Paper RF-16 MWSUG2015

Making Historical Versions of SAS® Code While Developing in Enterprise Guide 7.1

Roger D. Muller, Ph.D., Data-To-Events, Inc., Carmel, IN

ABSTRACT

A major new feature in SAS Enterprise Guide version 7.1 is the capability to retain historical versions of the SAS code as the programming is proceeding. The programmer has the option to return to a previous point where a "wrong turn in the road" was taken and resume the programming effort from there. All of this is done without keeping a folder of assorted backup versions with cleverly-coded sequential names. This facility essentially contains a menu item to commit the version to the history of the file, and a menu item to review this history and select one to restore if desired. This feature is only for the SAS code files stored internally (embedded) within the Enterprise Guide project. The capability is not for the project itself. Likewise it is not for SAS code files stored externally to the project. External files can be historically versioned by a 3rd party program such as Subversion which will be briefly addressed. These approaches are referred to as "Git" repositories for version control (the acronym has no direct computer meaning, but originates in English slang).

A simple program workflow will show the use of the new versioning tool, where it is used and where it is not used.

INTRODUCTION

SAS Enterprise Guide version 7.1 has introduced the capability of making historical versions of user-coded SAS program files when they are created as embedded SAS code files within the SAS Enterprise Guide project file (*.egp files) While this is a nice capability, it falls far short of a total versioning requirements that would be ideal in SAS EG.

This paper will address three different types of “SAS files” that are encountered while using SAS EG and worthy of versioning, backup and archiving:

1. The SAS EG project file (.egp file). This file contains information about the project, external SAS code files referenced, and internal SAS coded files developed within EG’s editor and stored as embedded files within the project. Information about Process Flows are stored here. There may be SAS code developed within the Process Flow window that is not stored as a visible SAS code file. The contents of the “egp” file are shown in the Project Tree Window at the top left of the screen.

2. SAS program code files developed by a programmer that are “embedded” into the projects. These files are visible in the Program section of the Project Tree window . They are not visible in system file directories when such directories are viewed by tools such as Windows File Explorer. They do not have “shortcut symbols” on the icon in the project tree.

3. SAS program code files that were developed by a programmer that are “external” to the project and are visible not only in the “Programs” portion Project Tree window, but are visible in external file directories when such directories are viewed by tools such as the Microsoft Windows File Explorer. The icons in the Project Tree window to have the shortcut symbol, indicating they are linked.

Versioning vs. Backup vs. Archiving

For purposes of definition here, generally industry-accepted terms for backup would be to have copies of entire files stored at some location that have a date-time stamp that are available for retrieval at some future point. Archiving would be defined as storing backup files in such fashion that they may be retained for a much longer time than simple backups. Such time may be well beyond the actual use time for the file and may be continued to a time when the file is no longer considered to be “on-line”. Both archiving and backup traditionally involve central IT departments

Versioning is referred to in the context of being able to retain changes made to the file during the file’s development with such changes being recorded at the desire of the programmer, at fixed time intervals or when programs are saved. Versioning is done within the file itself and roll-back is easy to do by the programmer. Multiple versions are available to roll-back to. Comparisons of differences should be easy to view retrieval should be easy. The changes should be labeled as to which user made them. In addition, the programmer should be able to compare the current content of the editor (which has not yet been saved) to the last saved version and see differences identified. Ideally “snippets” of the code could be highlighted with the Microsoft Windows copy function (Ctrl-C) and pasted into the current document (Ctrl-V), rather than being bound to retrieve entire files.

A REVIEW OF THE SAS ENTERPRISE GUIDE PROJECT STRUCTURE

Let's review some of the files associated with the use of SAS EG and look at a simple workflow containing some of these files. Indeed, it is possible (although probably very unrealistic) to have a SAS Enterprise Guide project that contains no visible or apparent SAS code programs that were developed by conventional programming.

