

# Version Control in SAS Enterprise Guide 7.1



Shahriar Khosravi  
Senior Analyst, Risk Management  
Risk Capital and Model Development

---

Business Initiatives  
Winter 2018

# Introduction

## Main Objectives

- Provide a brief review of version control and tracking for code development
- Showcase the version control capabilities available in SAS Enterprise Guide (EG) 7.12

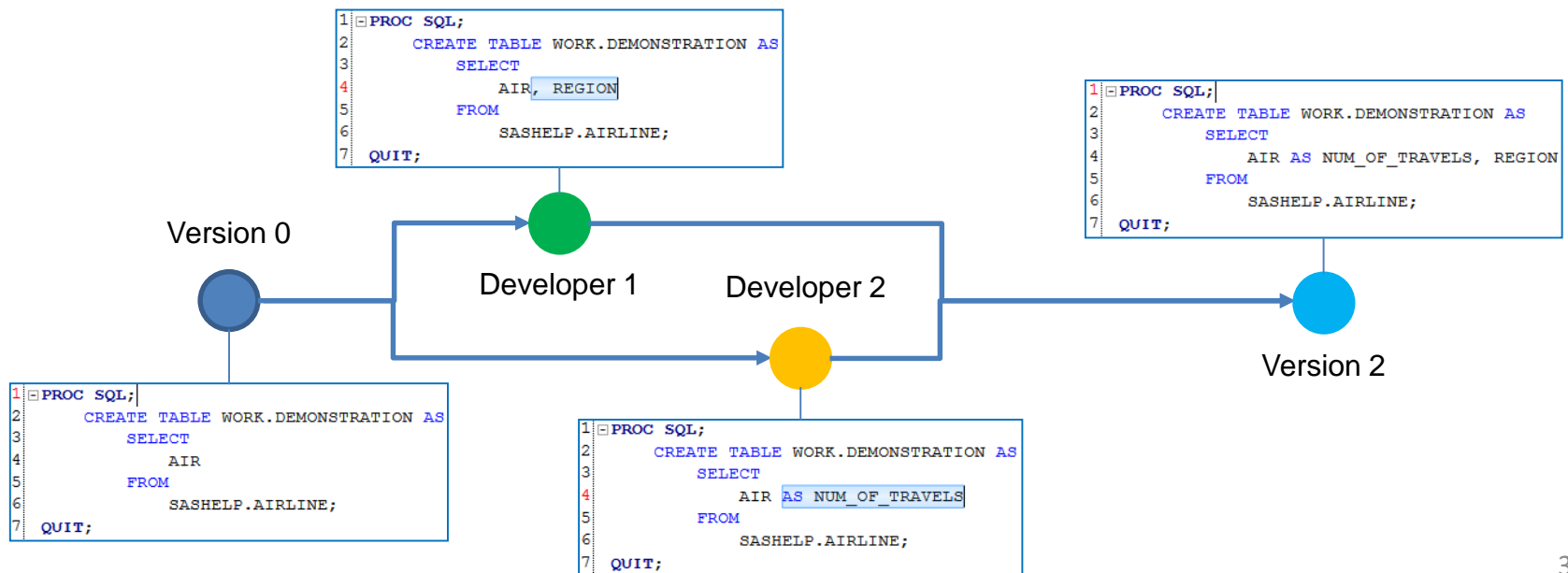
## Background

- BMO is upgrading EG to version 7.12
- The version control functionality is a new feature in release 7.12

# Background

## Version Control

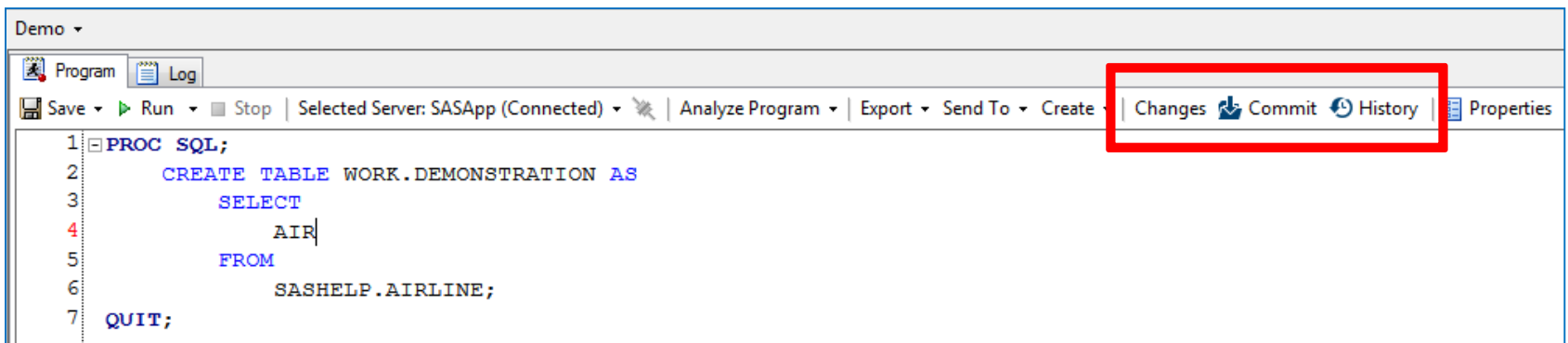
- Allows efficient and organized tracking of code changes throughout the development process
- Facilitates collaboration between multiple developers working on the same project



# Version Control System in SAS EG

## Git in SAS EG 7.12

- SAS EG includes a simplified derivative of the version control system 'Git'
- The version control options are available when editing a program



The screenshot shows the SAS EG interface with a program editor window. The menu bar includes 'Save', 'Run', 'Stop', 'Selected Server: SASApp (Connected)', 'Analyze Program', 'Export', 'Send To', 'Create', 'Changes', 'Commit', 'History', and 'Properties'. The 'Changes', 'Commit', and 'History' options are highlighted with a red box. The program editor contains the following code:

```
1 PROC SQL;  
2     CREATE TABLE WORK.DEMONSTRATION AS  
3     SELECT  
4         AIR  
5     FROM  
6         SASHELP.AIRLINE;  
7 QUIT;
```

