PharmaSUG 2015 - Poster PO09

Adding Subversion® Operations to the SAS® Enhanced Editor

Oscar F. Cheung, PPD, Morrisville, NC
Kenneth W. Borowiak, PPD, Morrisville, NC

ABSTRACT

Apache™ Subversion® (SVN) is an open-source version control system whose use is growing amongst clinical trial programming departments. SVN can be used for versioning many files types, such as Microsoft Word and Excel, XML, RTF and SAS® data sets and logs. However, clinical trial programmers have a valuable tool in SVN at their disposal for tracking the development and maintenance of SAS programs. This paper demonstrates how to enable two of the most common SVN operations, namely SVN ADD and SVN COMMIT, in the SAS Enhanced Editor.

INTRODUCTION

Version control is the management of access and changes to files, which is partially accomplished by tracking the *revisions* of each file in a *repository* (i.e. a type of database). Once a revision of a file is *committed*, you can track the who, what, when and why of file revisions. You can think of a commit of a file as a 'save operation on steroids'. Apache™ Subversion® (SVN) is an open-source version control system whose use is growing amongst clinical trial programming departments.

SAS programs are an ideal candidate for being managed under version control. Programs are often revised a number of times as they are being developed, modified for changes in specifications and scope, errors are found and peer reviewed. The number of programs to support a clinical trials project can be quite voluminous and may need to be maintained over a number of years. Moreover, there may be multiple programmers who will be editing the file, either somewhat concurrently, in collaboration or in succession. The principle of *traceability* can be applied to programs just as it is the data files provided to regulatory agencies.

In order to track the evolution of a SAS program, even with version control, the onus still resides with programmers. They must still commit their programs in order to reap the benefits of version control, such as not losing any edits and being able to restore code that has since been removed. The easier it is to commit files the more likely it will be done in the optimal way with the optimal frequency. Committing programs with SVN can be done with command-line calls (e.g. batch scripts or system commands in a SAS program) or with its GUI interface TortoiseSVN. For programmers doing program development with SAS Enhanced Editor, the latter of the choices is not ideal because the commit operation needs to happen outside of the SAS environment. We will demonstrate in this paper how to add the SVN operations ADD and COMMIT to the Enhanced Editor environment to facilitate the ease in which SAS users can apply some key version control activities to program development.

USAGE

The buttons circled below in Figure 1 are for two common SVN operations. The red PLUS sign is for SVN ADD, which we choose because it is similar to the blue plus icon overlay that appears in a Windows directory when the operation is performed. The ADD operation tells the SVN server that a file or directory exists. A file only needs to be declared once with ADD and is a prerequisite for the COMMIT operation. The ADD button applies to the active program in the SAS session. No further action is needed after you click the ADD button.

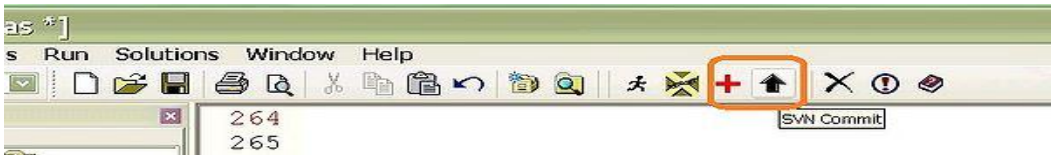


Figure 1 – Toolbar buttons for SVN Operations

The black ARROW is for the SVN COMMIT operation, which initially places a file under version control. COMMIT then tracks changes between programs revisions with each subsequent commit. A SAS program that has been loaded into an interactive session which has been modified either in that session or externally with another editor (e.g. TextPad or UltraEdit) can be committed to the repository, with or without an explicit SAVE of the file. Clicking the COMMIT icon will launch the Tortoise

SVN Commit window with the SAS program name pre-populated in the message window, as shown in Figure 2. A best practice is to add a meaningful message describing the change(s) to the program associated with the commit[1].

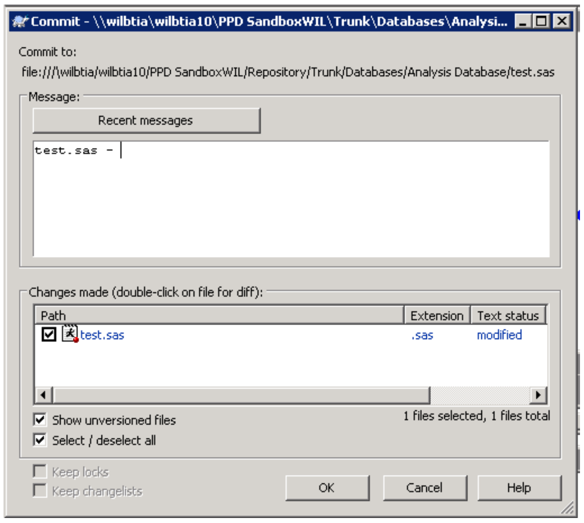

Commit - \\wilbtia\wilbtia10\PPD SandboxWIL\Trunk\Databases\Analysi...
Commit to:
file:///\wilbtia/wilbtia10/PPD SandboxWIL/Repository/Trunk/Databases/Analysis Database/test.sas
Message:
Recent messages
test.sas -
Changes made (double-click on file for diff):
Path
Extension
Text status
test.sas
.sas
modified
Show unversioned files
Select / deselect all
1 files selected, 1 files total
Keep locks
Keep changelists
OK
Cancel
Help

