

Today's topics



- What is Git™?
- Using Git with SAS applications
- Git functions in SAS programming language
- DevOps opportunities: collaboration and continuous integration
- Where to learn more

Git and the Git logo are either registered trademarks or trademarks of Software Freedom Conservancy, Inc., corporate home of the Git Project, in the United States and/or other countries.

Where to learn more and practice

youtube.com/sasusers



github.com/sascommunities

git-workshop

README.md

Using Git with your SAS projects

This project contains supporting materials for the Git & SAS hands-on workshop presented at PharmaSUG 2022.

Setup and tools

To help you to get the most from this workshop, we recommend that you install at least some of the following tools. You probably won't have all of these tools; select the options that are most interesting to you. We'll demonstrate techniques for all of these and you'll be able to follow along and practice with those that you have.

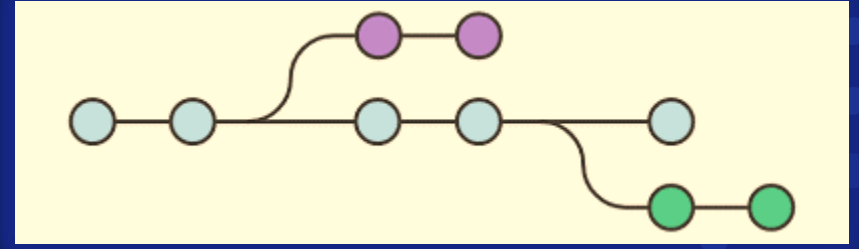
- Git client: Install from <https://www.git-scm.org/>. This may require administrative privileges on your laptop to install.
- GitHub account: free at <http://www.github.com/join>
- SAS OnDemand for Academics account (for some exercises)
- Optional: SAS Enterprise Guide 8.2 or 8.3 with access to a SAS environment (can be local)
- Optional: VS Code -- and optionally with the [new SAS language extension](#)
- Optional: [GitHub Desktop](#) - Git UI specifically for GitHub projects
- Optional: [SAS Studio 3.81 Basic](#) with access to local SAS (requires admin privileges to install/configure)

Materials and exercises

- [Presentation slides \(PDF\)](#)
- [Exercise 01: Create your first Git project](#)
- [Exercise 02: Connect your SAS to Git \(SAS Enterprise Guide\)](#)
- [Exercise 03: Connect your SAS to Git \(SAS Studio\)](#)
- [Exercise 04: Clone an existing SAS code project](#)
- [Exercise 05: Using SAS functions to work with Git](#)
- [Exercise 06: Using VS Code to manage SAS projects](#)

What is Git?

Git is a **distributed version control system**



- **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 MarketShareParam
    
```

Version control


DevOps/automation

GitHub Maintainers > sassoftware.org ...

Conversations Files Wiki +

GitHub 5/1 12:36 PM

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

 kesmit13
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-dlpj
Pull request #: 133

[Add a comment](#) [View in GitHub](#)


← Reply

Collaboration

Start Page covid-19-sas x

master Pull Push


Commit (95) History

Graph	Message	Author	Date	Comm
	new cc templates	Mike Henderson	4/30/2020 15:10:2	9161e
	Merge pull request #6 from sassoftware/Social-Distanc	Mike Henderson	4/30/2020 13:17:2	d4e09
	origin/Social-Distancing-Shifts Adjusted to handle	Mike Henderson	4/30/2020 13:05:1	6ec13
	Added supporting documentation	Chris Hemedinger	4/30/2020 11:45:2	8980c
	rebuild: change in Sigma default in keyword paramete	Mike Henderson	4/30/2020 10:48:3	65969f2
	Add SAS programs to fetch public COVID-19 data	Chris Hemedinger	4/30/2020 10:37:3	ee08ba9
	✓ 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

Branch: — 2s No changes

Commit: — a few seconds ago Started by user Gate Demo078

Start — List Environment variables — Run SAS program — End



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 interface with the following components:

- 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 window shows the 'sas-youtube-api' repository with a commit history table.
- Commit History Table:**

Graph	Message	Author	Date	Commit ID
	✓ main origin/main origin/master Catch up work	saschrh	8/12/2020 21:29:5	e5d63c7
	Merge branch 'master' of gitlab.sas.com:saschrh/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
- Commit Details:** Below the table, it shows 'Author: saschrh', '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_buildvtable.sas', 'yt_sasuserschannel_countries.sas', 'yt_sasuserschannel_groups.sas', and 'yt_sasuserschannel_uploads.sas'.
- Code Editor:** The 'adhocplots.sas' file is open, showing SAS code with a diff view. The code includes:

