

How to Use Git with Your SAS Projects



Chris Hemedinger is the Director of SAS User Engagement. His talented team looks after SAS online communities, SAS user groups, developer experience and GitHub, tech newsletters, expert webinars and tutorials. Chris is a recovering software developer who helped build popular SAS products such as SAS Enterprise Guide. Inexplicably, Chris is still coasting on the limited fame he earned as an author of *SAS For Dummies*.

What you'll learn

- ▶ What is Git™?
- ▶ Getting started with Git commands
- ▶ Using Git with SAS tools
 - SAS Studio (SAS Viya and SAS v9.4)
 - SAS Enterprise Guide
- ▶ Git functions in SAS programming language
- ▶ DevOps opportunities: collaboration and continuous integration
- ▶ Where to learn more


Prerequisites & setup

These tools/services will help you get the most from this hands-on session:

- ▶ Git client: Install from <https://www.git-scm.org/>
- ▶ GitHub account: free at <http://www.github.com/join>
- ▶ SAS OnDemand for Academics account (for some exercises)
- ▶ Optional: **SAS Enterprise Guide 8.2 or 8.3** with access to SAS
- ▶ Optional: **VS Code**
- ▶ Optional: **SAS Studio 3.8** with access to SAS

Credit – *Professional Git* by Brent Laster

- ▶ Extensive Git reference, explanations, and examples
- ▶ First part for non-technical
- ▶ Beginner and advanced reference
- ▶ Hands-on labs

 Amazon Customer

★★★★★ I can't recommend this book more highly

February 12, 2017

Format: Kindle Edition

Brent Laster's book is in a different league from the many print and video sources that I've looked at in my attempt to learn Git. The book is extremely well organized and very clearly written. His decision to focus on Git as a local application for the first several chapters, and to defer discussion about it as a remote application until later in the book, works extremely well.

Laster has also succeeded in writing a book that should work for both beginners and people with a fair bit of experience with Git. He accomplishes this by offering, in each chapter, a core discussion followed by more advanced material and practical exercises.

I can't recommend this book more highly.

★★★★★ Ideal for hands-on reading and experimentation

February 23, 2017

Format: Paperback | **Verified Purchase**

I just finished reading *Professional Git*, which is well organized and clearly presented. It works as both a tutorial for newcomers and a reference book for those more experienced. I found it ideal for hands-on reading and experimentation with things you may not understand at first glance. I was already familiar with Git for everyday use, but I've always stuck with a convenient subset. It was great to be able to finally get a much deeper understanding. I highly recommend the book.

Professional Git 1st Edition

by Brent Laster (Author)

★★★★★ 7 customer reviews

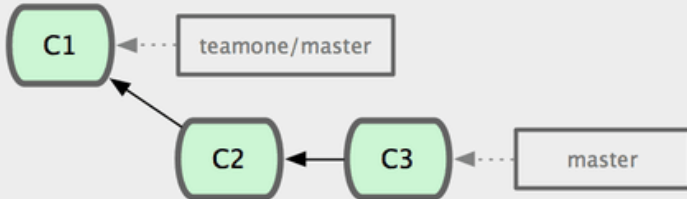
[Look inside](#) ↓



git.team1.ourcompany.com



My Computer



Source: <https://git-scm.com>

What is Git?

Git is a **distributed version control system** that supports several collaborative and standalone workflows

- ▶ **Distributed** – each developer has a “clone” of the code repository
- ▶ **Workflows** – feature-based, experiment branches, prod/test/dev, patch/cherry-pick
- ▶ **Open source** – Git software is free and open source.

Several commercial systems (GitHub, GitLab, Bitbucket, and more) add features and enterprise-readiness.

Benefits of using Git with SAS

Author: Mike Henderson | Date: 4/29/2020 15:54:21 | Commit hash: a059de1248793e17b9ff00ea8b194ecfb81f384e

flexible number of ISOChangeDate and SocialDistancingChange

- ISOChangeDate is now a string delimited by : that can handle none, sing
- SocialDistancingChange is now a string delimited by : that can handle no
- Break - If the file run_scenarios.csv has a column added it will not be use

CCF/COVID19.sas

```

7
8 %macro EasyRun (Scenario,Incubatio
9 - SocialDistancing,
10 - ISOChangeDate3,So
11 + SocialDistancing,
12 %*atShareParam

```

Version control

Start Page covid-19-sas x

master Pull Push

Commit (95) History

Graph	Message	Author	Date	Commit ID
	new cc templates	Mike Henderson	4/30/2020 15:10:2	9161
	Merge pull request #6 from sassoftware/Social-Distanc	Mike Henderson	4/30/2020 13:17:2	d4e0
	origin/Social-Distancing-Shifts Adjusted to handle	Mike Henderson	4/30/2020 13:05:1	6ec1
	Added supporting documentation	Chris Hemedinger	4/30/2020 11:45:2	8980
	rebuild: change in Sigma default in keyword paramete	Mike Henderson	4/30/2020 10:48:3	6596
	Add SAS programs to fetch public COVID-19 data	Chris Hemedinger	4/30/2020 10:37:3	ee08
	✓ master Add SAS code to fetch public COVID-19	Chris Hemedinger	4/30/2020 10:11:1	d2e382c
	progress	Mike Henderson	4/30/2020 09:46:2	379c816
	flexible number of ISOChangeDate and SocialDistancir	Mike Henderson	4/29/2020 15:54:2	a059de1
	Update orchestrate.sas	Andrew Williams	4/29/2020 11:04:3	9be4870
	remove extra variables from model_final	Mike Henderson	4/28/2020 11:30:3	e5ae023