Figure 2 – SVN window prompted by the Commit toolbar button

To complete the commit process you need to click the **OK** button. Note that the operation pertains to only a single program, namely the program in the active window of the SAS session. Programs from other project folders pointing to the same repository location that are active in the SAS session can also use the SVN toolbar buttons.

INSTALLATION

This section describe some methods for adding the SVN buttons to the Enhanced Editor, which have been tested under SAS V9.2 in Windows XP and SAS V9.3 in Windows7 environments. Adding the SVN buttons to your SAS profile will entail using Display Manager (DM) commands[2]. To add the SVN buttons to your SAS profile, make sure only one interactive SAS session is open. You may want to close all interactive SAS sessions and start a new session to ensure you can write to your SAS user profile catalog. Do the following steps twice, once for the SVN ADD button and then for the SVN COMMIT button.

- Right click in the empty space in the buttons toolbar; then choose **Customize** from the menu, as shown in Figure 3.

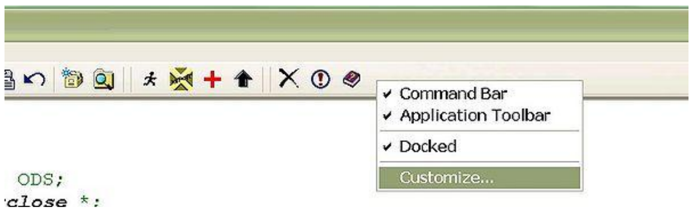
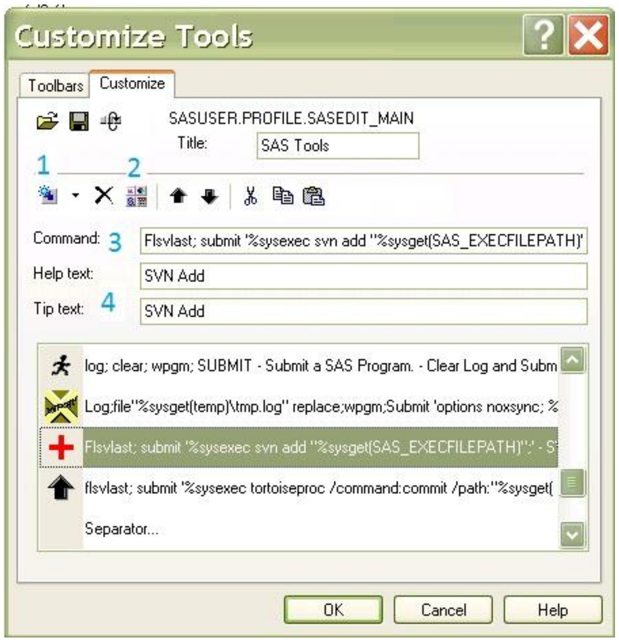


Figure 3 – Launching the CUSTOMIZE TOOLS interface

[1] Williams[3] mentions adding a hook-script to ensure a non-null message is added, which our suggested implementation would satisfy. However, a more meaningful message is appropriate. Stating what and why the file was edited are the key features of a good message. The specific details of the change can be determined by DIFFing two revisions of a file, which show the differences between them.

[2] See Carpenter [1] for more details on using the Display Manager.



SASUSER.PROFILE.SASEDIT_MAIN

Title: SAS Tools

Command: Flsvlast; submit '%sysexec svn add "%sysget(SAS_EXECFILEPATH)'

Help text: SVN Add

Tip text: SVN Add

log; clear; wpgm; SUBMIT - Submit a SAS Program. - Clear Log and Subm

Log;file"%sysget(temp)\tmp.log" replace;wpgm;Submit 'options noxsync; %

Flsvlast; submit '%sysexec svn add "%sysget(SAS_EXECFILEPATH)";' - S

flsvlast; submit '%sysexec tortoiseproc /command:commit /path:"%sysget(

Separator...