```
1 diff --git a/adhoc_plots.sas b/adhoc_plots.sas
2 new file mode 100644
3 index 0000000..8d0a130
4 --- /dev/null
5 +++ b/adhoc_plots.sas
6 ----- Difference (@@ -0,0 +1,27 @@) -----
7 +ods html5(id=eghtml) gtitle;
8 +title "SAS User YouTube views";
9 +proc sgplot data=yt.youtube_sasuser_va;
10 + vbar month / response=views dataskin=matte;
11 + where level="CHANNEL";
12 + format views commal2.;
13 + yaxis grid display=(nolabel);
14 + xaxis label="Monthly views";
15 +run;
16 #
```
- Servers:** A sidebar at the bottom left showing a tree view of servers including 'Local', 'SASApp', 'SASApp_TLS', 'SASApp_WFA', 'Special', 'OLAP Servers', and 'Private OLAP Servers'.

Git integration with SAS Studio

The screenshot displays the SAS Studio interface for a Git repository named 'wordle-sas'. The top navigation bar includes 'New', 'Options', 'View', 'Open', and 'Save All'. The left sidebar shows 'Git Repositories' with 'wordle-sas' as the current repository and 'main' as the current branch. The main area shows the commit history for the 'main' branch, listing several commits by Chris Hemedinger. The selected commit message is 'Clean up a few t...'. Below the commit list, the author information and the commit message are displayed, along with a diff view showing changes to the 'wordle-sas.sas' file.

Commit	History	Message	Author	Date	Commit ID
0685848		✓ main origin/main Clean up a few t...	Chris Hemedinger	Mar 18, 2022, 11:10:...	0685848
c8ae053		fix yet another typo in README	Chris Hemedinger	Mar 17, 2022, 2:25:0...	c8ae053
a5844b2		fix another typo in README	Chris Hemedinger	Mar 17, 2022, 9:49:1...	a5844b2
323b187		fix typo in README	Chris Hemedinger	Mar 16, 2022, 8:14:0...	323b187
6d3f74d		add Twitter poll image	Chris Hemedinger	Mar 16, 2022, 3:44:4...	6d3f74d
ea54b2b		change to using DATA step object for g...	Chris Hemedinger	Mar 16, 2022, 2:56:2...	ea54b2b
97babaf		add example showing SASViya	Chris Hemedinger	Mar 16, 2022, 12:52:...	97babaf
f08361a		add more instructions and example ima...	Chris Hemedinger	Mar 16, 2022, 12:39:...	f08361a
d7b73cf		initial commit for Wordle in SAS	Chris Hemedinger	Mar 16, 2022, 10:45:...	d7b73cf

Author: Chris Hemedinger Mar 18, 2022, 11:10:15 AM 0685848427054bdf11fdc680bb66aec58... — □

Clean up a few things I found while writing the blog

Parent: c8ae053

wordle-sas.sas