# How to Use Git in SAS EG (1 of 6)

## Committing Code Changes

- In order to start keeping track of changes to programs in SAS EG, first save all programs in the project
- Make the initial commit by clicking on 'Commit'
- Every time the code is committed to the repository, a snapshot of the changes (if any) is saved
- This allows for maintaining a history of all changes to the programs in EG



# How to Use Git in SAS EG (2 of 6)

## Committing Code Changes

- Once the user invokes the 'Commit' option, a new window appears
- This window allows the user to choose which programs to commit to the repository at this stage, and to include a description of changes for future reference
- If the current commit is not the first one, then Git will display a snapshot of changes that are about to be committed



# How to Use Git in SAS EG (3 of 6)

The screenshot shows the 'Commit Changes' dialog box in SAS EG. The left pane displays a file tree with the following items:

- ModelValidation
- + Demo
- + Run\_Model
- + Test\_Model\_Output

The main pane displays the following SAS code:

```
1 PROC SQL;  
2     CREATE TABLE WORK.DEMONSTRATION AS  
3     SELECT  
4     AIR  
5     FROM  
6     SASHELP.AIRLINE;  
7 QUIT;
```

At the bottom, the 'Description of changes:' text area contains the text: 'Making the initial commit'.

Two blue callout boxes provide instructions:

- 'Select the programs you would like to commit' points to the file tree.
- 'Include a comment for future reference' points to the commit message text area.

Buttons at the bottom right include 'Commit', 'Reset Changes', and 'Cancel'.

# How to Use Git in SAS EG (4 of 6)

Commit Changes

ModelValidation

Demo

Previous (Red)

New (Green)

Demo

```
1 ----- Difference (@@ -1,7 +1,7 @@) -----
2 PROC SQL;
3     CREATE TABLE WORK.DEMONSTRATION AS
4     SELECT
5 -     AIR
6 +     AIR AS NUM_OF_TRAVELS
7
8     FROM
9     SASHELP.AIRLINE;
10 QUIT;
```

User can also discard all changes since the last commit

Description of changes:

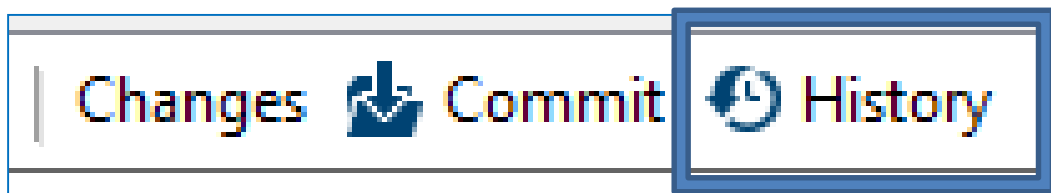
Commit Reset Changes Cancel



# How to Use Git in SAS EG (5 of 6)

## Reviewing History

- When the user selects 'History', a new window appears
- This window allows the user to see a history of changes made to the current program, compare different versions, find who made changes to which line of the code through the 'Blame' option, and 'Revert' back to a previous commit



# How to Use Git in SAS EG (6 of 6)

The screenshot shows the 'History for Demo (Model\_Validation)' window in SAS EG. The window has a toolbar with buttons for 'Compare', 'Compare with Editor', 'Edit Description', and 'Revert'. The 'Revert' button is highlighted with a red box and a blue callout box that says 'Revert to a previous commit'. Below the toolbar is a table of commit history:

Version	Date	Author	Description
3	28/07/2017 11:57:35 AM	Shahriar Khosravi	Adding 'REGION' to the table 'DEMONSTRATION'
2	28/07/2017 11:57:02 AM	Shahriar Khosravi	'AIR' will appear as 'NUM_OF_TRAVELS' to indicate the number of travels
1	28/07/2017 11:50:35 AM	Shahriar Khosravi	Making the initial commit

Below the table is a 'Blame' view for 'Version 3'. The 'Blame' button is highlighted with a red box and a blue callout box that says 'Find out who changed which line of code'. The 'Changes made in version 3' button is also highlighted with a red box and a blue callout box that says 'Review changes'. The code editor shows the following SQL code with a difference view:

```
1 ----- Difference (@@ -1,7 +1,7 @@) -----
2 PROC SQL;
3     CREATE TABLE WORK.DEMONSTRATION AS
4     SELECT
5 -     AIR AS NUM_OF_TRAVELS
6 +     AIR AS NUM_OF_TRAVELS, REGION
7     FROM
8     SASHELP.AIRLINE;
9 QUIT;
10 \ No newline at end of file
11
```

# Summary

- Version control is an essential process in any code development process, especially ones that involve multiple developers
- SAS EG 7.12 provides a simple and intuitive tool for version control of programs contained within an EG project
- The simplified Git version control system allows users to
  - Maintain a history of changes made to programs
  - Discard changes since the last commit
  - Revert back to a previous commit
  - Collaborate with other developers in an organized and efficient way

# Extracting the Git Repository from a SAS Enterprise Guide 7.1 Project

Shahriar Khosravi  
Senior Analyst, Risk Management  
Risk Capital and Model Development

---

Business Initiatives  
Winter 2018

# Introduction

## Question from SAS Support

- Is it possible to extract the Git repository from a SAS EG project?

## SAS Response



Edmund,

Currently there is not a way to extract the embedded program history in SAS Enterprise Guide. The SAS Enterprise Guide R&D manager believes this is a good suggestion and I will be opening a formal suggestion for this feature to be added to a future release of SAS Enterprise Guide.

Sorry to report that this is not possible currently.