Programming With the Process Flow Window – No Code Developed by a SAS Programmer

By dragging objects on to a process flow window (see the first figure later in this paper), a report can be obtained that does not list any program files in the project tree window. All of the details of doing this are outside the scope of this paper. Is there SAS code "under the hood"? Of course! But it is not visible and does not require a separate "sas code" file. Can such code be exported out to a "regular sas file"? Of course! But it is not necessary. The program can be run, generate meaningful output and retained for future use by simply storing as an "egp" file. Is this the way most traditional SAS programmers are used to working? No, but it does have its place.

Programming With Conventional SAS Files – Some Thoughts on Their Use.

As discussed earlier, programmers develop and store code as either embedded or external files.

What's Better --- Embedded SAS program files or External Program Files?

The answer to that like so many things in life is: "That Depends!". An all-inclusive discussion of this is worthy of a separate paper in itself and is beyond the scope of this discussion.

Embedded. Briefly, if you are a "stand-alone" programmer doing your own thing on your own project, embedded files can be a great way to work, particularly early during the program development process. The only caveat to this is as that as you accumulate many projects and programs, it can become increasingly difficult to find historic code unless you have a meticulously maintained a log of some sort of what is where. A tool developed by Hemedinger(2013) can be used to "look around" inside project files to search for embedded SAS code files.

External Files. If you are working in a programming team environment, having files stored externally in well-organized directories can make them much more visible to other programmers and thus more retrievable. Most traditional programmers are more comfortable with external files and continue to do so.

File Movement. Can files be moved from embedded to external and vise-versa? Absolutely! Just beware that you can get duplicate copies and versions of files all over the place if you are not careful and can make it really difficult to identify the current most-recent version of the program. Again a complete discussion on the pros and cons of doing this are outside the scope of this presentation and are worthy of a separate presentation. Should files be moved from external status to embedded status simply to take advantage of this new capability? Possibly! It depends! If you do so, make sure you take steps to not get multiple versions of the files residing both externally and embedded. Also, I would suggest doing this only if you are the sole programmer on a project.

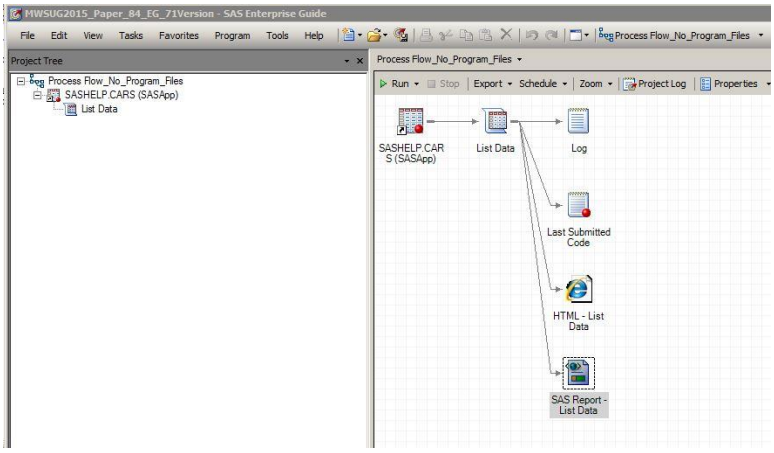
The New Versioning Control. What does this new versioning control in SAS 7.1 do.? It provides a user-controlled capability of creating versions of EMBEDDED SAS programs as the code is being developed. The user has the capability of retrieving code from previous versions "before it was messed it up" and restore the code to the previous content.

Does this new capability have anything to do with creating versions of (1) the EGP project file itself, and (2) Externally referenced SAS files? NO, ABSOLUTELY NOTHING AT THIS TIME. YOU ARE ON YOUR OWN FOR EITHER CREATING VERSIONS OF THESE OR FINDING OTHER TOOLS THAT MAY BE AVAILABLE within your organization's file management capabilities.