```
diff --git a/wordle-sas.sas b/wordle-sas.sas index 9c5e857..d2b735c 100644 --- a/wordle-sas.sas +++ b/wordle-sas.sas
@@ -59,7 +59,7 @@ supply 'seed' value to set the word explicitly, good for testing
run;
%end;
data status;
- array check{5} $ 8 checked1-checked5;
+ array check{5} $ 1 checked1-checked5;
length status $ 5;
stop;
run;
@@ -67,6 +67,7 @@ supply 'seed' value to set the word explicitly, good for testing
```


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”](https://support.sas.com) on support.sas.com.

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

Name change: GITFN* → GIT* in SAS Viya

Demo

Connecting SAS to Git

SSH or HTTPS

SAS Studio

Supports SSH or HTTPS

The screenshot shows the 'Edit Profile' dialog in SAS Studio. At the top, there are two tabs: 'SSH' (selected) and 'HTTPS'. The form contains the following fields:

- Profile name: * Chris (GitHub)
- User name: * cjdingner
- Email: * [redacted]
- Public SSH file path: [redacted] /ssh/id_ed25519.pub
- Private SSH file path: [redacted] /ssh/id_ed25519

Buttons for 'OK' and 'Cancel' are at the bottom.

SAS Enterprise Guide 8.2

Supports HTTPS

The screenshot shows the 'Manage Git Connections' dialog in SAS Enterprise Guide. It has two tabs: 'Profiles' and 'Repositories'. Under 'Profiles', 'GitHub' is selected. An 'Edit Profile' sub-dialog is open, showing the following fields:

- Profile name: * GitHub
- User name: * cjdingner
- Email: * [redacted]
