SAS® GLOBAL FORUM 2020 #SASGF

SAS5194-2020

# Git it? Got it? Good: Migrating from SAS® to SAS® Viya® Using Git

Cameron Estroff and Weston McManus, SAS Institute Inc.

## ABSTRACT

You have established your workflows on SAS®9 but are excited about the possibilities of moving to SAS® Viya®. What does it look like to do SAS® in-memory, and in a clientless world? And what are you going to do with your current code? Let us help you get there with some ideas about how to migrate your code using the power of Git!

## INTRODUCTION

This paper makes a simple assumption: For years, you and your associates have used Base SAS® or SAS® Enterprise Guide® to create your fair share of SAS programs. These programs create models, manipulate data, run ad hoc processes, create reports, and perform tasks that include a complexity well beyond a simple %put ‘Hello World’;.

Given the amount of investment that has been built into your SAS programs, there might be some apprehension involved when hearing the words SAS® Viya®. Although there are lots of reasons to upgrade (including additional SAS Viya improvement to your original code), this paper is not focused on the benefits of the *new* but on the continuity of the *working* that you depend on. Will these same programs from SAS®9 continue to operate as originally scripted on SAS Viya?

Although this paper does not exhaustively iterate through the types of programs that might have been written, it does aim to address one of the more common scenarios:

How can a SAS programmer easily migrate their SAS code to SAS Viya?

The answer to that is simple: Git. SAS has established Git as the preferred code versioning control tool. As such, the main integrated development environments (IDEs) from SAS (SAS Enterprise Guide and SAS® Studio for SAS®9 and SAS Studio on SAS Viya) provide tight integration with Git. This paper outlines the simple process of using Git to allow for a seamless transfer between SAS®9 and SAS Viya.

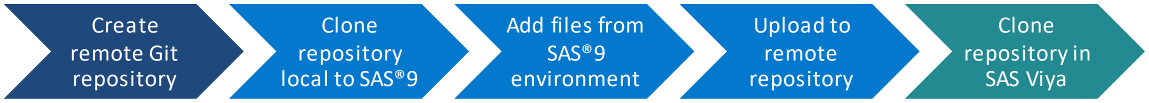

Create remote Git repository
Clone repository local to SAS®9
Add files from SAS®9 environment
Upload to remote repository
Clone repository in SAS Viya

**Figure 1: Code Migration Process Outlined in This Paper**

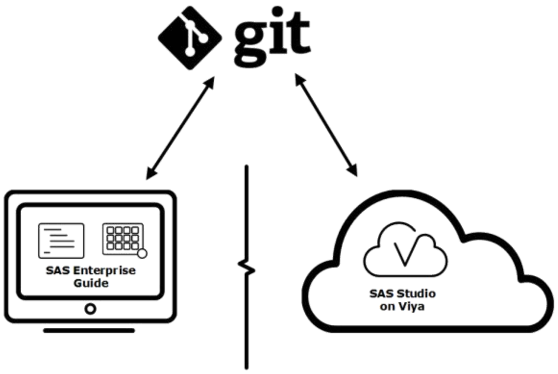


**Figure 2: Migrate from SAS Enterprise Guide to SAS Studio Using Git**

## FIRST THINGS FIRST

### WHAT IS GIT?

Git is a code version control system for tracking changes in files and coordinating work on those files among multiple people. Unlike CVS or SVN, there is no need for a dedicated "main" repository. All contributors will have copies of the entire repository locally.

### WHAT IS AN ONLINE GIT REPOSITORY HOST?

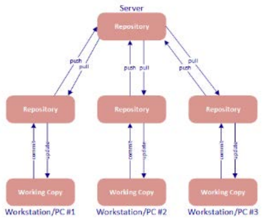


**Figure 3: Remote Git Repository Design**

An online Git repository host (OGRH) is essentially a clone of your Git repository on a remote server that is also running Git. This clone enables Git to behave more like a traditional CVS or SVN. Some of the more common online Git repository hosts (OGRHs) include GitHub, GitLab, and GitBucket.[1] Some OGRHs offer internal enterprise hosting that would be perfect for migration between SAS and SAS Viya.

### WHY USE GIT?

If you have not established code versioning as a part of your development work cycle, now is as good a time as any. SAS has established Git as the preferred code versioning control tool. Using Git provides source control, code collaboration, and continuous integration and deployment to your SAS processes.