Tiffany

# Introduction

## Main Objectives

- Demonstrate how to extract the Git repository from an EG project
- Show how to view the Git history of externally version-controlled SAS scripts in Enterprise Guide (EG) 7.12

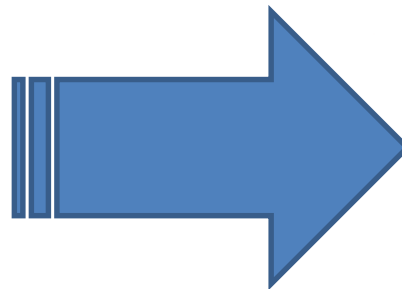
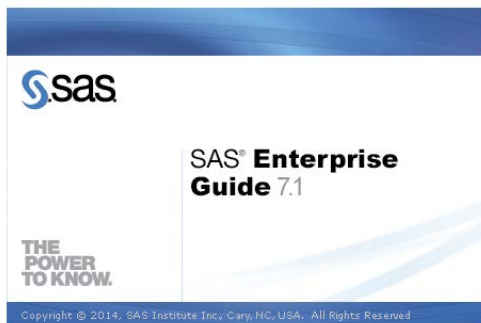
## Background

- SAS EG's version control capabilities are limited
- In some cases, it may be necessary to extract the Git repository from SAS EG and continue the version tracking outside of EG while preserving the existing history

# Background

## Limitations of Version Control in SAS EG

- Multiple developers cannot contribute concurrently to the same project
- Not possible to create and maintain a 'central repository' that multiple developers could interact with simultaneously
- No functionality for working on separate 'branches' within the same project



# Extracting the Git Repository from SAS EG

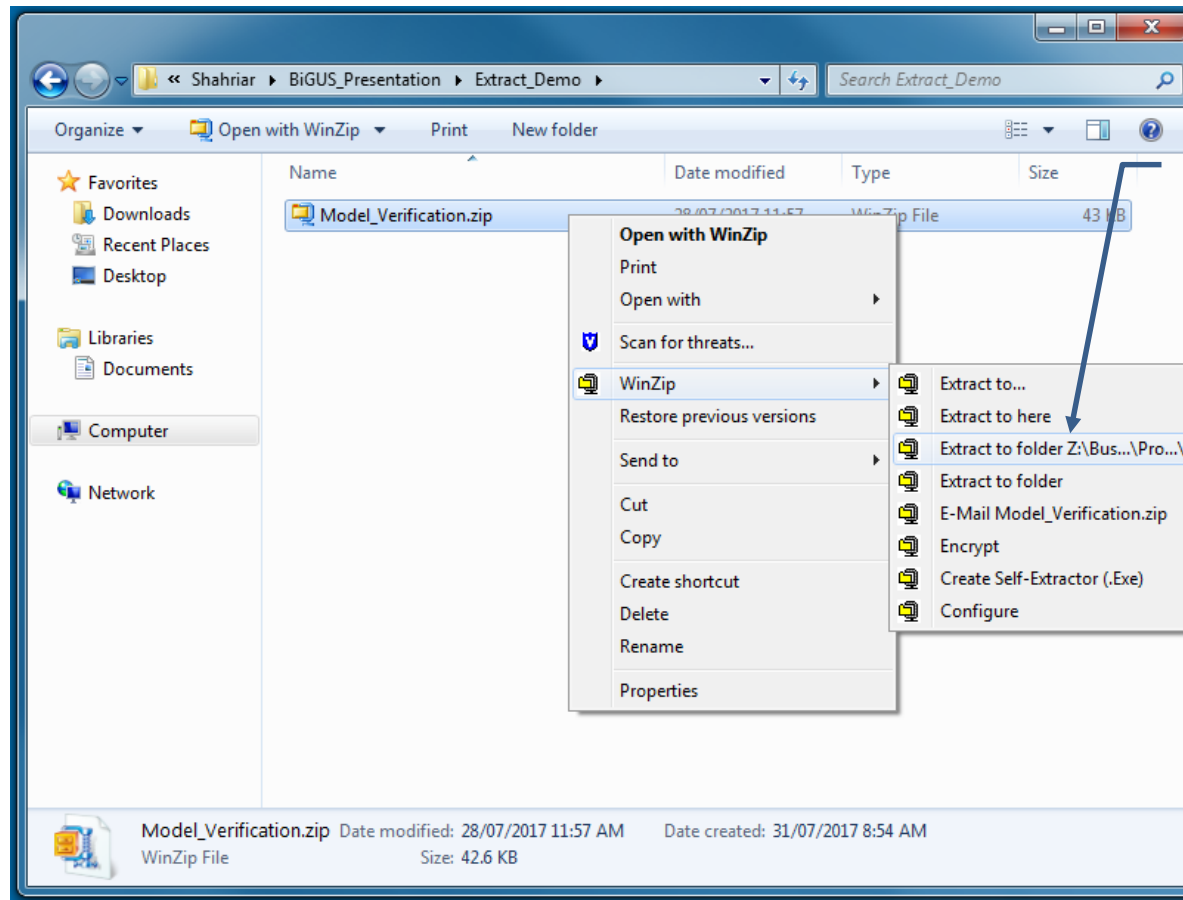
The screenshot shows a Windows File Explorer window with the address bar set to « Shahriar » BiGUS\_Presentation » Extract\_Demo. The file list contains one item: Model\_Verification.egg, dated 28/07/2017 11:57, with a size of 43 KB. A context menu is open over this file, listing options such as Open, Scan for threats..., WinZip, Restore previous versions, Send to, Cut, Copy, Create shortcut, Delete, Rename, and Properties. The 'Rename' option is highlighted. A blue callout box on the left contains the text: "Select 'Rename' to replace '.egg' with '.zip'". An arrow points from this box to the 'Rename' option in the context menu.