Collaboration

DevOps/automation

GitHub Maintainers > sassoftware.org

Conversations Files Wiki

GitHub 5/1 12:36 PM

Pull request closed: "fix a bug in load_weights\(\)"

kesmitt3
5/1/2019 12:36 PM (UTC-05:00)

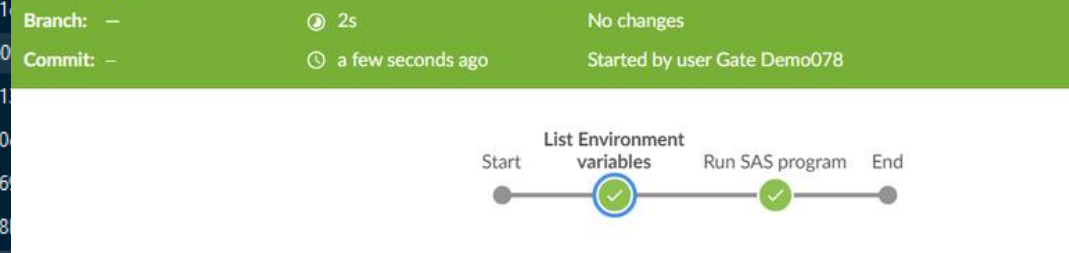
The push fixes a bug in load_weights(). The issue happens when load caffemodel.h5 as a model and model weights. The path option is a server-side path. 'if os.path.isfile(path):' is intended to check if the file exists but it is operating on client-side. I don't think we have a method to check server-side file by client python, so I remove the outer if statement.

Repository: sassoftware/python-dlpy

Pull request #: 133

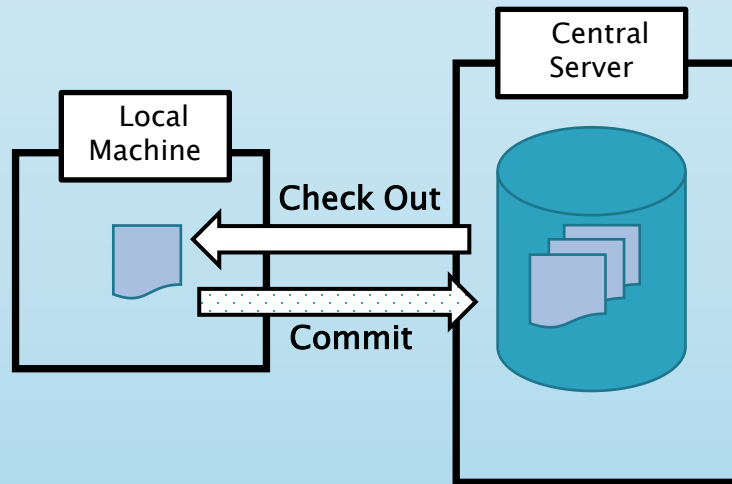
Add a comment View in GitHub

Reply

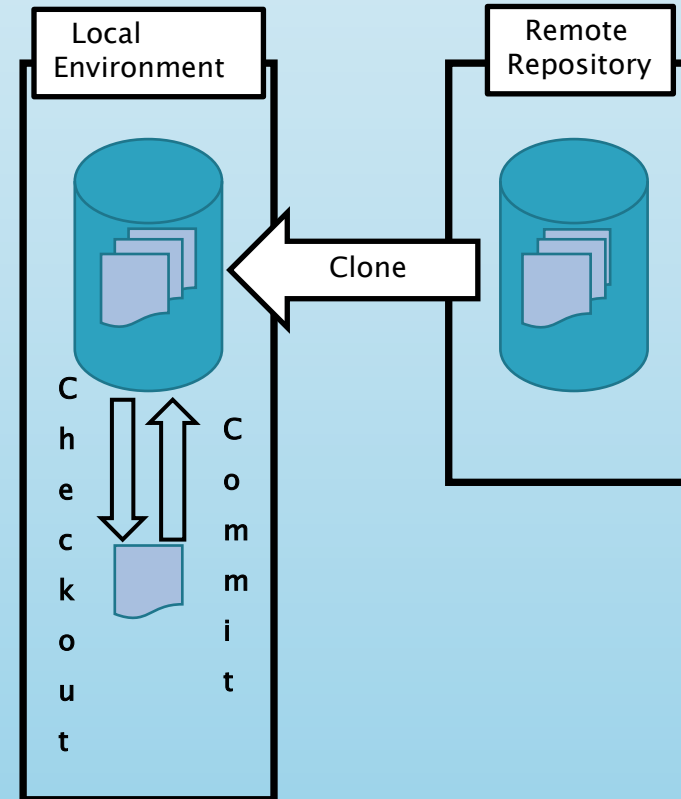


Centralized vs. Distributed VCS

Centralized Version Control Model



Distributed Version Control Model



Git has its own lingo

Clone

- Create a local copy of a repository

Fork

- Make a local copy of a repo where you don't have push access

Pull request

- Open a discussion with proposed code changes to be merged to upstream repo

Branch

- A logical location to divert from the "main line" of code to stage changes

Blame

- Annotated code listing with revision/author for each line

Commit

- Record a set of changes to the local repo

Commit History

- Log that shows chronological list of changes to the repo

Push / Pull

- Update a remote repo with local changes, or sync/merge local repo with remote changes