Other Tools. Versioning tools available from sources other than SAS for files such as Enterprise Guide Project files and external SAS code files are available. You are on your own to find out about them in your own organization. Often they are on servers and may or may not be highly visible and/or advertised by your IT organization. I am not aware of such tools being available on stand-alone PC's. Start asking your systems personnel if you have versioning control software, "Git" software, "snapshotting" Subversion, etc. I recently learned that snapshotting was already being performed on some of the drives I was accessing. No one told me about it and the folder where all of this was located was set to hidden! Nice! Once "show hidden folders" was activated, a folder named "Snapshot" appeared. It contained all kinds of versions of files available to me: hourly versions for the previous 24 hours, daily versions for the previous week, and then weekly versions out through 3 weeks.

A SAS Enterprise Guide Project File Under Development Using Different Techniques

1. **SAS Enterprise Guide with No User Coded Programs**. In the adjacent figure, an Enterprise Guide Project (egp) file has been created by dragging the dataset SASHELP.CARS into the process flow window. A "List Data" task has also been dragged to the process flow. The Process Flow has been renamed "Process_Flow_No_Program_Files". The "Run" button was hit and a log, HTML output and a SAS Report List File generated. This entire project has been given the name "MWSUG2015_Paper_84".



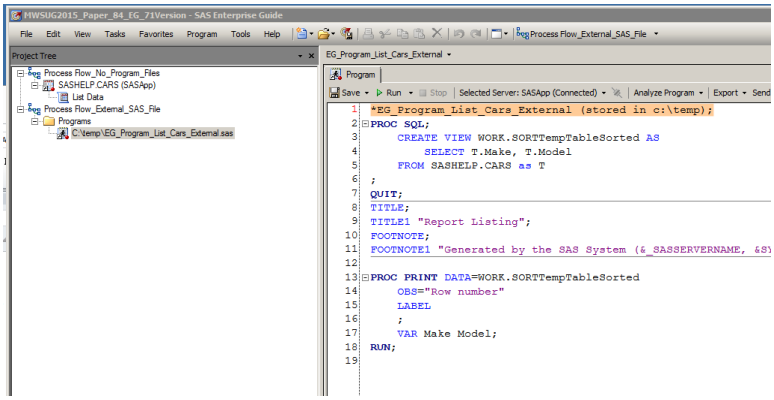
   a. Is there a "user-coded" SAS program file visible? No.

   b. Is there SAS code "behind the scenes under the hood"? Yes, but you don't have to worry about it. Enterprise Guide manages this.

   c. Does the MWSUG2015_Paper_84.egp file need to be backed up, versioned etc. by the user? Yes, the new feature does not help do this.

   d. Does the new versioning control feature in EG 7.1 have anything to do in helping create backup versions of this project? NO, ABSOLUTELY NOTHING.

2. **SAS Enterprise Guide With an External SAS Program File Linked**. A second Process Flow has been added (File – New – Process Flow) and renamed "Process_Flow_External_SAS File". This SAS Program can be created internally in EG and then stored externally, or it can created by an external editor and linked and opened here.



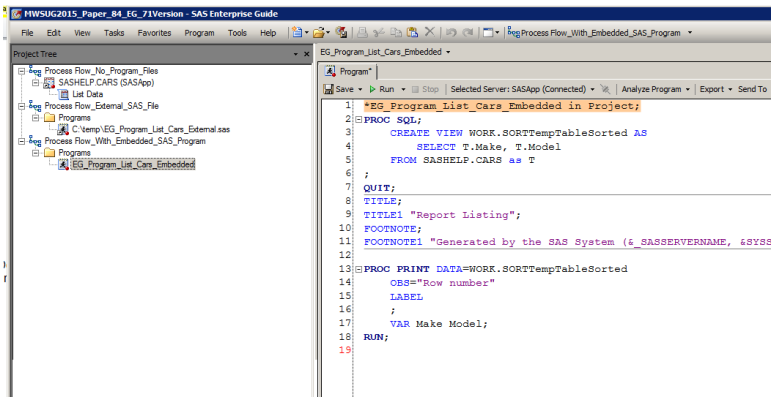
   a. Is there a "user-coded" SAS program file visible? Yes.

   b. Is there SAS code "behind the scenes under the hood"? Yes, but you don't have to worry about it. Obviously it contains the information for the process flows. But this code is not visible as SAS Code.

   c. Does the "MWSUG2015_Paper_84.egp" file need to be backed up, versioned etc. by the user? Yes, the new feature does not help do this.

   d. Does the external file "c:\temp\ EG_Program_List_Cars_External.sas" need to be backed up, versioned etc. by the user? Yes, the new feature does not help do this.