### WHAT IS SAS STUDIO?

SAS Studio is functionally the online equivalent of SAS Enterprise Guide. Because SAS Studio is accessed through the web instead of your desktop, it maintains the ability to access the full capability set of SAS Viya. If you have used SAS Enterprise Guide 8 and later, the SAS Studio interface should look and feel familiar.

[1] A more complete list of the most popular OGRHs can be found at https://git.wiki.kernel.org/index.php/GitHosting.

### CAN I NOT USE SAS ENTERPRISE GUIDE WITH SAS VIYA?

For help with easing the transition to SAS Viya, SAS Enterprise Guide 7.15 and later is fully capable of communicating with SAS Viya servers, connecting to the SAS Viya workspace server like you would a SAS®9 workspace server, and enabling you to submit SAS code. However, if your program files are stored in a location that cannot be properly accessed by SAS Viya (such as your PC), a file migration is still necessary to fully integrate with the clientless architecture SAS Viya provides.

### WHAT IF I STORED ALL OF MY CODE INSIDE SAS ENTERPRISE GUIDE PROJECTS?

There are two migration strategies for code inside Enterprise Guide: The first is to leverage the migration exercise to get code out of Enterprise Guide projects (as a best practice in general). The second strategy is to wait to leverage the capabilities of SAS Studio to import .egp files, *when this functionality becomes available*. This work is on the SAS Studio roadmap to address during 2020.

### WHEN SHOULD I MIGRATE MY CODE?

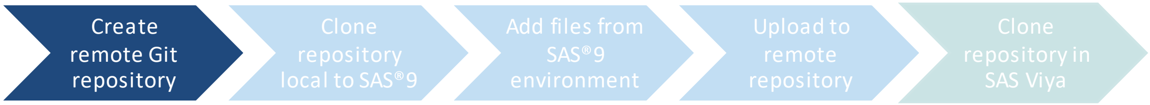
Given many possibilities of transitions from SASR®9 products to SAS Viya, there is not one answer for when to migrate your SAS workflows to SAS Viya.

The great benefit of using Git is that it is very flexible and allows this migration process to work for many common scenarios. SAS workflows can be synchronized to the Git repositories before SAS Viya is even fully set up. After the SAS Viya environment is available, the Git repository is accessed from SAS Studio on SAS Viya and programs can continue to be used.

### WHY DOES A SAS PROGRAMMER NEED TO BE INVOLVED WITH MIGRATION? IS THAT NOT THE ROLE OF A SAS ADMINISTRATOR?

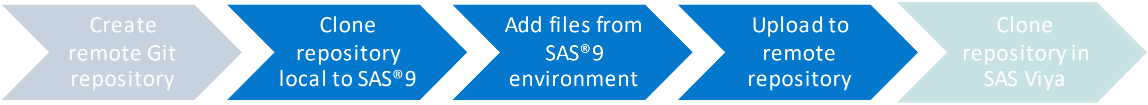
If your company has a well-established structure for code storage that is followed and maintained by all team members, then a SAS programmer might not need to be involved. The process outlined in this paper is for code that lives in a non-structured environment that might be unknown or unseen by the SAS administrator.[2] The power to migrate programs, independently of SAS administrator help, is made easy.

## CREATING A REMOTE GIT REPOSITORY


Create remote Git repository
Clone repository local to SAS®9
Add files from SAS®9 environment
Upload to remote repository
Clone repository in SAS Viya

Before beginning the migration process, an OGRH account needs to be established. If your company does not leverage an internal OGRH, you might consider a publicly available OGRH. Within your OGRH (usually done within a web browser), simply create a project (for example, "SGF Demo Repository").

[2] Based on the authors' experience and working on coding teams, SAS code often lives wherever it needs to, which includes the desktop or in an Enterprise Guide project.