Figure 4 – Steps for creating the SVN buttons

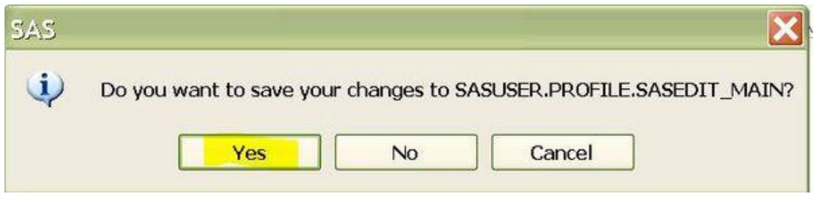
- 1- Click to choose **Add Blank Tool**
- 2- Click to browse and choose an icon for the button
- 3- Enter the following the commands in the input field for SVN ADD:

```
flsvlast; submit '%sysexec svn add "%sysget(SAS_EXECFILEPATH)";'
```

- 4- Enter brief description in both help and tip input field

Now repeat Steps 1 to 4 above for SVN COMMIT, except in Step 3 use the following commands:

```
flsvlast; submit '%sysexec tortoiseproc /command:commit
/path:"%sysget(SAS_EXECFILEPATH)" /logmsg:"%sysget(SAS_EXECFILENAME)- " /closeonend;'
```



Do you want to save your changes to SASUSER.PROFILE.SASEDIT_MAIN?

- Click **Yes** to finalize customization

Below is a description of the individual commands, functions and variables used in the implementation.

- `flsvlast` – DM statement command to save the current file.
- `submit '…code…'` – DM statement command to run the code enclosed in the quotes.

- `%sysexec` – Statement to issue operating environment commands. It is similar to the X statement, but without the commands enclosed with quotes.
- `%sysget( <environment variable>)` – Macro function that returns the value of an environment variable.
- `SAS_EXECFILEPATH, SAS_EXECFILENAME –` SAS environment variables. Example: SAS_EXECFILEPATH is the full path filename, such as `c:\study\temp\test.sas`. SAS_EXECFILENAME is the name of the active program. Both environment variables are available for use as soon as program is open. However in earlier versions of SAS (i.e.V9.1 and V8), these variables are only available after a program is run. This is a limit of this tool.
- `tortoiseproc /command:commit` - GUI style Tortoise SVN command for committing file.
- `/path:"%sysget(SAS_EXECFILEPATH)"` – A fully qualified path with filename of program to commit.
- `/logmsg:"%sysget(SAS_EXECFILENAME)- "` - Option to pre-fill the commit log message with the current program name.

DISTRIBUTION

To distribute this tool to the SAS profile of other users, first export a copy of your user catalog to a common area, such as a network library. This can be accomplished using PROC CATALOG.

```
libname sb '<some folder>';

*** Export SAS Toolbox buttons catalog ***;
proc catalog ;
  copy in  = sasuser.profile
       out = sb.svnbuttons2015 ;
  select sasedit_main / et = toolbox ;
quit ;
```

Other users can then import the catalog from the previous step by executing a similar PROC CATALOG step, with the values of the IN= and OUT= options swapped.

```
libname sb '<location of catalog created in previous step>';

*** Import SAS Toolbox buttons catalog ***;
proc catalog ;
  copy in  = sb.svnbuttons2015
       out = sasuser.profile ;
  select sasedit_main / et = toolbox ;
quit ;
```

Note that when importing the catalog it will overwrite any existing custom toolbar buttons in the users SAS profile. If a user already has other custom buttons in their profile then they should follow the steps in the INSTALATION section of this paper.

CONCLUSION

The authors believe that having SVN operation toolbar buttons directly on the same user interface of SAS Enhanced Editor increases efficiency and saves time from switching windows. It also helps in promoting the good habit of frequently writing reasonable comments when committing programs.

REFERENCES

[1] Carpenter, Arthur, "*Doing More with the SAS® Display Manager: From Editor to ViewTable - Options and Tools You Should Know*", SAS Global Forum 2012 Proceedings
http://support.sas.com/resources/papers/proceedings12/151-2012.pdf

[2] Carpenter, Arthur, "*The Path, The Whole Path, And Nothing But the Path, So Help Me Windows*", SAS Global Forum 2008 Proceedings
http://www2.sas.com/proceedings/forum2008/023-2008.pdf

[3] Williams, Tim, "*A Guide to Deploying Subversion for Version Control of SAS Programs in the Pharmaceutical Industry*", Pharmaceutical Software Exchange 2013 Proceedings

http://www.lexjansen.com/phuse/2013/AD/AD01.pdf

[4] Wirtz, Oliver, "*How to Add TortoiseSVN-Functionality to SAS® Enhanced Editor*", Pharmaceutical Software Exchange 2013 Proceedings

http://www.lexjansen.com/phuse/2013/cc/CC03.pdf

ACKNOWLEDGMENTS

The authors would like to thank Wes Murray for reviewing the paper.

DISCLAIMER

The content of this paper are the works of the authors and do not necessarily represent the opinions, recommendations, or practices of PPD.

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Oscar Cheung
PPD
3900 Paramount Parkway
Morrisville, NC 27560

Fan.Cheung@ppdi.com
(607) 429-8776

Ken Borowiak
PPD
3900 Paramount Parkway
Morrisville, NC 27560

Ken.Borowiak@ppdi.com
(919) 456-5373