3. **SAS Enterprise Guide With an Embedded SAS Program file added**. A third Process Flow has been added (File – New – Process Flow) and renamed "Process_Flow_With_Embedded _SAS Program". This SAS Program is usually created internally in EG (File-New-Program)



   a. After creating the new empty program, click on Save (not save as – save as will steer you towards storing an external file)

   b. Rename the file from Program to the name shown in the figure "EG_Program_List_Cars_Embedded".

   c. Optionally, confirm the program is embedded. Right click on the file name and Select Properties General. Note that the file path shows (Embedded in Project). There is a figure later in this paper showing this information.

      Authors Opinion: There are other ways to get Embedded SAS files in a project, but this is the most logical path for a new user of Enterprise Guide to "stumble" through. Consider if you will: File New Program creates a Program Folder and then a file named Program. Performing another "File New Program " creates another Program (Program(1)) underneath the Program Folder. This can be extremely confusing with all of the things named "Program". It would be much better to have 3 choices pop up when a user selects "File New Program" --- (1) new embedded program, (2) new external program, (3) new link to existing external program.

   d. Now, a repeat of the questions above:

      i. Is there a "user-coded" SAS program file visible? Yes

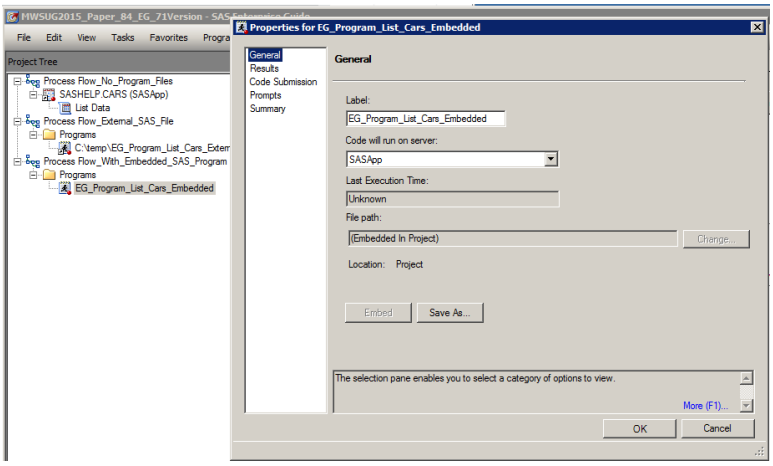
      ii. Is there SAS code "behind the scenes under the hood"? Yes, but you don't have to worry about it. Obviously it contains the information about the process flows. But it is not visible as SAS Code.

      iii. Does the MWSUG2015_Paper_84.egp file need to be backed up, versioned etc. by the user? Yes, the user must do this independently. The new feature does not help do this.

      iv. Does the file " EG_Program_List_Cars_Embeded" need to be backed up, versioned etc. by the user? Backup -- Sort of NO, backup is taken care of when the all-enclosing project file (EGP) is backed up which is user dependent. Versioning -- NO, versioning is available here within the project. VERSIONING EMBEDDED FILES IS WHAT THIS PAPER AND THE NEW 7.1 FEATURE ARE ALL ABOUT.

      v. An additional question, what if you mess up the EGP file or the embedded SAS file by deletion or some other catastrophic event. Will this versioning capability help you in restoring the egp file or the embedded SAS files? NO, you must have externally backed up EGP file so that it can be restored. If you lose the EGP file, you have also lost the embedded SAS program and its histories.

Some discussion on an embedded file. How did we create it? (1) "File-New_Program", (2) it was immediately renamed to EG_Program_List_Cars_Embedded. We then entered the code and hit the Save menu buttom. If we had hit "save as) it would have prompted for filename, path etc. leading us to an external file.