Create remote Git repository

Clone repository local to SAS®9

Add files from SAS®9 environment

Upload to remote repository

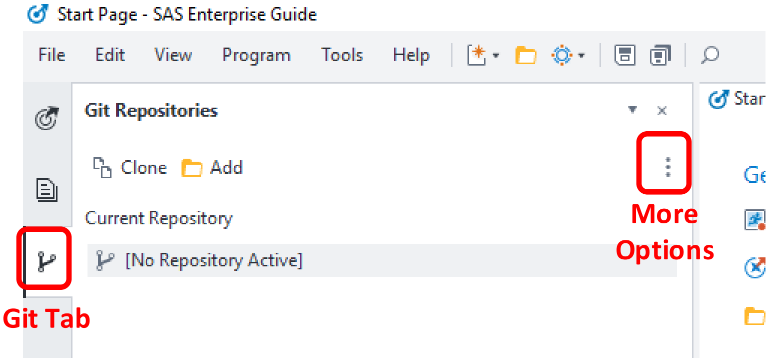
Clone repository in SAS Viya

## SAS ENTERPRISE GUIDE 8 USERS:

SAS Enterprise Guide 8 has an incredibly intuitive interface to interact with your online Git repository directly in the product. Navigate to the Git tab on the far left.

### Clone your remote repository

To clone an online OGRH repository within SAS Enterprise Guide for the first time, click More Options in the upper right corner, and then click Manage Git Connections.



More Options

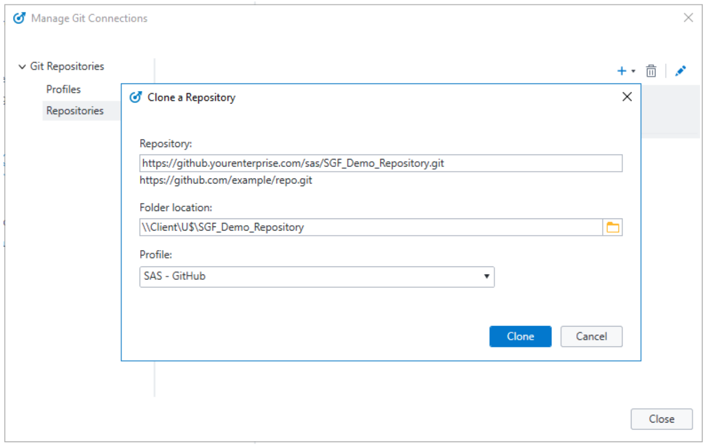
Git Tab

**Figure 4: SAS Enterprise Guide 8 – Git Repository Interface**

Select Add a Profile to connect SAS Enterprise Guide with the online Git repository host. The profile name is used when cloning the repository in the next step.

After the profile is connected, cloning an online repository to make it accessible to SAS Enterprise Guide can be done in one of two ways:

1. Use the Manage Git Connections window, which is similar to creating the Profile. From the Repositories tab, select Clone a new repository and complete the fields.
2. Use the main Git Repositories interface within Enterprise Guide. Click Clone, and then complete the fields.



Repository:
https://github.yourenterprise.com/sas/SGF_Demo_Repository.git
https://github.com/example/repo.git

Folder location:
\\Client\U$\SGF_Demo_Repository

Profile:
SAS - GitHub

**Figure 5: SAS Enterprise Guide 8 – Clone Your Repository to a Local Computer**

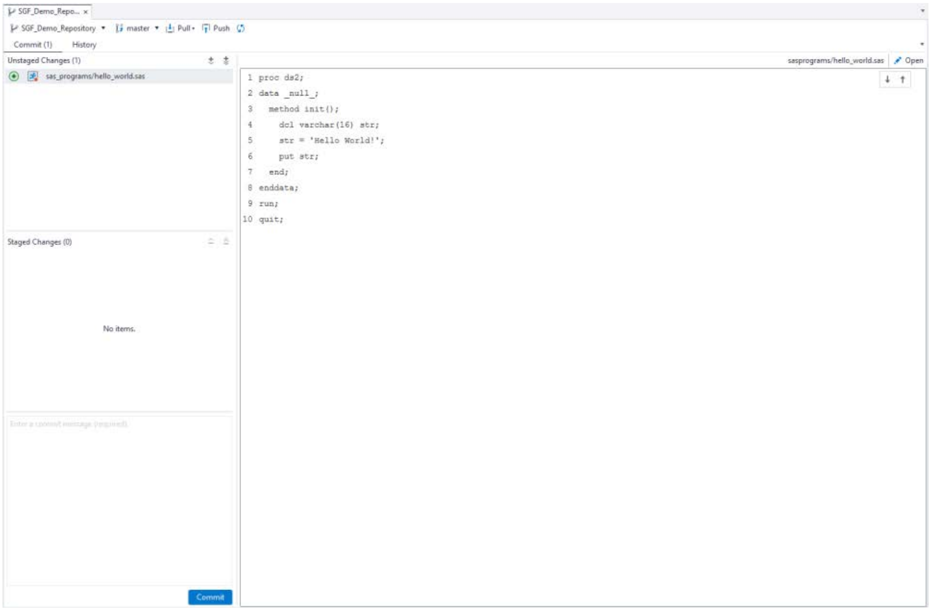
For both methods, the repository URL can be pulled from the online Git repository.