SAS tools that integrate with Git

SAS Enterprise
Guide

SAS Studio

SAS
programming

SAS Data
Integration
Studio

And more!
SAS Viya products
add more Git
integration

SAS Enterprise Guide with external repo

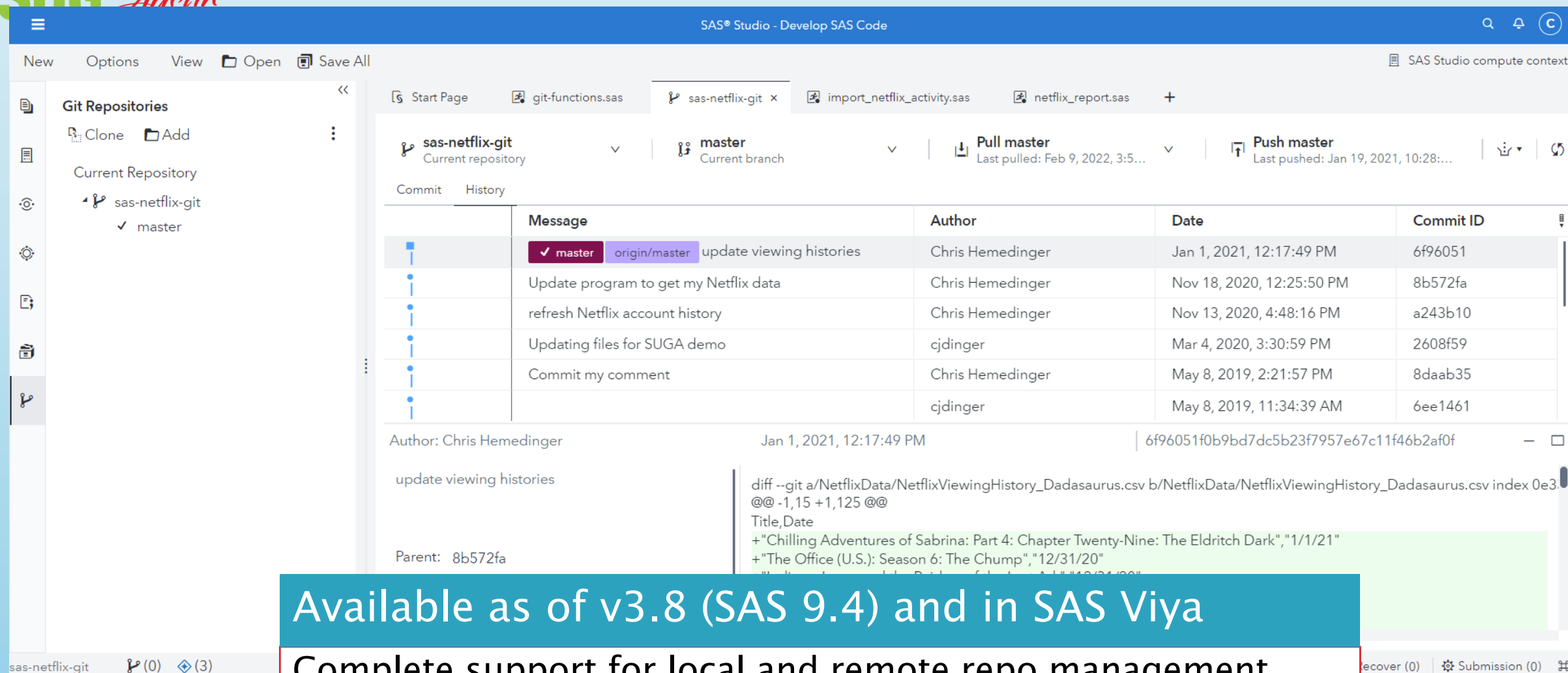
The screenshot displays the SAS Enterprise Guide (MIS 9.4) interface with a Git repository named 'sas-youtube-api' open. The interface is divided into several panes:

- Git Repositories:** A sidebar on the left showing the current repository 'sas-youtube-api' and other repositories like 'sas-dummy-blog', 'sas-netflix-git', 'learn-sas', and 'sas-zip-macros'.
- Start Page:** The main pane shows the 'sas-youtube-api' repository with a commit history table.

Commit (0)	History	Graph	Message	Author	Date	Commit ID
			✓ main origin/main origin/master Catch up work	sasrh	8/12/2020 21:29:5	e5d63c7
			Merge branch 'master' of gitlab.sas.com:sasrh/sas-youtube-api	Chris Hemedinger	5/7/2020 16:27:07	e416ef5
			Add support for Brightcove SASGF videos	Chris Hemedinger	5/7/2020 16:24:27	daa1702
			Add README.md	Chris Hemedinger	5/1/2020 14:50:27	e274846
			Add shell script to run	Chris Hemedinger	5/1/2020 14:25:58	3b12157
			Add SAS programs to pull data from YouTube	Chris Hemedinger	5/1/2020 14:14:01	70424d2
- Authors:** A section below the commit history showing the author 'sasrh', date '8/12/2020 21:29:51', and commit hash 'e5d63c7994af2dbdbaba66af63f3f6c78a6d361b'.
- Files:** A list of files in the repository, including 'adhoc_plots.sas', 'adhoc_plots2.sas', 'bc_prep.sas', 'bc_sasgf_monthly.sas', 'youtube.sh', 'yt_sasgf_meta.sas', 'yt_sasuserschannel_buildvatable.sas', 'yt_sasuserschannel_countries.sas', 'yt_sasuserschannel_groups.sas', and 'yt_sasuserschannel_uploads.sas'.
- Code Editor:** The right pane shows the content of 'adhocplots.sas', which includes a diff view and SAS code for generating a plot of YouTube views.

