Updates may only be merged from the repository head to the local version.

3.9. Compare to Revision



This menu item is enabled only when the model supports graphical value comparison and the model is not new or in a directory not managed by SVN. When pressed the Revision History dialog is shown. Once a specific revision has been selected the multi-component comparison dialog opens showing the changes between the local model and repository head. Updates may only be merged from the repository head to the local version.

3.10. Revert to Base



This menu item replaces the current file with the version checked out from the repository. Any local modifications will be lost.

3.11. Update to Head



This menu item replaces the current model with the head version of the repository. A warning dialog will be displayed if the working model has unsaved changes or has been locally modified. Updating the file will discard local changes.

3.12. Update To Revision



This menu item retrieves the commit log for the current file on the server and displays it in a dialog. The local file will be replaced with the selected version. This operation supports navigating through renamed files, and will update both the current file and the selected file to the desired version. This will remove the original file from disk, and download the selected renamed file. A warning dialog will be displayed if the working model has unsaved changes or has been locally modified. Updating the file will discard local changes.

3.13. Query File Status



This menu item updates the current file status. The icon of this menu item displays the current file status as described in [Section 2.4, “Status Button”](#). When the operation is complete, the File Status Dialog will be displayed, which provides the details on the current file.

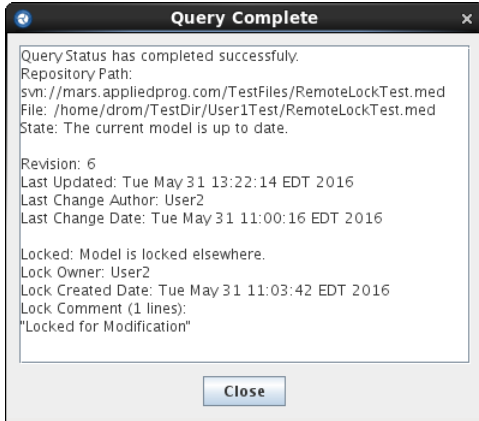



Figure 3.2. File Status Dialog

3.14. Show Commit Log

 This menu item retrieves the full commit log for the current file in the current working directory. This log will include file renames. An example of the Log History Dialog is shown below in [Figure 3.3, “Log History Dialog”](#).

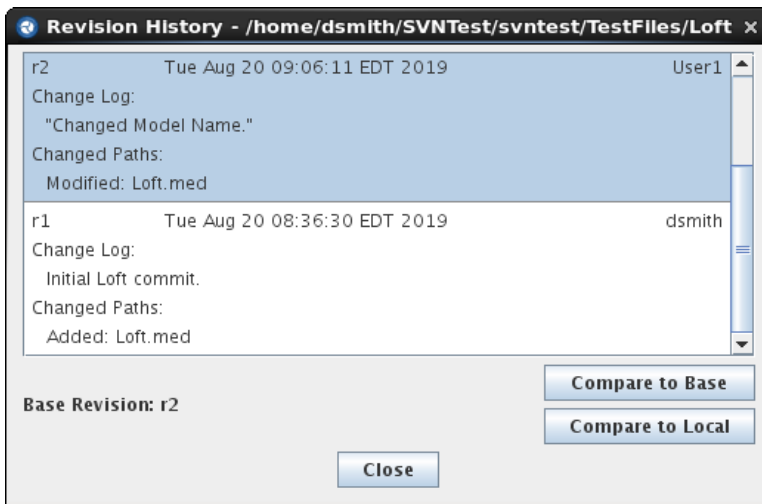


Figure 3.3. Log History Dialog

The Compare and Compare to Local buttons only appear if the working model supports graphical value based comparison. The Compare button allows comparing two revisions against one another or comparing a selected revision against the base revision. In either case the remote revisions will be downloaded and opened in the model editor background. The Multi-Component Comparison dialog will be opened to show the differences between the revisions. Differences will be flagged but values cannot be merged into either model. The Compare to Local button compares the local model (with modifications) to the selected revision.

3.15. Operation Availability

| | Description |  |  |  |  |  |  |  |
|--|---|---|---|---|--|---|---|---|
|  Add Model | Add and commit current model to repository. | X | ✓ | X | X | X | X | X |
|  Rename Model | Rename the current model in the repository. | X | X | ✓ | ✓ | X | X | X |
|  Lock/ Unlock Model | Lock or unlock current model in repository | X | X | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 3.1. SVN Operation Availability

| | Description |  |  |  |  |  |  |  |
|---|--|---|---|---|--|---|---|---|
|  Resolve Conflicts | Begin resolving conflicts between the repository head and the working model. | X | X | X | X | X | ✓ | ✓ |
|  Accept Local Changes | Accept local changes and clear conflict state. | X | X | X | X | X | X | ✓ |
|  Commit Local Changes | Commit local changes to the repository. | X | X | X | ✓ | X | X | X |

Table 3.2. SVN Operation Availability (continued)

| | Description | | | | | | | |
|---------------------|---|---|---|---|---|---|---|---|
| Compare to Base | Compare local model to base. | X | X | X | ✓ | ✓ | ✓ | ✓ |
| Compare to Head | Compare local model to head. | X | X | ✓ | ✓ | ✓ | ✓ | ✓ |
| Compare to Revision | Compare local model to a specific revision. | X | X | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 3.3. SVN Operation Availability (continued)

| | Description | | | | | | | |
|--------------------|--|---|---|---|---|---|---|---|
| Revert to Base | Revert local model to base (discard any changes). | X | X | X | ✓ | X | ✓ | ✓ |
| Update to Head | Update local model to head (discard any changes). | X | X | X | X | ✓ | ✓ | ✓ |
| Update to Revision | Update local model to specific revision (discard any changes). | X | X | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 3.4. SVN Operation Availability (continued)

| | Description | | | | | | | |
|------------------------|--|---|---|---|---|---|---|---|
| Query Model Status | Query and update the current model's status. | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Show Commit Log | Show commit log | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 3.5. SVN Operation Availability (continued)