After you click Clone, a created folder structure is home to the new Git repository.

### Add files to your repository

Now that the online Git repository is created and cloned into a local file system that is accessible by SAS Enterprise Guide, files can be added into the repository. The repository can be accessed through the folder location that was specified in the previous step and files can be moved into, updated, or deleted just like any other directory.

Adding any SAS programs or other files to the local repository allows them to be pushed to OGRH in the following step. For example, the 'hello_world.sas' program has been added into a **`sas_programs`** folder in the local repository.

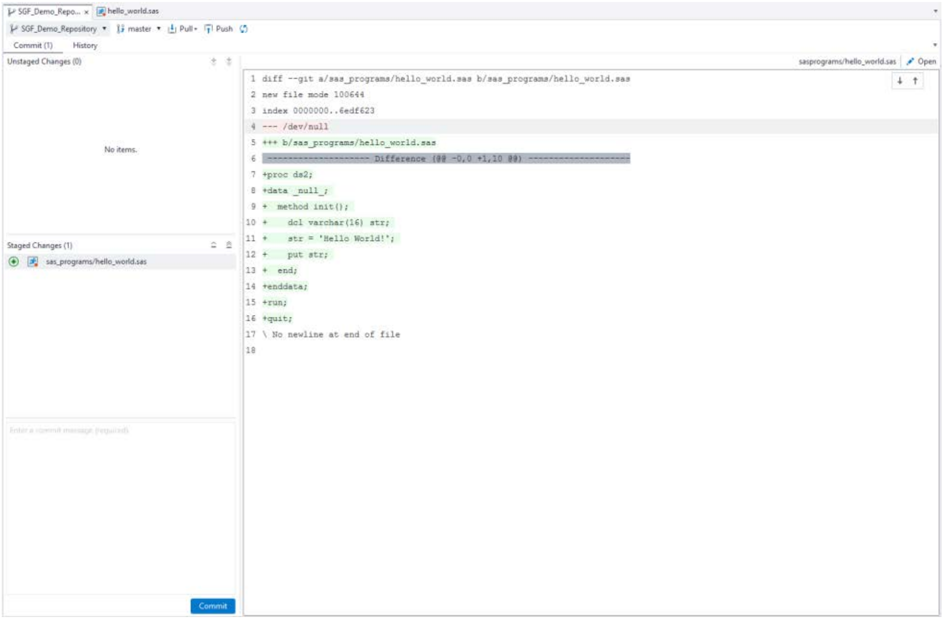


```
proc ds2;
data _null_;
  method init();
    dcl varchar(16) str;
    str = 'Hello World!';
    put str;
  end;
enddata;
run;
quit;
```

Figure 6: SAS Enterprise Guide 8 – New Program Added to Local Repository

### Commit and push to remote repository

After the files that you want to transfer to the OGRH have been added into the local repository, follow the general Git workflow of staging, committing, and pushing changes. To make this process even simpler, this workflow can be completed without ever leaving Enterprise Guide.



```
diff --git a/sas_programs/hello_world.sas b/sas_programs/hello_world.sas
new file mode 100644
index 0000000..6edf623
--- /dev/null
+++ b/sas_programs/hello_world.sas
-------------------- Difference (@@ -0,0 +1,10 @@) --------------------
+proc ds2;
+data _null_;
+  method init();
+    dcl varchar(16) str;
+    str = 'Hello World!';
+    put str;
+  end;
+enddata;
+run;
+quit;
\ No newline at end of file
```

Figure 7: SAS Enterprise Guide 8 – Git Workflow Interface

The Git Repository interface makes this workflow extremely easy in a window that shows each change that is unstaged, staged, or committed to the local repository. After any updates or new programs are committed to the local repository and pushed, the online Git repository is synchronized with the local repository.