All history maintained within Git repo available to any tool that integrates with Git.

Git integration with SAS Studio



SAS Studio - Develop SAS Code

New Options View Open Save All

SAS Studio compute context

Git Repositories

Clone Add

Current Repository

sas-netflix-git

✓ master

Start Page git-functions.sas sas-netflix-git x import_netflix_activity.sas netflix_report.sas +

sas-netflix-git Current repository

master Current branch

Pull master Last pulled: Feb 9, 2022, 3:5...

Push master Last pushed: Jan 19, 2021, 10:28:...

Commit	History	Message	Author	Date	Commit ID
✓ master	origin/master	update viewing histories	Chris Hemedinger	Jan 1, 2021, 12:17:49 PM	6f96051
		Update program to get my Netflix data	Chris Hemedinger	Nov 18, 2020, 12:25:50 PM	8b572fa
		refresh Netflix account history	Chris Hemedinger	Nov 13, 2020, 4:48:16 PM	a243b10
		Updating files for SUGA demo	cjdingler	Mar 4, 2020, 3:30:59 PM	2608f59
		Commit my comment	Chris Hemedinger	May 8, 2019, 2:21:57 PM	8daab35
			cjdingler	May 8, 2019, 11:34:39 AM	6ee1461

Author: Chris Hemedinger Jan 1, 2021, 12:17:49 PM 6f96051f0b9bd7dc5b23f7957e67c11f46b2af0f

update viewing histories

diff --git a/NetflixData/NetflixViewingHistory_Dadasaurus.csv b/NetflixData/NetflixViewingHistory_Dadasaurus.csv index 0e3...@@ -1,15 +1,125 @@Title,Date+ "Chilling Adventures of Sabrina: Part 4: Chapter Twenty-Nine: The Eldritch Dark", "1/1/21"+ "The Office (U.S.): Season 6: The Chump", "12/31/20"

Parent: 8b572fa

recover (0) Submission (0)

Available as of v3.8 (SAS 9.4) and in SAS Viya

Complete support for local and remote repo management

New Git functions in SAS language

Functions added in SAS 9.4 Maint 6 and SAS Viya

New SAS functions that mirror most of the Git commands that users are familiar with.

Select functions – All documented by searching [“Git functions” on support.sas.com](https://support.sas.com).

GITFN_CLONE	Clones a Git repository (for example, from GitHub) into a directory on the SAS server.
GITFN_COMMIT	Commits staged files to the local repository
GITFN_DIFF	Returns the number of diffs between two commits in the local repository and creates a diff record object for the local repository.
GITFN_PUSH	Pushes the committed files in the local repository to the remote repository.
GITFN_NEW_BRANCH	Creates a Git branch

Name change: GITFN* → GIT* in SAS Viya

Connecting SAS to Git

SSH or HTTPS

SAS Studio

Supports SSH or HTTPS

Edit Profile

SSH HTTPS

Profile name: *

Chris (GitHub)

User name: *

cjdinger

Email: *

Public SSH file path:

Private SSH file path:

OK Cancel

SAS Enterprise Guide 8.2

Supports HTTPS

Manage Git Connections

Git Repositories

Profiles Repositories

GitHub GitLab

Edit Profile

Profile name: *

GitHub

User name: *

cjdinger

Email: *

Password:

OK Cancel

SAS GIT* functions

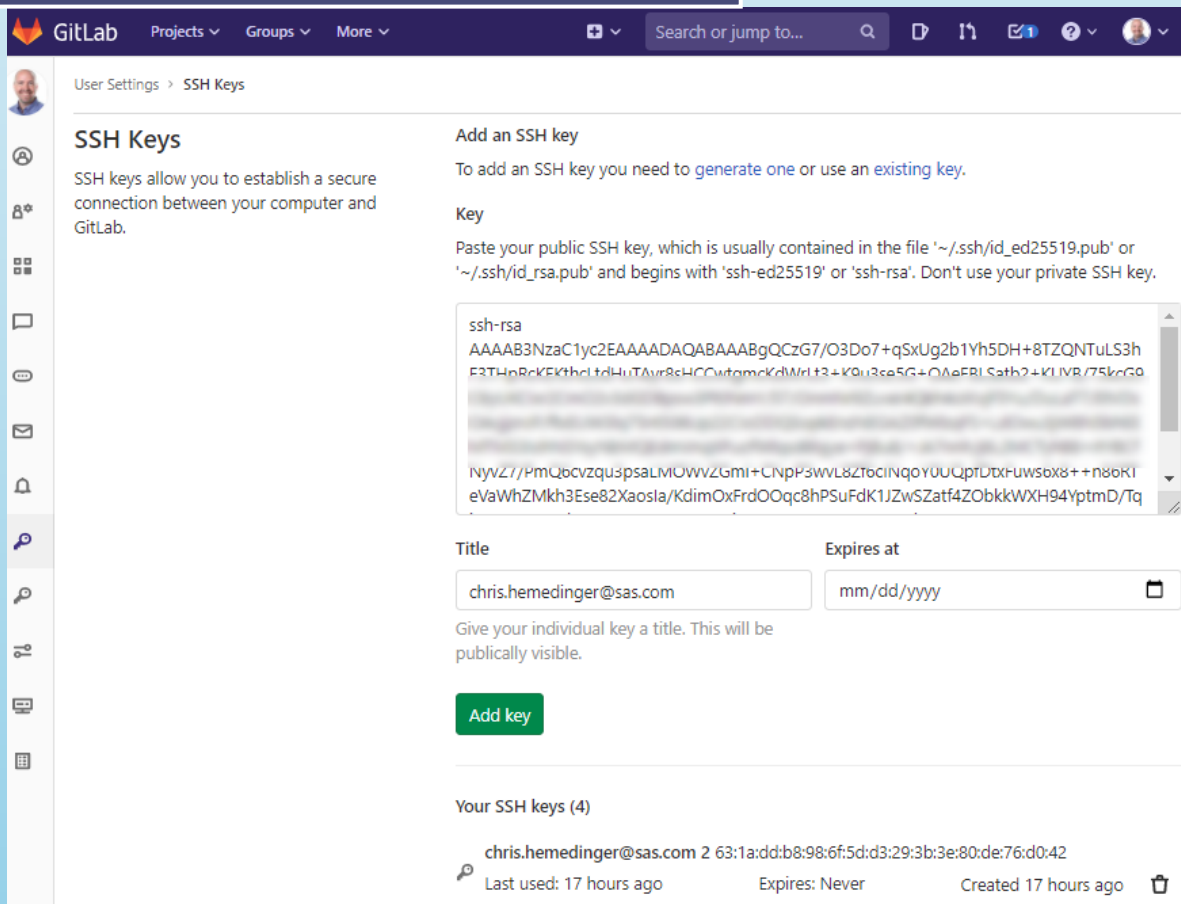
Supports SSH or HTTPS

```
/* Fetch latest code from GitHub */  
data _null_;  
  rc = gitfn_clone(  
    "git@github.com:sascommunities/sas-dummy-blog.git",  
    "&repoPath.",  
    "cjdinger", ""  
    "/u/sasrh/.ssh/id_rsa.pub",  
    "/u/sasrh/.ssh/id_rsa");  
  put rc=;  
run;
```

Register SSH key

GitLab example

GitLab settings



The screenshot shows the GitLab 'User Settings' page for 'SSH Keys'. The left sidebar contains navigation icons. The main content area has a heading 'SSH Keys' and a description: 'SSH keys allow you to establish a secure connection between your computer and GitLab.' Below this is a section 'Add an SSH key' with instructions: 'To add an SSH key you need to generate one or use an existing key.' A 'Key' section contains a text area with a public SSH key. Below the key is a form with 'Title' (set to 'chris.hemedinger@sas.com') and 'Expires at' (set to 'mm/dd/yyyy'). A green 'Add key' button is at the bottom. A section 'Your SSH keys (4)' shows a table with one entry: 'chris.hemedinger@sas.com 2 63:1a:dd:b8:98:6f:5d:d3:29:3b:3e:80:de:76:d0:42', with 'Last used: 17 hours ago', 'Expires: Never', and 'Created 17 hours ago'.

GitLab Projects Groups More Search or jump to...

User Settings > SSH Keys

SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab.

Add an SSH key

To add an SSH key you need to [generate one](#) or use an [existing key](#).

Key

Paste your public SSH key, which is usually contained in the file '~/.ssh/id_ed25519.pub' or '~/.ssh/id_rsa.pub' and begins with 'ssh-ed25519' or 'ssh-rsa'. Don't use your private SSH key.