Name	Date modified	Type	Size
Model_Verification.egg	28/07/2017 11:57 ...	SAS Enterprise Gui...	43 KB

Model\_Verification.egg Date modified: 28/07/2017 11:57 AM Date created: 31/07/2017 8:54 AM  
SAS Enterprise Guide Project Size: 42.6 KB

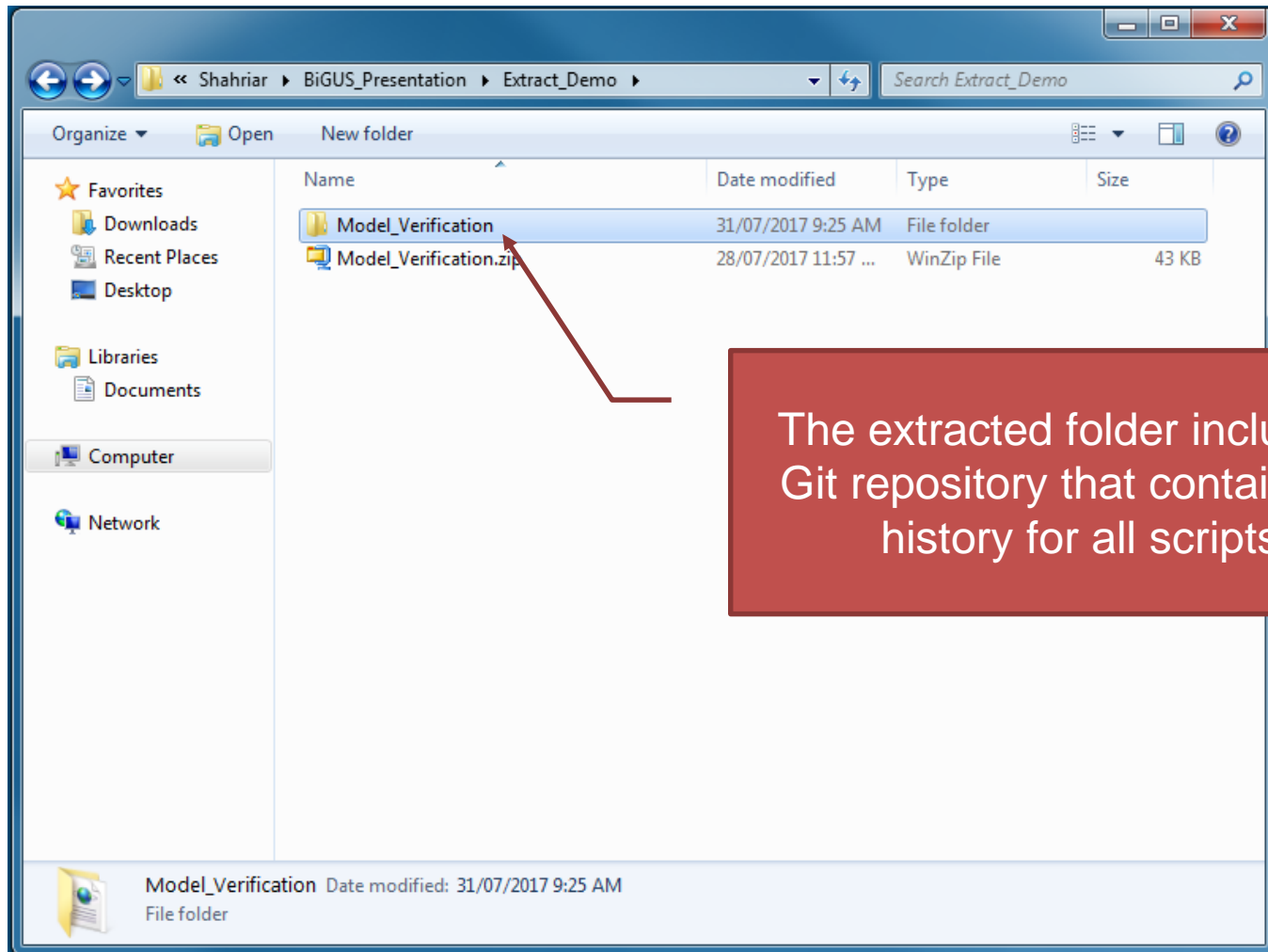


# Extracting the Git Repository from SAS EG



Extract the contents of the compressed folder

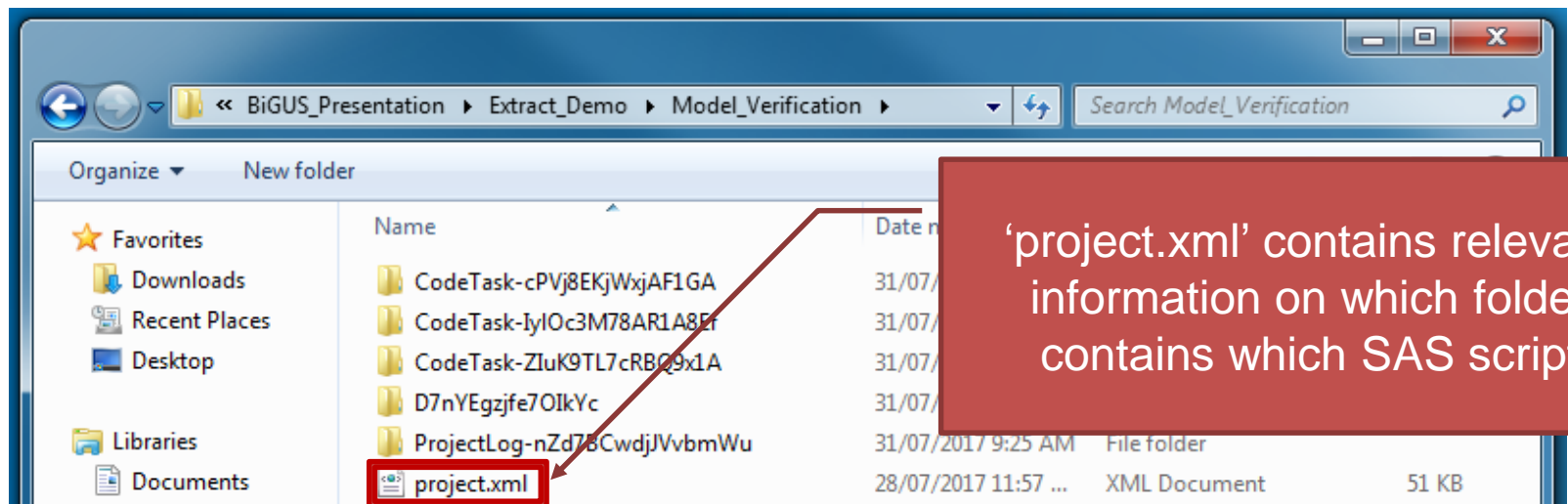
# Extracting the Git Repository from SAS EG