Anytime a file or SAS program is added, updated, or removed from the local repository, follow the Git workflow to synchronize them to the online repository.

Congratulations! You have migrated code out of SAS® 9 and are ready to pull down your files into SAS Viya!

## SAS ENTERPRISE GUIDE 7 AND BASE SAS USERS:

Because SAS products have only recently become tightly integrated with Git and able to use the online Git repository[3], the expectation for SAS Enterprise Guide 7 and Base SAS users was established independently of the Git framework. Essentially, this means that much of the functionality—to clone or add files to an online Git repository—must be done outside of SAS.

If this seems disappointing or cumbersome, remember that Git is an incredibly mature program. There are multiple programs that, depending on the OGRH you choose, can make it incredibly easy to migrate your code. (For example, the OGRH GitHub has a GitHub Desktop application.) From a migration standpoint, the most important thing is to get your code off your local server and into an OGRH!

Note: Because interfaces for OGRHs are very different, the next few steps generically address the basic process of adding files to your OGRH. Online documentation is available for the specific OGRH that you choose.

### Clone your remote repository

Clone your project to an environment that can be accessed by both Git and your SAS programs. This might be a SAS environment.

### Add files to your repository

Next, add your important SAS programs and supporting files to the directory structure that contains your Git repository.

### Commit to your local repository

Your files now live in the directory structure, but they are not officially part of your Git repository, so commit them.

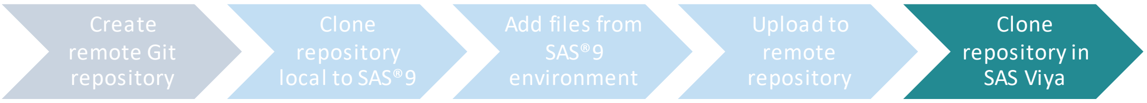
### Push to your remote repository

Now that you have added all your important files to your local Git repository, make sure they get pushed into the remote repository!

Congratulations! You have migrated code out of SAS® 9 and are ready to pull down your files into SAS Viya!

[3] Limited Git functionality was introduced to SAS Enterprise Guide 7, but because this functionality does not include OGRH communication, it is not discussed in this migration paper.

# MIGRATING CODE INTO SAS VIYA


Create remote Git repository
Clone repository local to SAS®9
Add files from SAS®9 environment
Upload to remote repository
Clone repository in SAS Viya

Now that your code is stored in an OGRH, it can be migrated into SAS Viya through SAS Studio. SAS Studio, like SAS Enterprise Guide 8, has Git integration built fully into the product. When you open SAS Studio, navigate to the Git tab on the far left.

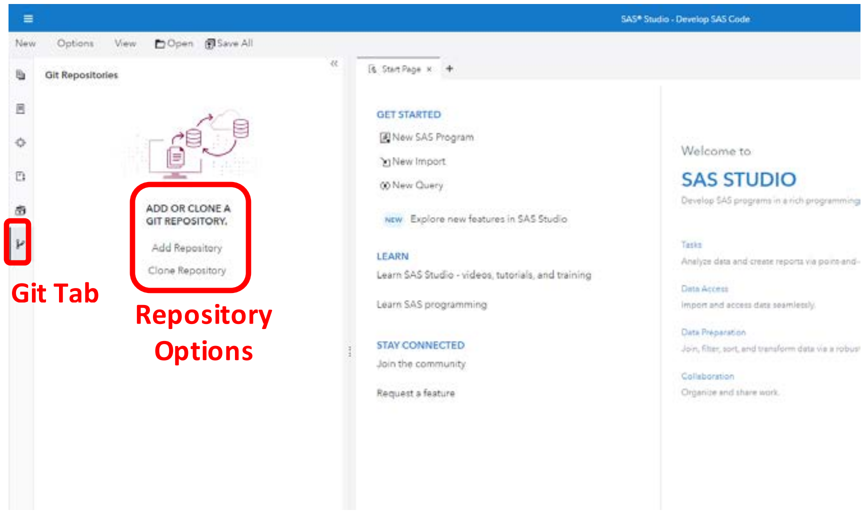
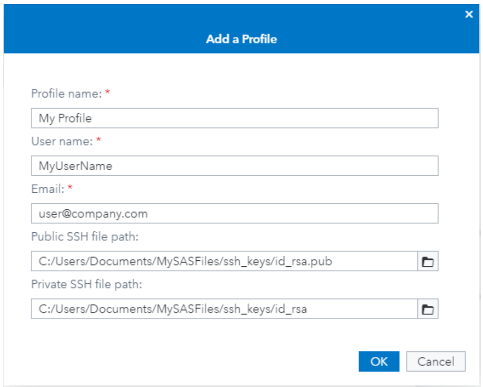