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGCzG7/O3Do7+qSxUg2b1Yh5DH+8TZQNTuLS3h
E3THoRcKEKthcltdHuTAyr8HCCvfmKdMrlt3+K9u3se5G+QAeFRLSath2+KLIvR/75kcG9
...
InyVZ//PmQbcvzqu3psaLMUwvZGmi+CnpPswVL8ZtcINqoY0UQpTUTxUwsox8++n8oKI
eVaWhZMkh3Ese82Xaosla/KdimOxFrDOOqc8hPSuFdK1JJwSZatf4ZObkkWXH94YptmD/Tq
```

Title **Expires at**

chris.hemedinger@sas.com mm/dd/yyyy

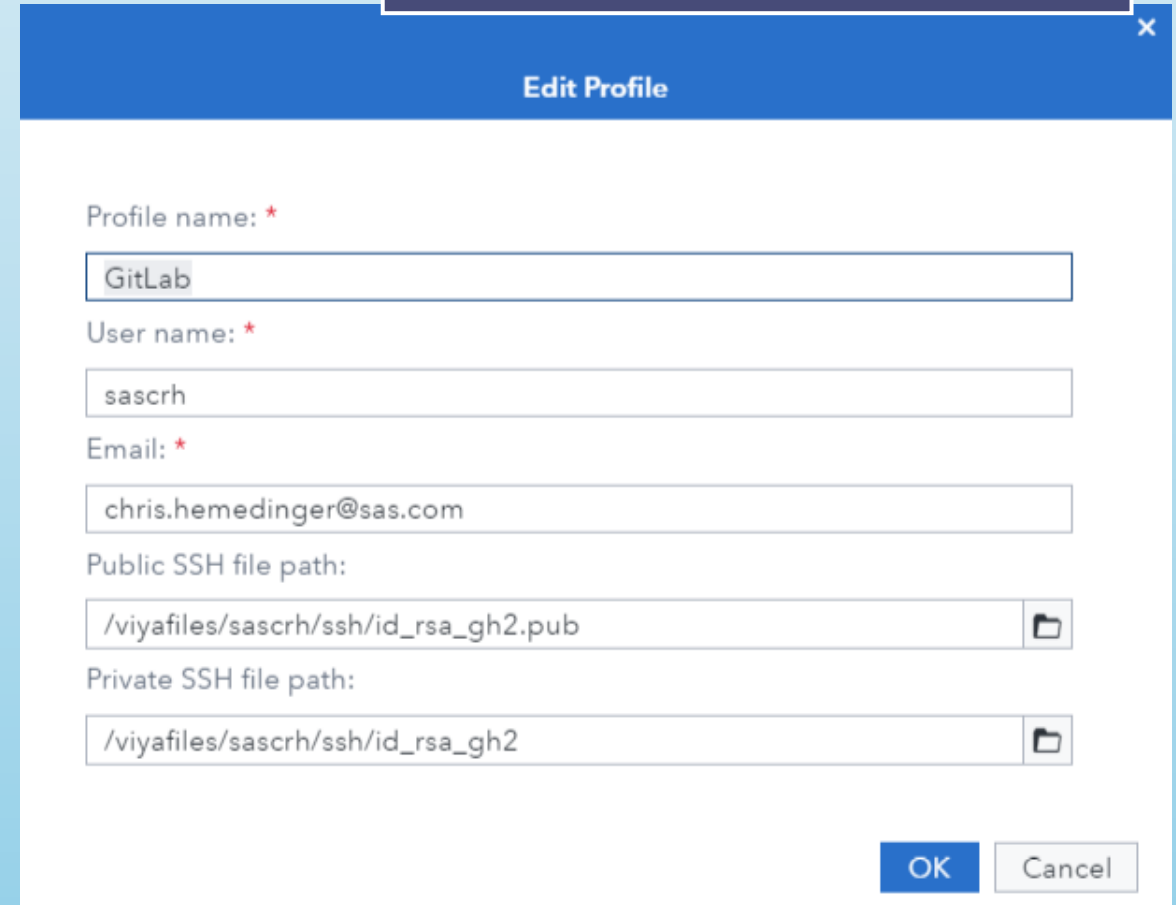
Give your individual key a title. This will be publically visible.

Add key

Your SSH keys (4)

chris.hemedinger@sas.com 2	63:1a:dd:b8:98:6f:5d:d3:29:3b:3e:80:de:76:d0:42
Last used: 17 hours ago	Expires: Never Created 17 hours ago

SAS Studio settings



The screenshot shows the 'Edit Profile' dialog in SAS Studio. It has a blue header bar with the title 'Edit Profile'. The form contains several fields: 'Profile name' (set to 'GitLab'), 'User name' (set to 'saschr'), 'Email' (set to 'chris.hemedinger@sas.com'), 'Public SSH file path' (set to '/viyafiles/saschr/ssh/id_rsa_gh2.pub'), and 'Private SSH file path' (set to '/viyafiles/saschr/ssh/id_rsa_gh2'). At the bottom right are 'OK' and 'Cancel' buttons.

Edit Profile

Profile name: *

GitLab

User name: *

saschr

Email: *

chris.hemedinger@sas.com

Public SSH file path:

/viyafiles/saschr/ssh/id_rsa_gh2.pub

Private SSH file path:

/viyafiles/saschr/ssh/id_rsa_gh2

OK Cancel

Generate an access token for HTTPS

Required for single-signon (SSO) or 2-factor auth

- ▶ HTTPS access requires username and password
- ▶ Instead of account password, use access token
- ▶ Access tokens convey specific permissions (api, read_repository, write_repository, etc.)
- ▶ Protect your access token!