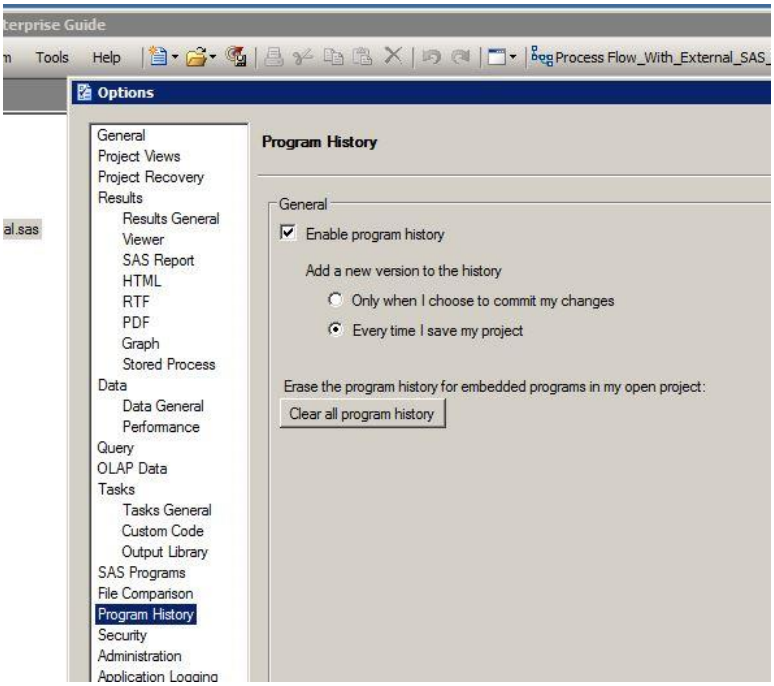


How do we confirm that the file is embedded?

1. Note the absence of a shortcut symbol on the icon in the Project Tree window. Linked files have shortcut symbol on the icon.
2. We can confirm by right-clicking on the file name and selecting Properties General
3. As shown, the file path says that it is "embedded in the project".

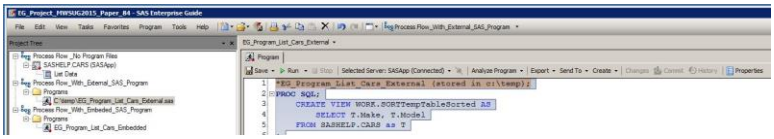
Activating the Versioning Capability

By default, the ability to create version may or may not be enabled. To enable it, select the menu item Tools – Options and "Enable Program History". It is suggested to enable the button to save "every time I save my project". Note that this is every time the project is saved, not the file is saved. Files are typically saved more often than projects.



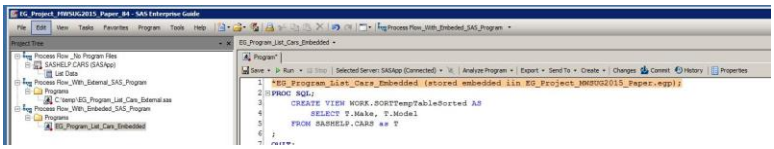
Using the Versioning Menu Tabs

Let's start by looking at an external SAS program file in the program editor. Note the menu tabs for "Changes", "Commit" and "History" are greyed out. These tools



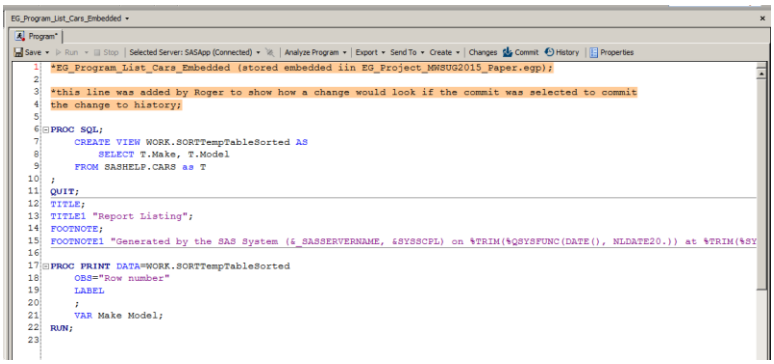
do not work for external SAS program files.