# The Project Folder

## A Look Inside

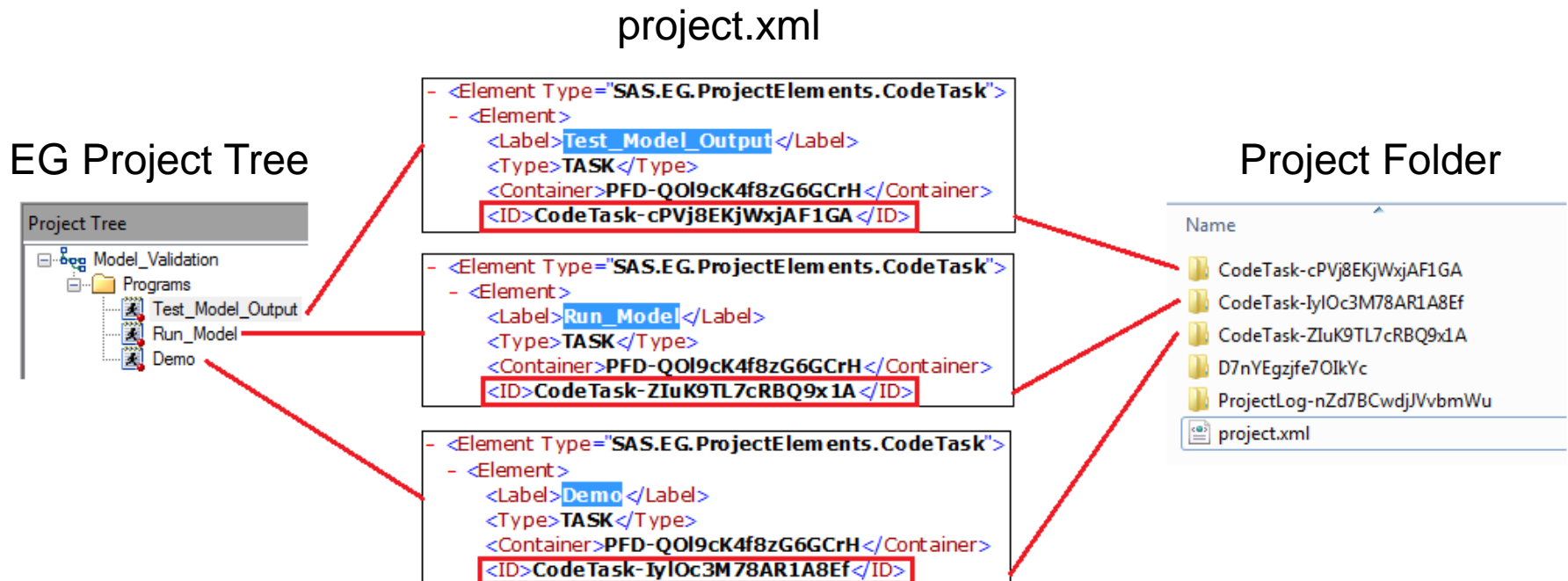
- Every script (or code) and its history is contained within a separate folder whose name starts with 'CodeTask-'
- The XML file 'project.xml' holds the mapping information for all scripts



# The Project Folder

## project.xml

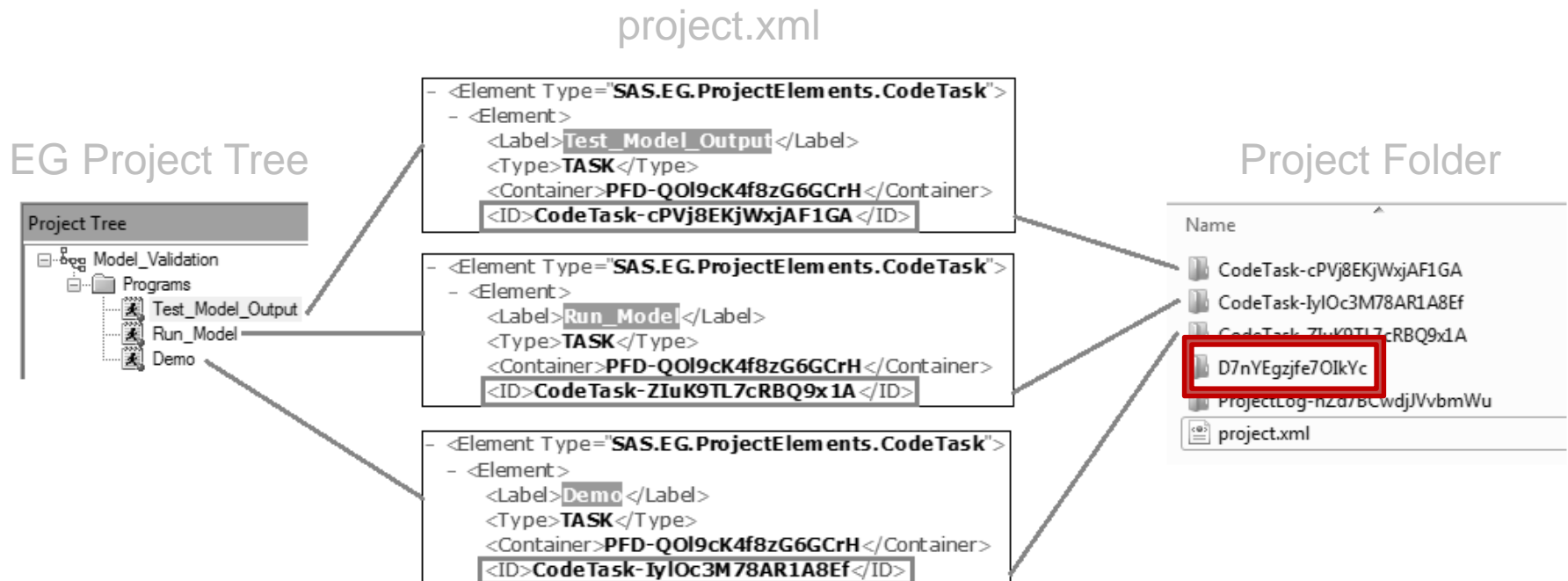
- User can identify the folders based on the information inside the 'project.xml' file