Creating a personal access token

You should create a personal access token to use in place of a password with the command line or with the API.

Personal access tokens (PATs) are an alternative to using passwords for authentication to GitHub when using the [GitHub API](#) or the [command line](#).

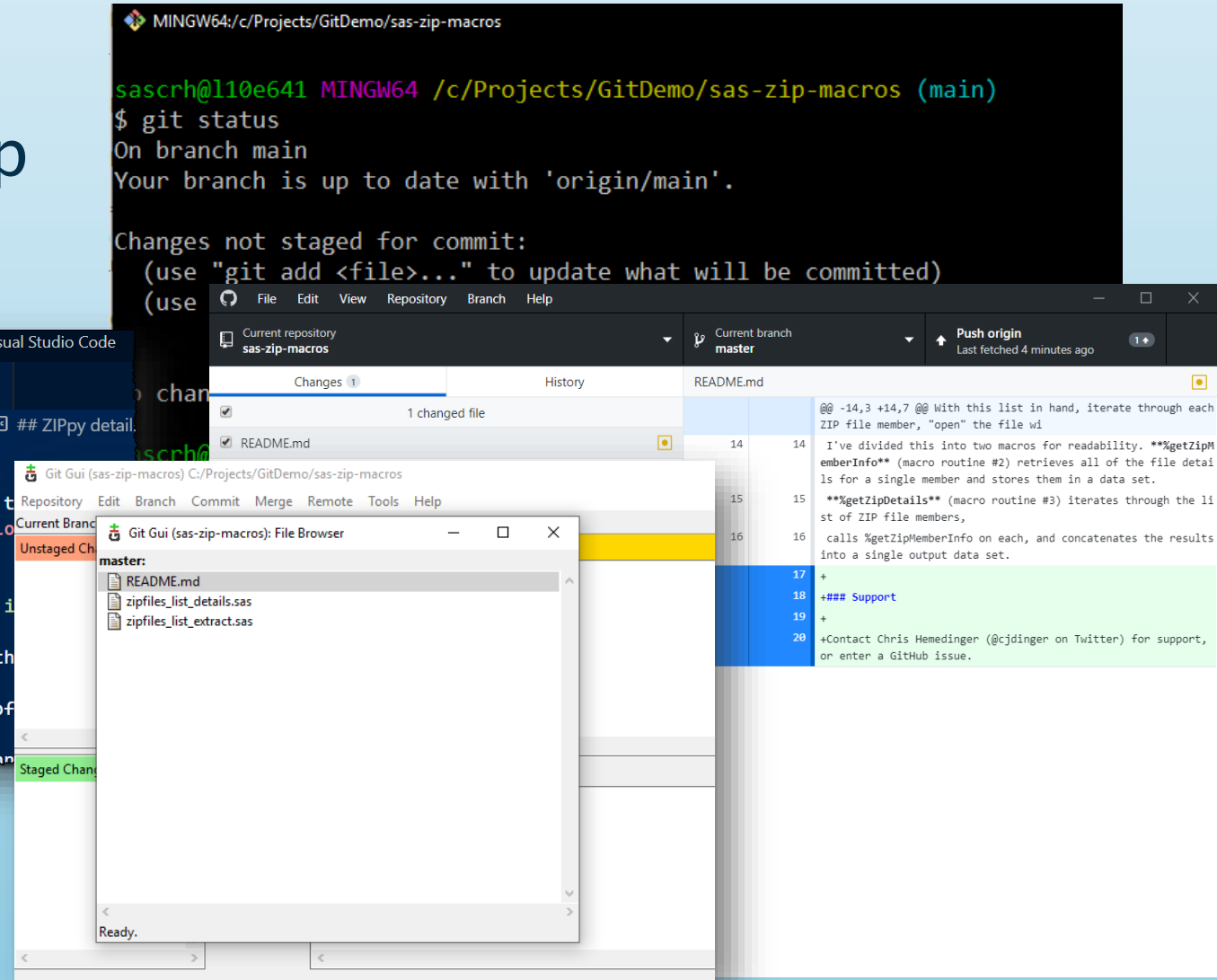
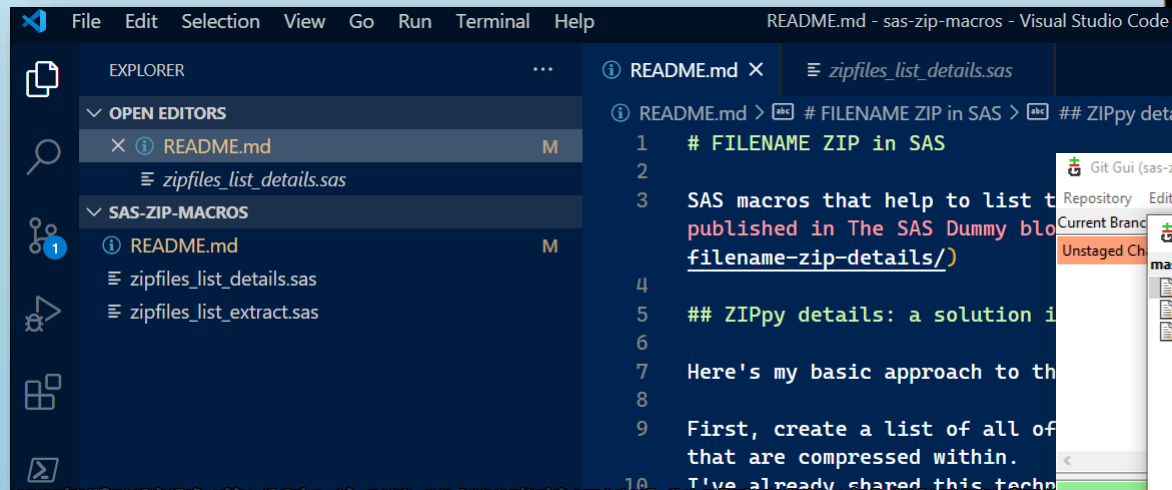
If you want to use a PAT to access resources owned by an organization that uses SAML SSO, you must authorize the PAT. For more information, see "[About authentication with SAML single sign-on](#)" and "[Authorizing a personal access token for use with SAML single sign-on](#)."

As a security precaution, GitHub automatically removes personal access tokens that haven't been used in a year.

From GitHub doc

Use other tools alongside SAS

- ▶ Git Bash (CLI)
- ▶ Git GUI or GitHub Desktop
- ▶ VS Code



How to organize your SAS projects in Git

Read the blog article

- ▶ Don't create one huge repo
- ▶ Don't organize based on dept org structure
- ▶ Design with collaboration in mind
- ▶ Re-use 👍, Ownership 👎
- ▶ Work with Git features, not against them

How to organize your SAS projects in Git

By [Chris Hemedinger](#) on [The SAS Dummy](#) | November 10, 2020

Topics | [Programming Tips](#)

The code and data that drive analytics projects are important assets to the organizations that sponsor them. As such, there is a growing trend to manage these items in the source management systems of record. For most companies these days, that means Git. The specific system might be GitHub Enterprise, GitLab, or Bitbucket -- all platforms that are based on Git.