- Password: [redacted]

Buttons for 'OK' and 'Cancel' are at the bottom.

SAS GIT* functions

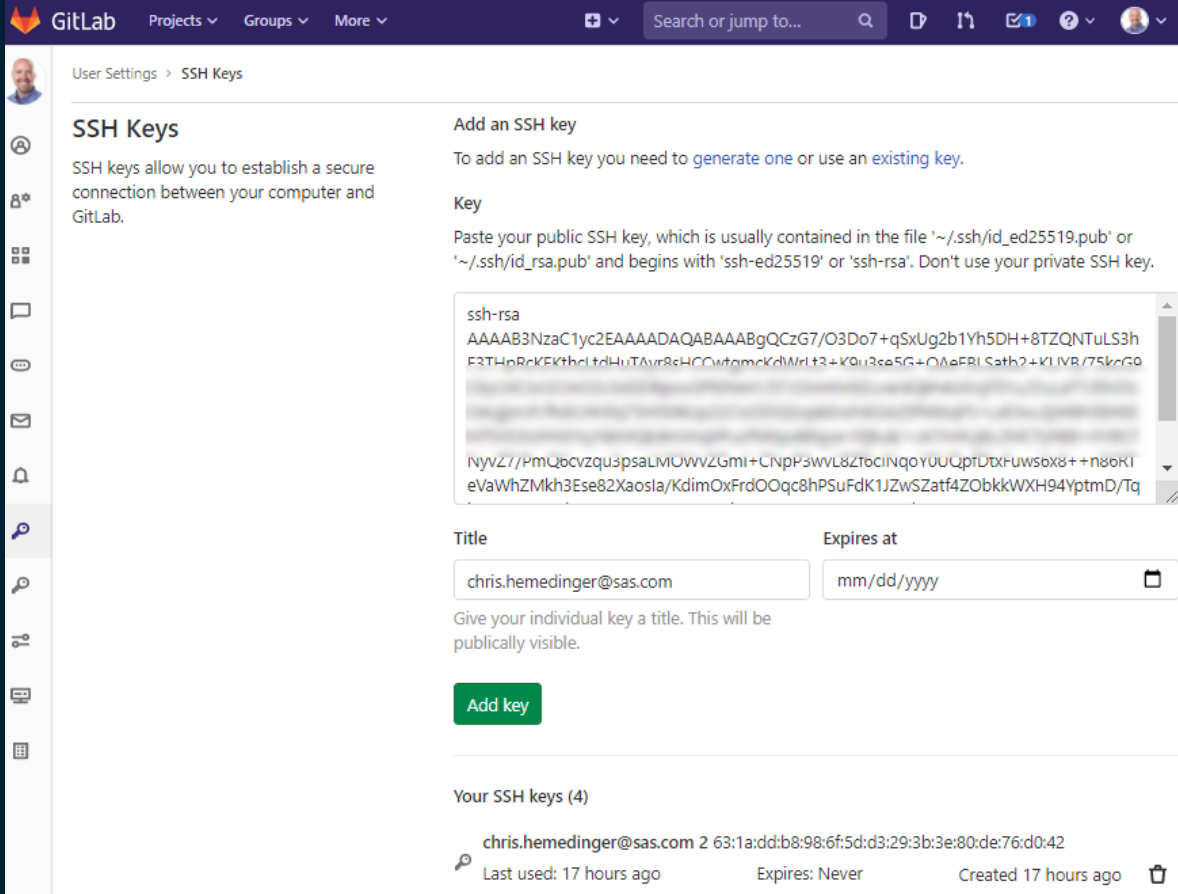
Supports SSH or HTTPS

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

Register SSH key

GitLab example

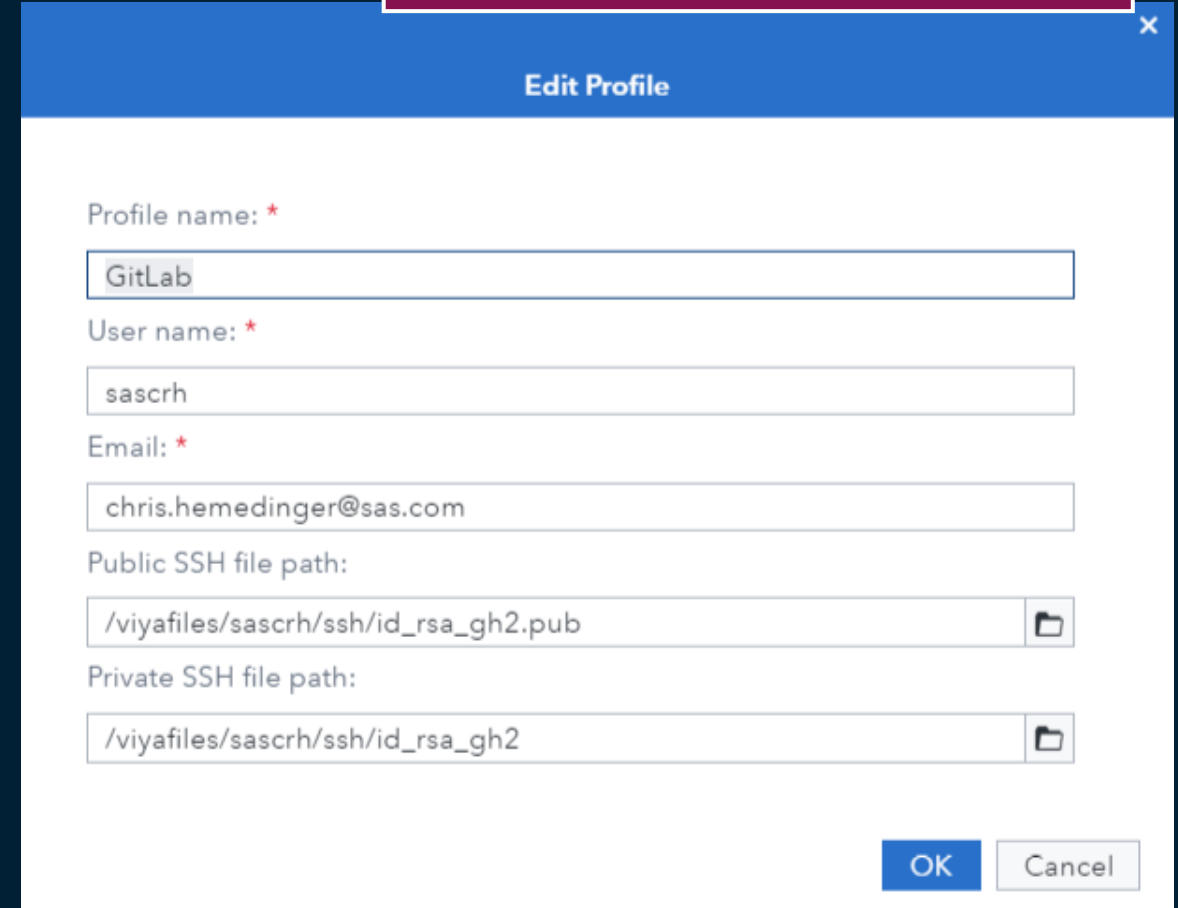
GitLab settings



The screenshot shows the GitLab 'SSH Keys' settings page. It includes a navigation bar with 'GitLab', 'Projects', 'Groups', and 'More'. The main content area is titled 'SSH Keys' and contains instructions on how to add a new SSH key. A text area contains a sample public key starting with 'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCzG7/O3Do7+qSxUg2b1Yh5DH+8TZQNTuLS3h...'. Below the key area are input fields for 'Title' (containing 'chris.hemedinger@sas.com') and 'Expires at' (containing 'mm/dd/yyyy'). A green 'Add key' button is visible. At the bottom, there is a section for 'Your SSH keys (4)' with a table listing existing keys.

SSH Key	Last used	Expires	Created
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 box in SAS Studio. It contains several input fields: 'Profile name:' (containing 'GitLab'), 'User name:' (containing 'saschr'), 'Email:' (containing 'chris.hemedinger@sas.com'), 'Public SSH file path:' (containing '/viyafiles/saschr/ssh/id_rsa_gh2.pub'), and 'Private SSH file path:' (containing '/viyafiles/saschr/ssh/id_rsa_gh2'). At the bottom right, there are 'OK' and 'Cancel' buttons.