# Identifying the Git Folder

## The Git Folder

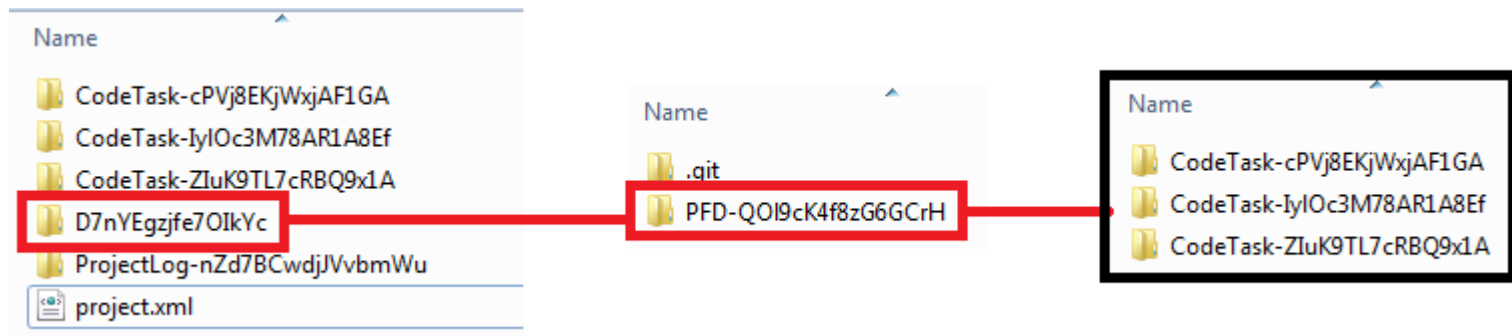
- The remaining folder is the one that contains the Git repository