Let's look at an embedded program. Look above the program window. Another way to confirm that the program is embeded is to look at the tabs at the top of the program screen..-- "Changes", "Commit" and "History" are no longer greyed out, but rather are active and available for use.

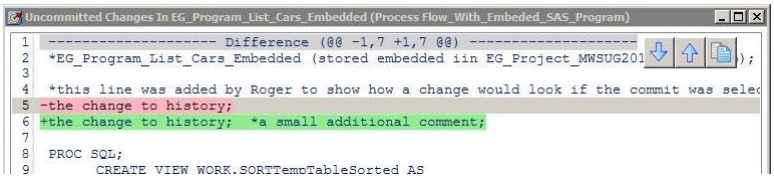


Start by clicking on the "Changes" menu tab. A window comes up that says no changes have been made since the last version commitment. We have never made a change to the file.

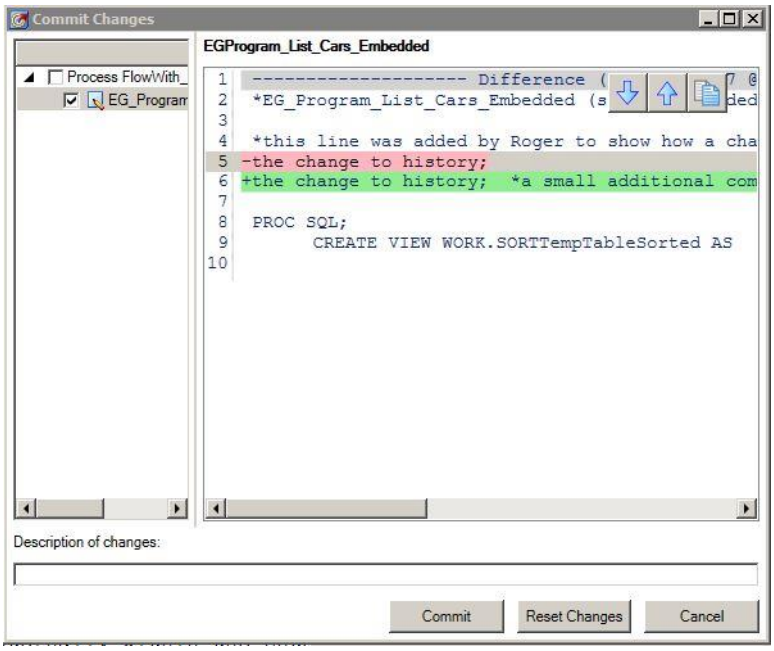
So, let's add some additional comments to make a change to the embedded SAS File. There are now 2 comments at the top of the file. Do not save the file.



Click on the "Changes" tab. A window pops up showing changes that are uncommitted to the versioning history. In this case the line in red is the previous version of the line and the line in green reflects what has been changed but is not yet committed to the versioning history.