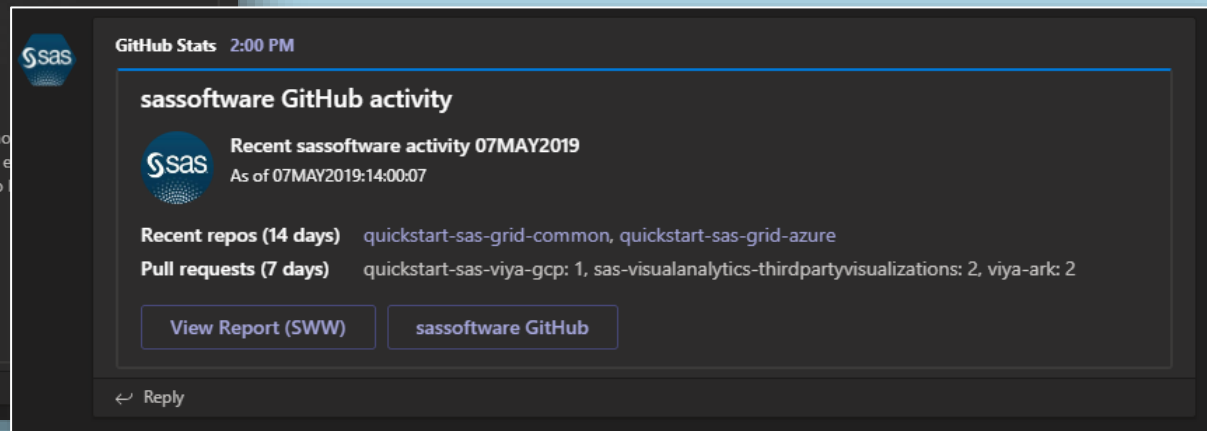
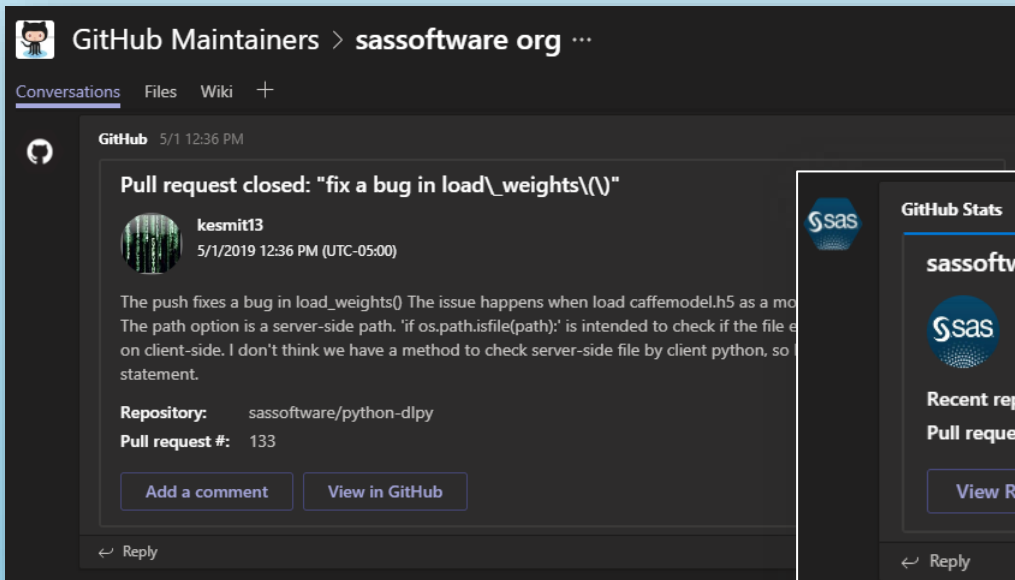
Many SAS products support direct integration with Git. This includes SAS Studio, SAS Enterprise Guide, and the SAS programming language. (That last one checks a lot of boxes for ways to use Git and SAS together.) While we have good documentation and videos to help you learn about Git and SAS, we often get questions around "best practices" -- what is **the best/correct way** to organize your SAS projects in Git?

<https://blogs.sas.com/content/sasdummys/sas-projects-git/>

Git integration is just the beginning

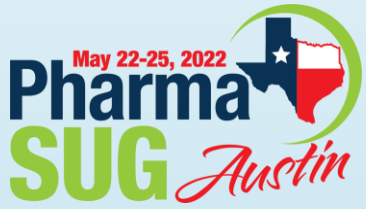
From there, trigger other operations

- ▶ Gerrit for code review
- ▶ Jenkins or Travis CI for build, deployment, and other continuous integration
- ▶ Trigger notifications/alerts in Slack, Microsoft Teams



Learn more

- ▶ [Using Git with SAS](#) (SAS Users YouTube)
- ▶ [Git functions in SAS 9.4 and SAS Viya](#) (doc)
- ▶ [Using Git in SAS Enterprise Guide](#) (doc)
- ▶ [Git with SAS Studio and SAS Enterprise Guide](#) (video)
- ▶ [SAS Software on GitHub](#)
- ▶ [Using built-in Git operations in SAS](#) (blog)
- ▶ [How to organize your SAS projects in Git](#) (blog)
- ▶ [DevOps with SAS 9: SAS code, GitLab, and Jenkins](#) (community)
- ▶ [developer.sas.com](#) for SAS app development
- ▶ [Pro Git](#) by Scott Chacon and Ben Straub, free online book about Git



Name:	Chris Hemedinger
Affiliation:	Director of SAS User Engagement
Contact Number:	919-757-5391
E-mail:	Chris.Hemedinger@sas.com
Website:	blogs.sas.com/sasdummy
Twitter:	@cjdinger
LinkedIn:	https://linkedin.com/in/sasdummy