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](#)."

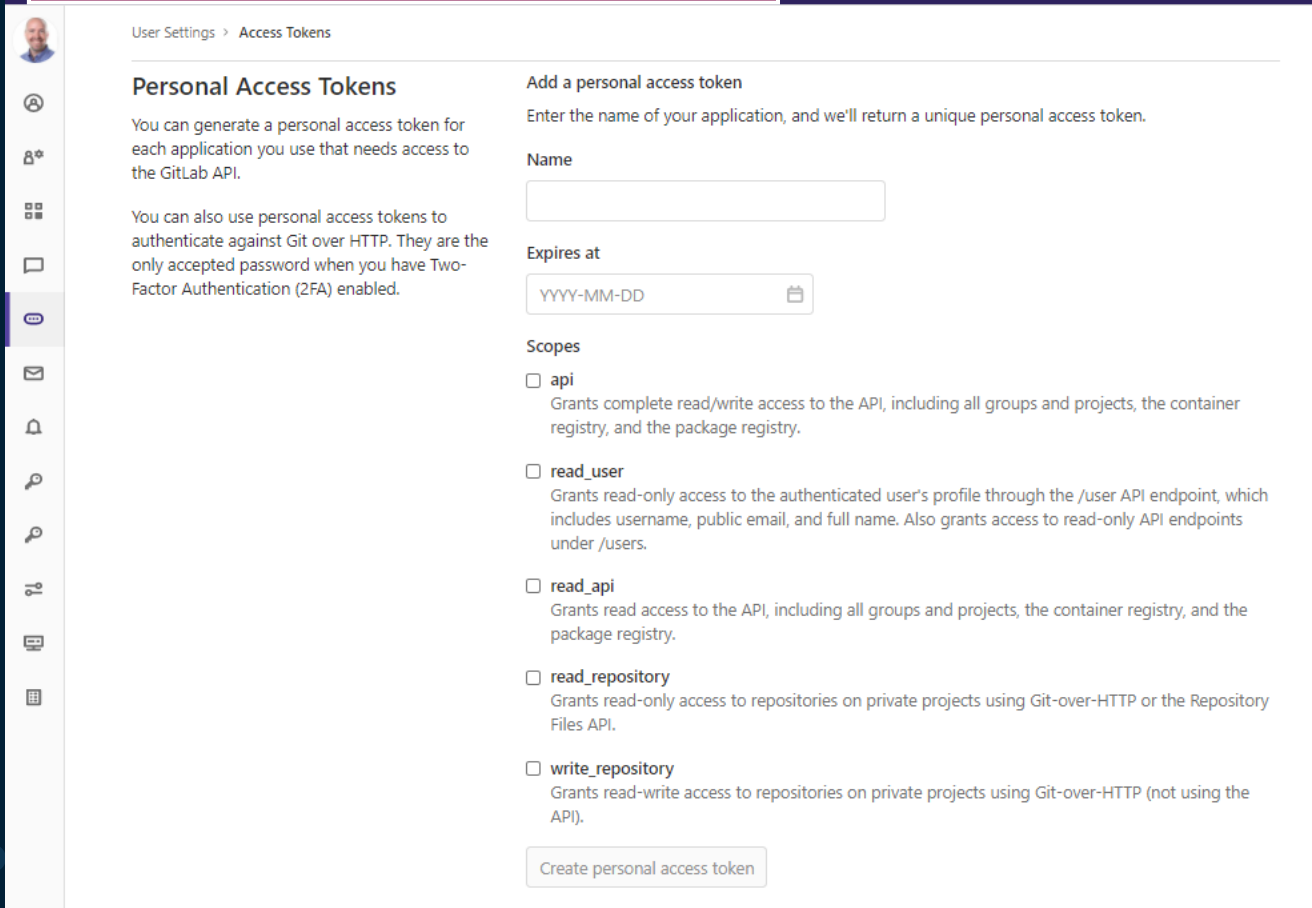
As a security precaution, GitHub automatically removes personal access tokens that haven't been used in a year.

From GitHub doc

Register access token

GitLab example

GitLab settings



The screenshot shows the GitLab 'User Settings > Access Tokens' page. It features a sidebar with navigation icons and a main content area. The 'Personal Access Tokens' section explains that tokens can be generated for applications and used for authentication. Below this, there are instructions on how to use tokens with Two-Factor Authentication (2FA). The 'Add a personal access token' section includes a form with fields for 'Name', 'Expires at' (with a calendar icon), and 'Scopes'. The 'Scopes' section lists several options: 'api', 'read_user', 'read_api', 'read_repository', and 'write_repository', each with a brief description of the permissions granted. A 'Create personal access token' button is located at the bottom of the form.

User Settings > Access Tokens

Personal Access Tokens

You can generate a personal access token for each application you use that needs access to the GitLab API.

You can also use personal access tokens to authenticate against Git over HTTP. They are the only accepted password when you have Two-Factor Authentication (2FA) enabled.

Add a personal access token
Enter the name of your application, and we'll return a unique personal access token.