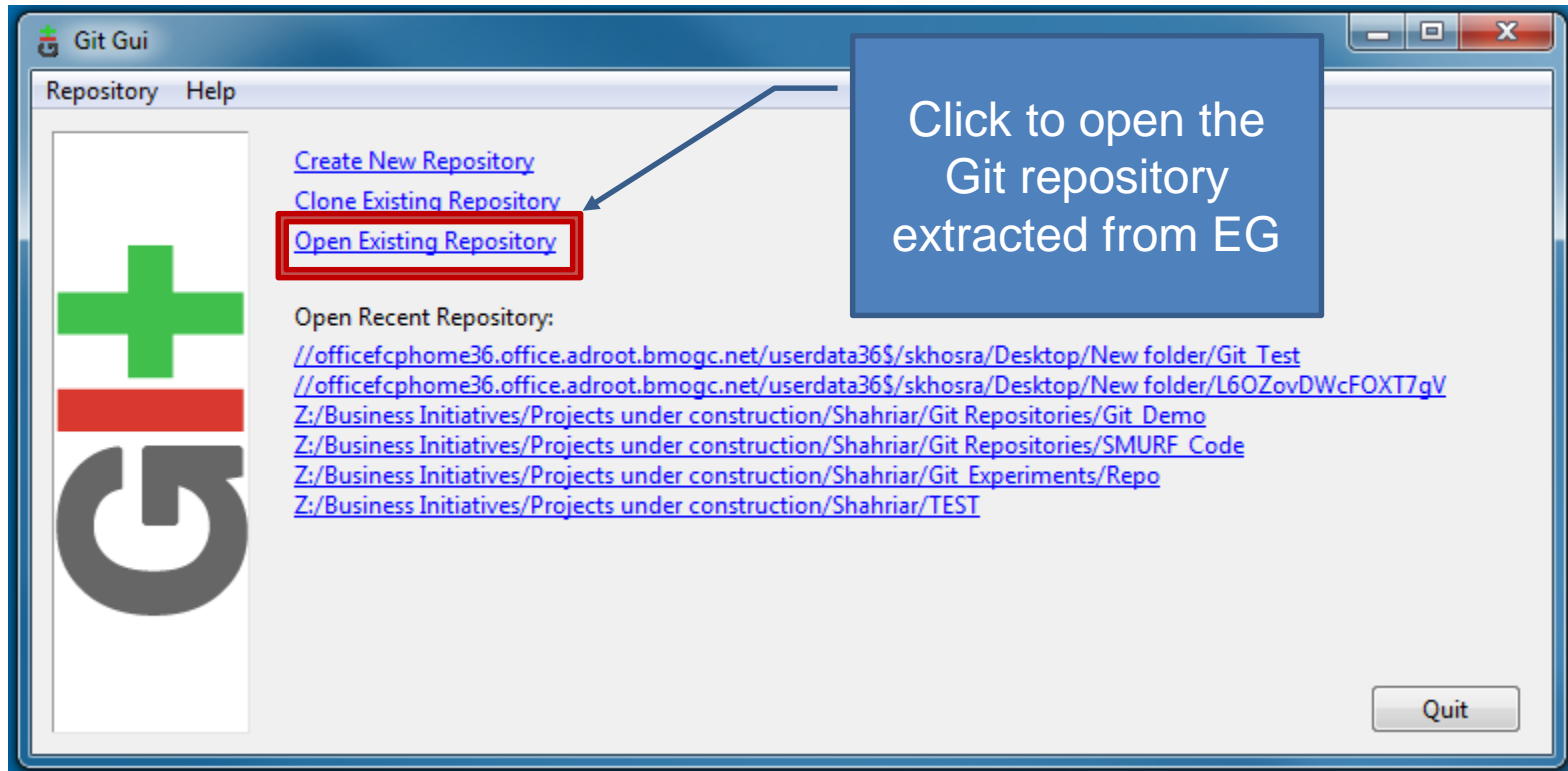
# Renaming the Git Folders

## Git Folder Contents

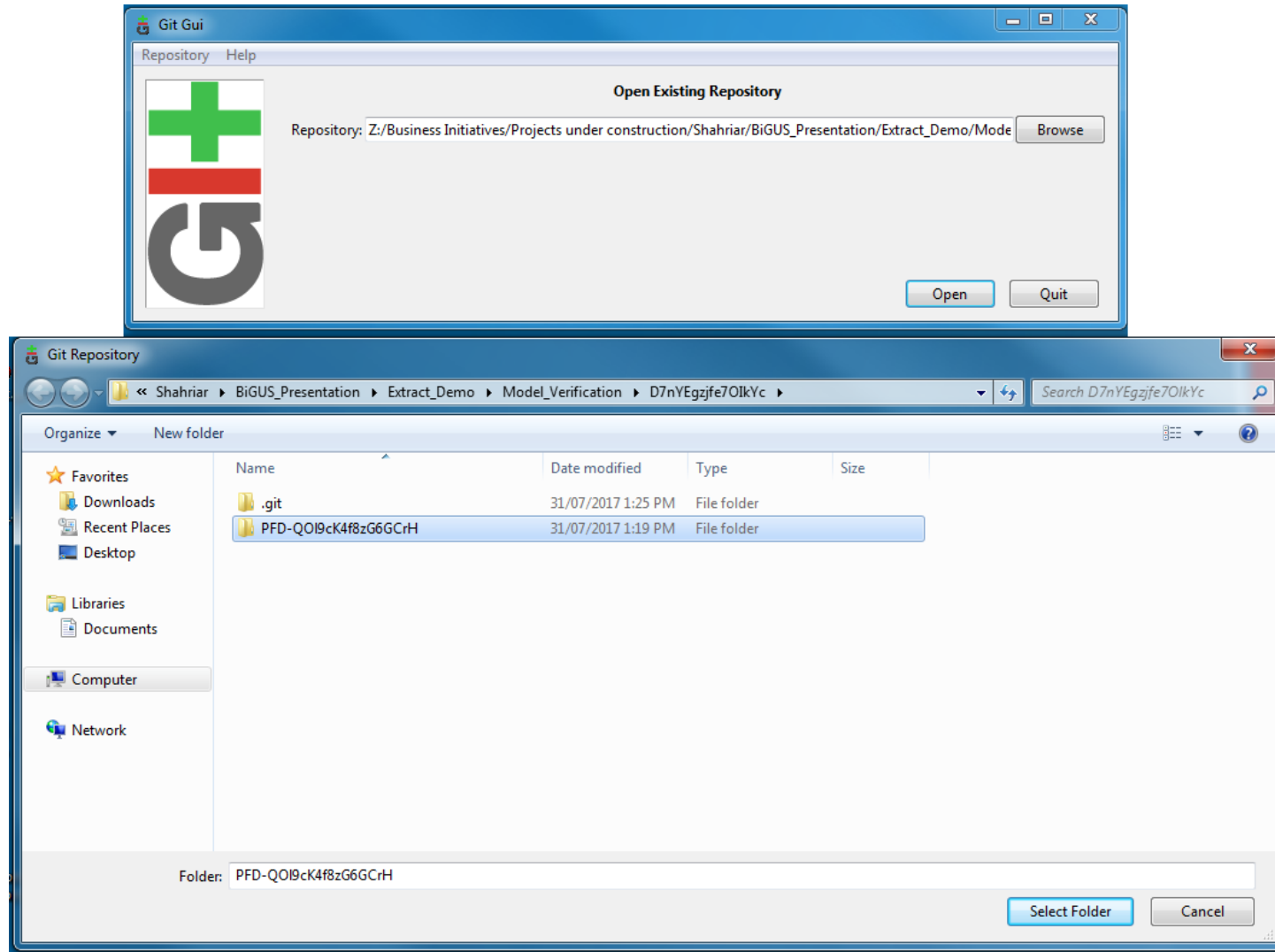
- User can rename the Git folder and its contents for ease of use and understanding
- This can be done in the Git version control system in order to keep the original histories, and the process can be automated



# Renaming the Git Folders (Using Git GUI)



# Renaming the Git Folders







# Viewing the Extracted Git History

The screenshot shows the Git GUI interface. At the top, the window title is "D7nYEgzfe7OIkYc: master - gitk". The menu bar includes "File", "Edit", "View", and "Help".

The main area is divided into two panes. The left pane shows the commit history with a yellow dot for the current commit (SHA1: 7785a5d2162fe17f416ec41868cce2ceb4952869) and two blue dots for previous commits. The commit messages are: "Renaming folders and files", "'AIR' will appear as 'NUM\_OF\_TRAVELS' to indicate the number of travels", and "Making the initial commit". The right pane shows the commit log with columns for author, email, and date. The authors listed are "Shahriar Khosravi <Shahriar.Khosravi@bmo.com>", "Shahriar Khosravi <sas@sas.com>", and "Shahriar Khosravi <sas@sas.com>".

Below the panes, there is a search bar with "SHA1 ID:" and "Find commit containing:". The search criteria is set to "Exact" and "All fields".