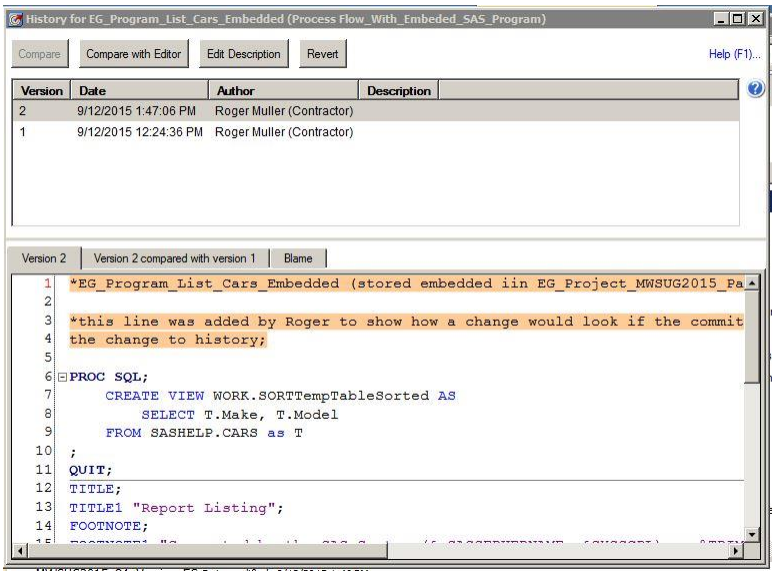


Close this window then hit the “Commit” menu tab above the program editor. This will bring up a somewhat redundant window to what we just saw, but it has a “Commit” button at the bottom. Go ahead and hit the Commit button and the changes to the program will be committed to the program’s versioning history.



.

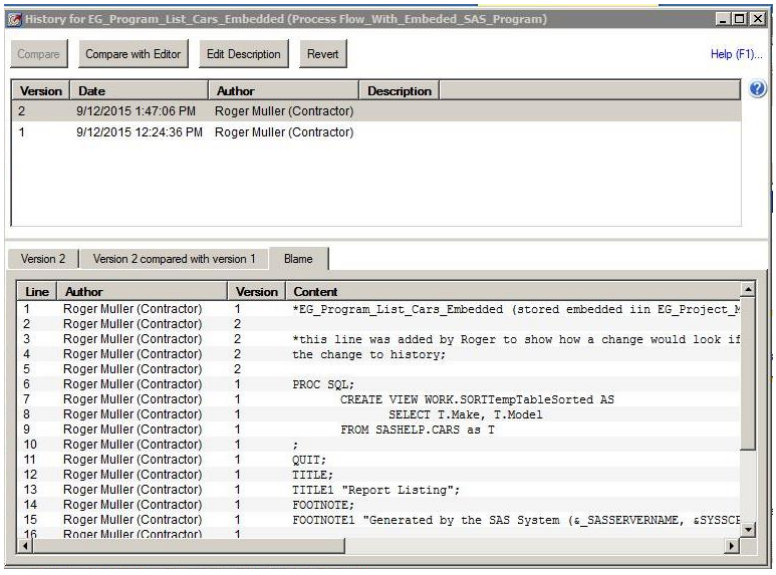
Now click on the “History” menu tab. As shown, we have 2 version histories. We can look at either by selecting it in the upper part of the window. There is a button to make “compare” differences of version 2 vs. version 1. There is a button to “Edit Description” that appears in this window. More will be said about the “Compare With Editor” and “Blame” Buttons” a couple of steps later in this document.



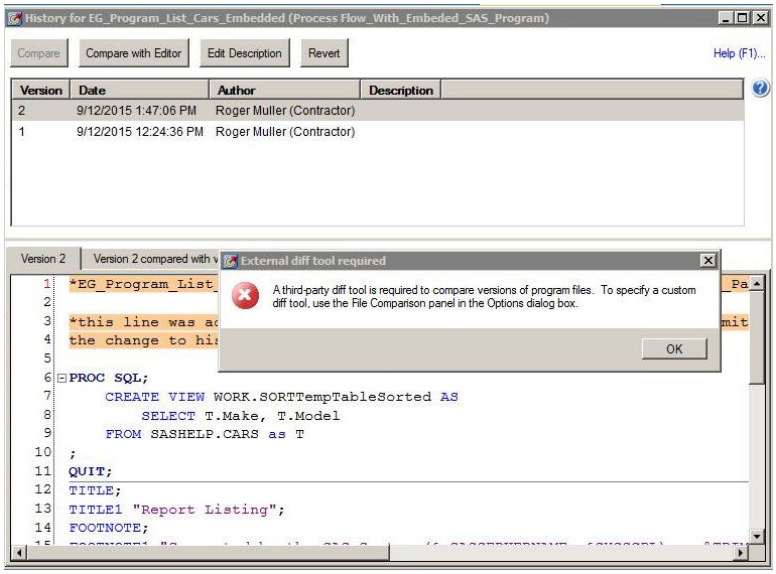
The “Revert” button allows the file to be reverted to a different version. The “Revert” button at the top screen clearly gives the capability of rolling back code. Needless to say, know what you are doing before you get into this! It would be a very good idea to make a backup copy of the embedded file to have it as well as the version history at hand should you make a mistake during the reversion process.