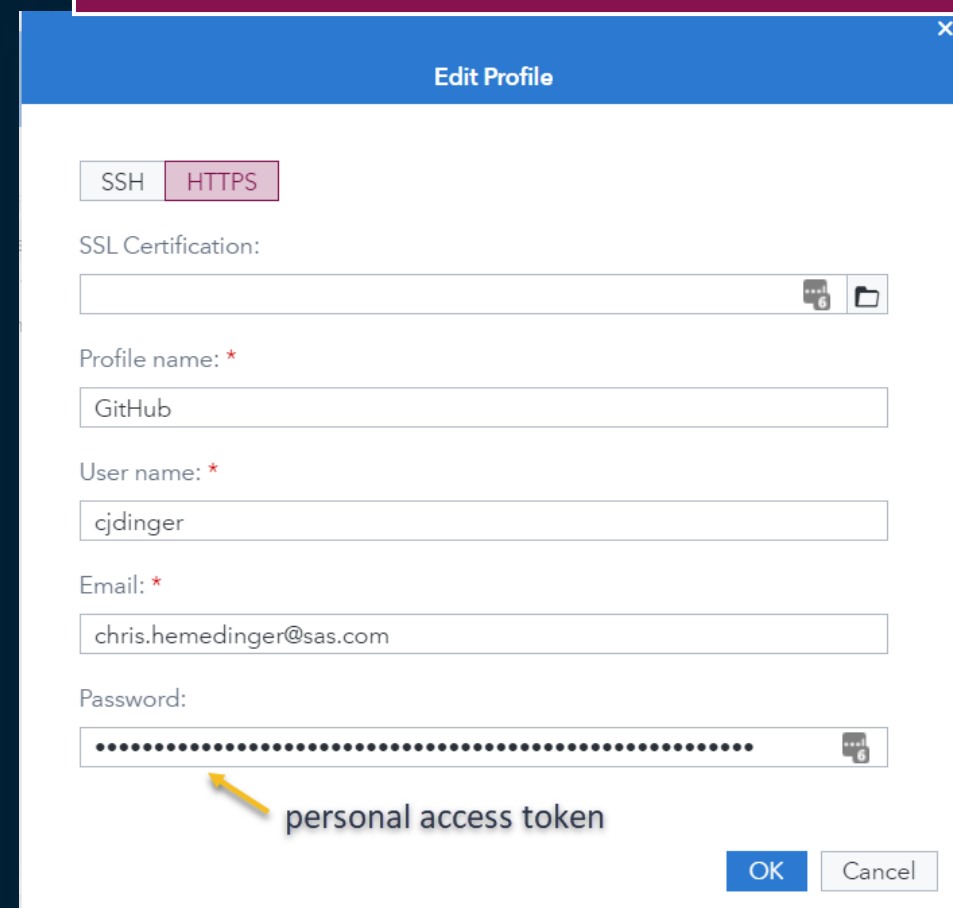
Name

Expires at

Scopes

- api**
Grants complete read/write access to the API, including all groups and projects, the container registry, and the package registry.
- read_user**
Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.
- read_api**
Grants read access to the API, including all groups and projects, the container registry, and the package registry.
- read_repository**
Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.
- write_repository**
Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).

SAS Studio settings



The screenshot shows the 'Edit Profile' dialog in SAS Studio. It has a blue header bar with the title 'Edit Profile'. Below the header, there are several input fields. At the top, there are two tabs: 'SSH' and 'HTTPS', with 'HTTPS' selected. Below the tabs is an 'SSL Certification' field with a file selection icon. The 'Profile name' field contains 'GitHub'. The 'User name' field contains 'cjdingner'. The 'Email' field contains 'chris.hemedinger@sas.com'. The 'Password' field is masked with dots and has a file selection icon. A yellow arrow points to the password field with the text 'personal access token'. At the bottom right, there are 'OK' and 'Cancel' buttons.

Edit Profile

SSH HTTPS

SSL Certification:

Profile name: *

User name: *

Email: *

Password:

personal access token

Use other tools alongside SAS

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