SAS® Studio - Develop SAS Code
New
Options
View
Open
Save All
Git Repositories
ADD OR CLONE A GIT REPOSITORY.
Add Repository
Clone Repository
Git Tab
Repository Options
Start Page
GET STARTED
New SAS Program
New Import
New Query
NEW Explore new features in SAS Studio
LEARN
Learn SAS Studio - videos, tutorials, and training
Learn SAS programming
STAY CONNECTED
Join the community
Request a feature
Welcome to
SAS STUDIO
Develop SAS programs in a rich programming
Tasks
Analyze data and create reports via point-and-
Data Access
Import and access data seamlessly
Data Preparation
Join, filter, sort, and transform data via a robus
Collaboration
Organize and share work.

**Figure 8: SAS Studio on SAS Viya Home Page**

## CLONE YOUR REMOTE REPOSITORY

Cloning a repository is one of the main options from the Git Repositories view.

When you click Clone Repository, you are prompted to create a connection and a Git profile. One key difference between this profile to OGRH and the profile that you set up in Enterprise Guide is that SAS Viya uses Secure Shell (SSH) keys to authenticate with the online Git repository. The public/private key pair should be created in a location that is accessible by SAS Viya because the path to these keys is needed to set up the Git profile. The public key also needs to be added to the online Git repository as a deploy key. For more information about deploy keys that are specific to OGRHs, click here.



Add a Profile

Profile name: * My Profile

User name: * MyUserName

Email: * user@company.com

Public SSH file path: C:/Users/Documents/MySASFiles/ssh_keys/id_rsa.pub

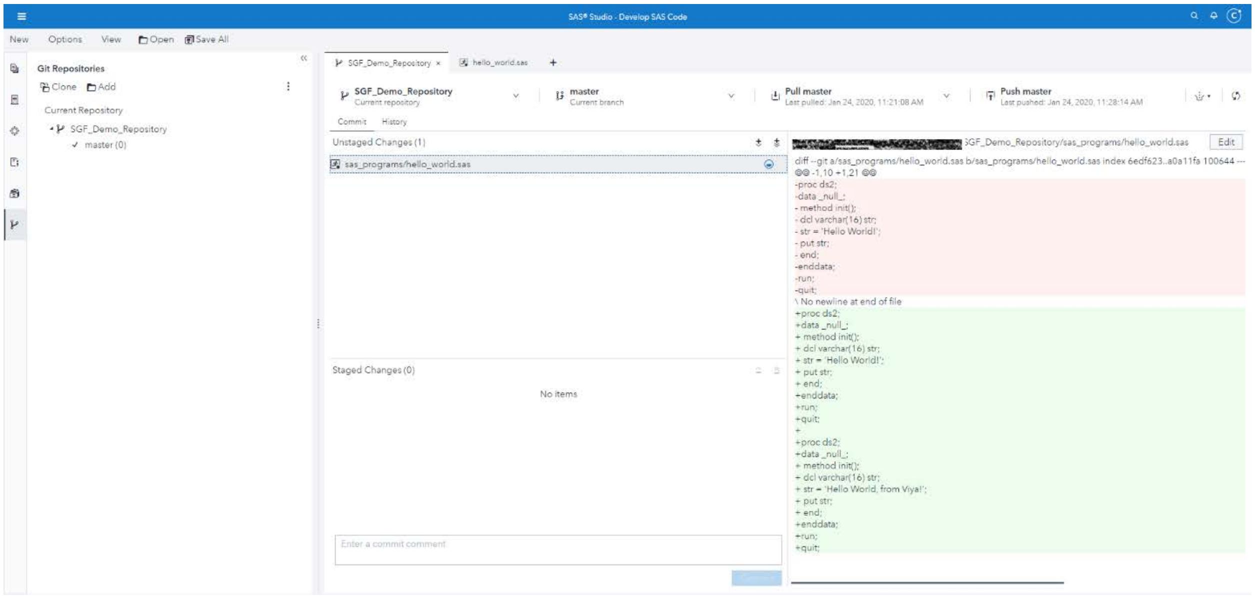
Private SSH file path: C:/Users/Documents/MySASFiles/ssh_keys/id_rsa