The bottom section shows the diff view for the selected commit. It includes the commit message "Renaming folders and files" and the diff output for three files:

```
----- Scripts/Demo.sas -----
similarity index 100%
rename from PFD-Q019cK4f8zG6GCrH/CodeTask-Iy1Oc3M78AR1A8Ef/code.sas
rename to Scripts/Demo.sas

----- Scripts/Run_Model.sas -----
similarity index 100%
rename from PFD-Q019cK4f8zG6GCrH/CodeTask-ZIuK9TL7cRBQ9x1A/code.sas
rename to Scripts/Run_Model.sas

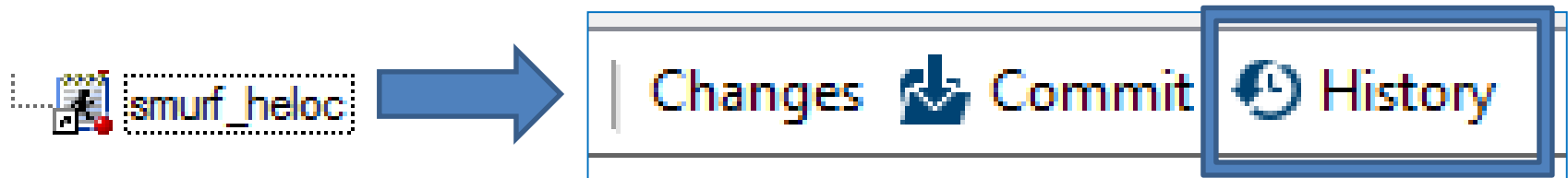
----- Scripts/Test_Mode_Output.sas -----
similarity index 100%
rename from PFD-Q019cK4f8zG6GCrH/CodeTask-ZIuK9TL7cRBQ9x1A/code.sas
rename to Scripts/Test_Mode_Output.sas
```

On the right side, there is a "Comments" pane showing a list of file paths: "PFD-Q019cK4f8zG6GCrH/CodeTask-Iy1Oc3M78AR1A8Ef/code.sas", "PFD-Q019cK4f8zG6GCrH/CodeTask-ZIuK9TL7cRBQ9x1A/code.sas", "PFD-Q019cK4f8zG6GCrH/CodeTask-CPVj8EKjWxjAF1GA/code.sas", "Scripts/Demo.sas", "Scripts/Run\_Model.sas", and "Scripts/Test\_Mode\_Output.sas".

# How to View an Existing Git History in EG?

## Externally-Controlled Files and EG

- It is possible to view the Git history of an externally managed script within SAS EG
  - drag and drop the external script file into the SAS EG
  - Click 'History' from the top bar menu



# How to View an Existing Git History in EG?

History for smurf\_heloc (Model\_Validation) in branch master

Compare Compare with Editor Revert Help (F1)...

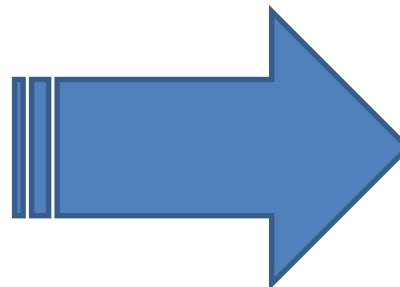
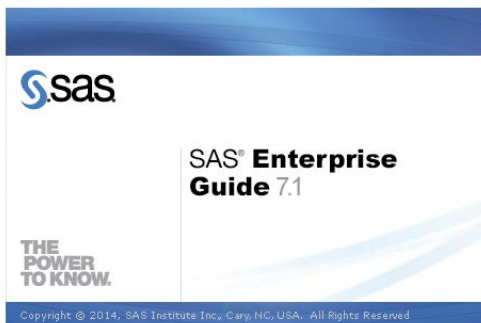
Version	Date	Author	Description
196a0ab	31/03/2017 1:13:23 PM	Shahriar Khosravi	Replaced the use of PROC FCMP with PROC FORMAT for the utilization model based on Preethi and Mahmoud's investigation into missing values.
64bdd31	29/03/2017 3:20:56 PM	Shahriar Khosravi	Changed "lgd" to "lgd_monthly" so that the "quarterly_lgd" can successfully be renamed to "lgd" when the treated table is being created. This reverts commit 8152bde.
8152bde	23/03/2017 9:20:03 AM	Shahriar Khosravi	Made changes to the US HELOC SMURF code in order for it to be implemented in BMOFG.
edce51f	13/04/2017 11:56:39 AM	Shahriar Khosravi	Revert "As per Mahmoud's instructions, I have made the annualized_pd equal to pd_pct"
f70b847	10/04/2017 3:48:30 PM	Shahriar Khosravi	As per Mahmoud's instructions, I have made the annualized_pd equal to pd_pct for the US HELOC and Mortgage SMURFs in order to correct the calculation.
a70ec58	07/04/2017 1:48:22 PM	Shahriar Khosravi	FRG Dev team have made changes to BMOFG SMURFs as part of the revised RWA calculations.
7a96991	06/04/2017 8:53:16 AM	Shahriar Khosravi	After a failed Master run, Chad fixed the US HELOC SMURF.

Version 196a0ab Changes made in version 196a0ab Blame

```
1 /*-----*/
2 * NAME: smurf_heloc.sas
3 *
4 * PURPOSE: Produces all BFC HELOC output (PD/LGD/Utilization/prepayment)
5 *
6 * NOTES:
7 *
8 * MODIFIED:
9 * 26AUG2015 - CM - Creation
10 * 17SEP2015 - CM - Updated coefficients and lookups
11 * 22SEP2015 - MS - fixed syntax issues
12 * 26SEP2015 - MS - added validvarname options
13 * 30SEP2015 - MS - changed delinquency logic to use reportedloanstatus
14 * 09OCT2015 - MS - included senior balance in outstanding for LGD calculation
15 * 14OCT2015 - MS - Prepared for promotion into SIT with issues detected during initial testing
16 * 16OCT2015 - MS - Initial switch to BALANCE_AMT pending undrawn treatment
17 * 20OCT2015 - MS - Enriched to create dummy drawn and undrawn records when needed
18 * 29OCT2015 - MS - Added coalesce to determine updated_cur_bal_amt when balance_amt was missing
19 * 03NOV2015 - CK/KG -Added regional macroeconomic variable lookup
20 * 11NOV2015 - CM - Addressed defects 1544 and 1546
21 * 11NOV2015 - MS - Modified LGD outstanding logic
22 * 11NOV2015 - MS - Modified LGD outstanding logic
```

# Summary

- It is possible to extract the Git repository from a SAS EG project
- Some parts of the process can be automated using bash scripting
- It is also possible to view the existing history of an externally-controlled file inside SAS EG



# Questions

Thank You