To roll back to a previous version in its entirety, select and hit revert. The code in the editor will then rollback. Obviously this screen becomes very busy if there are multiple rollback versions.

The “Blame” button shows which user made a change to each line in the file. Author’s note: the name assigned to this button is the first time in history a humorous item has been inserted on a SAS menu or button!



Lastly, the “Compare with Editor” button gives a comparison of (1) whatever is in the program editor that has yet to be saved or committed to (2) a previous version that has been committed. This could be very useful if one is not completely sure of what they have just changed and wants to compare this to the most recent committed version before actually making the commitment or doing a file save. Note the requirement for a third party difference tool. This requirement will greatly restrict convenient use of this feature in many tightly-controlled SAS environments.



A Summary of This Section

This process can go on and on with many more versions of the embedded file being revised and recommitted. For demonstration purposes, this simple process with two version shows the capabilities of the feature. In real usage, these revision histories could become so complex with many versionsthat they might be almost unusable. There is a provision to reset the history and start over. This should be used at significant file milestones, such as when it is “frozen” to go into production. Likewise if there are significant changes in development (eg from version 0.1 to 0.2), do not keep all of the old version information from the development process in the code.

Above all else, the programmer must be extremely careful is in restoring versions of files. It would be very easy to revert too far back and lose valuable content. A better approach might be to find the code you are looking for in a previous version and use Windows cut and paste to bring it back into the current edited file. The bottom line on this technique is that you do not need to do a complete file restoration to retrieve pieces of code. Make sure you are in control and know what you are doing.

CONCLUSION

The ability to add version control to embedded SAS program files contained within Enterprise Guide project files is a welcome addition to SAS Enterprise Guide 7.1. Hopefully future releases will have tools to version the project file itself and to version external SAS files.

REFERENCES

- Hemendinger, Chris, 2014. New SAS programming features in SAS Enterprise Guide 7.1. http://blogs.sas.com/content/sasdummy/2014/10/12/eg-71-new-programmer-features/
- Hemendinger, Chris, 2012. Using source control management with SAS Enterprise Guide. http://blogs.sas.com/content/sasdummy/2012/10/29/scm-with-sas-eg/
- Hemedinger, Chris, 2013. Re: How to search SAS Enterprise Guide files. https://communities.sas.com/t5/SAS-Enterprise-Guide/How-to-search-SAS-Enterprise-Guide-files/m-p/86656#M8218
- Muller, Roger D., 2014. File Management and Backup Considerations When Using SAS® Enterprise Guide (EG) Software. http://www.mwsug.org/proceedings/2014/RF/MWSUG-2014-RF05.pdf
- Slaughter Susan J. and Lora D. Delwiche. Writing SAS Code in SAS Enterprise Guide. http://www2.sas.com/proceedings/forum2008/184-2008.pdf
- Flynn, Joe, Casey Smith, Alex Song. Check It Out! Versioning in SAS Enterprise Guide. http://support.sas.com/resources/papers/proceedings14/SAS179-2014.pdf
- Smith, Casey. Find What You Are Looking For and More in SAS Enterprise Guide. http://support.sas.com/resources/papers/proceedings15/SAS1924-2015.pdf

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Roger D. Muller, Ph.D.
Enterprise: Data-To-Events, Inc.
Address: 14475 Stephanie St.

City, State ZIP: Carmel, IN USA
Work Phone: 317/846-5782
E-mail: rdmuller@hotmail.com
Web: www.data-to-events.com, www.rogermullervideography.com