**Figure 9: SAS Studio – Create a Git Profile**

After your profile is established, create a connection to the same online repository (for example, SASGF Demo Repository) that was used in previous steps to migrate your files off SAS®9. SAS Studio prompts you for a location to save this repository that is local to the SAS Viya server.

After successful cloning, sit back and admire your work. There are two local repositories connecting to one online Git repository—one that is accessible by the SAS®9 file system and one that is accessible by the SAS Viya file system.

Congratulations! You have successfully migrated your SAS programs from SAS®9 into SAS Viya!



```
diff --git a/sas_programs/hello_world.sas b/sas_programs/hello_world.sas index 6edf623..a0a11fa 100644 ---
@@ -1,10 +1,21 @@
-proc ds2;
-data _null_;
- method init();
- dcl varchar(16) str;
- str = 'Hello World!';
- put str;
- end;
-enddata;
-run;
-quit;
\ No newline at end of file
+proc ds2;
+data _null_;
+ method init();
+ dcl varchar(16) str;
+ str = 'Hello World!';
+ put str;
+ end;
+enddata;
+run;
+quit;
+
+proc ds2;
+data _null_;
+ method init();
+ dcl varchar(16) str;
+ str = 'Hello World, from Viya!';
+ put str;
+ end;
+enddata;
+run;
+quit;
```

**Figure 10: SAS Studio – Git Workflow Interface**

## BEYOND: CONTINUING TO USE GIT

Using Git to help migrate SAS programs and files from SAS®9 to SAS Viya is just the beginning of how SAS can leverage Git.

With SAS Studio in SAS Viya containing a very useful integration with Git, you can continue to use the Git system to track changes and manage version control without leaving SAS Viya.

SAS Studio contains features to clone repositories, commit and stash file changes, pull and push files, view repository history, create and merge branches, rebase a branch, and perform a basic differentiation between files in your local repository.

For more information about how SAS Studio integrates with Git, see the documentation here.

## CONCLUSION

With a little extra guidance and a little help from Git, you should have no apprehension about moving your code to SAS Viya by yourself. Because SAS has adopted Git as its main integrated code versioning tool, you can have confidence that your SAS is looking to leverage Git and make your move to SAS Viya as easy as possible!

## REFERENCES

Kinnebrew, Brian. 2019. "Leveraging SAS® Viya® to Improve SAS® Processes." https://communities.sas.com/t5/Ask-the-Expert/Leveraging-SAS-Viya-to-Improve-SAS-Processes/ta-p/600509. Last modified October 30, 2019.

SAS Institute Inc. 2019. "Understanding Git Integration in SAS Studio." In *SAS Studio 5.2 User's Guide*. Cary, NC. SAS Institute Inc. https://go.documentation.sas.com/?activeCdc=webeditorcdc&cdcId=sasstudiocdc&cdcVersion=5.2&docsetId=webeditorug&docsetTarget=p0puc7muifjjycn1uemlm9lj1jkt.htm&locale=en.

GitHub Developer. 2020. "Managing Deploy Keys." https://developer.github.com/v3/guides/managing-deploy-keys/#deploy-keys.

Flynn, Joe, Casey Smith, and Alex Song. 2014. "Check It Out! Versioning in SAS® Enterprise Guide®." *Proceedings of the SAS Global Forum 2014 Conference*. Cary, NC: SAS Institute Inc. http://support.sas.com/resources/papers/proceedings14/SAS179-2014.pdf

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors:

Cameron Estroff
SAS Institute Inc.
cameron.estroff@sas.com

Weston McManus
SAS Institute Inc.
weston.mcmanus@sas.com