The image displays three overlapping screenshots illustrating the use of development tools alongside SAS:

- Terminal Window (MINGW64):** Shows the execution of `git status` in a terminal. The output indicates the current branch is `main` and is up to date with `origin/main`. It also lists changes not staged for commit, suggesting the use of `git add`.
- VS Code Editor:** Shows the `README.md` file for the `sas-zip-macros` repository. The content includes a title "# FILENAME ZIP in SAS", a reference to "The SAS Dummy blog", and a section for "ZIPpy details: a solution".
- Git GUI Window:** Shows a file browser for the `sas-zip-macros` repository, listing files like `README.md`, `zipfiles_list_details.sas`, and `zipfiles_list_extract.sas`.

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.

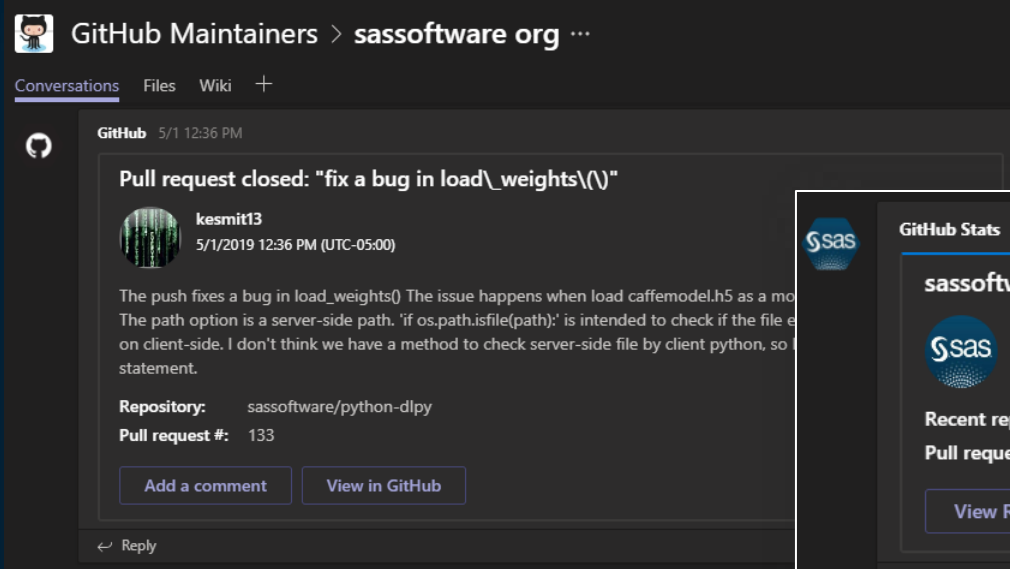
Many SAS products support direct integration with Git. This includes SAS Studio, SAS Enterprise Guide, and the SAS programming language. (That last one checks a **lot** of boxes for ways to use Git and SAS together.) While we have good documentation and videos to help you learn about Git and SAS, we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

<https://blogs.sas.com/content/sasdummy/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




GitHub Maintainers > sassoftware org ...

Conversations Files Wiki +

GitHub 5/1 12:36 PM

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

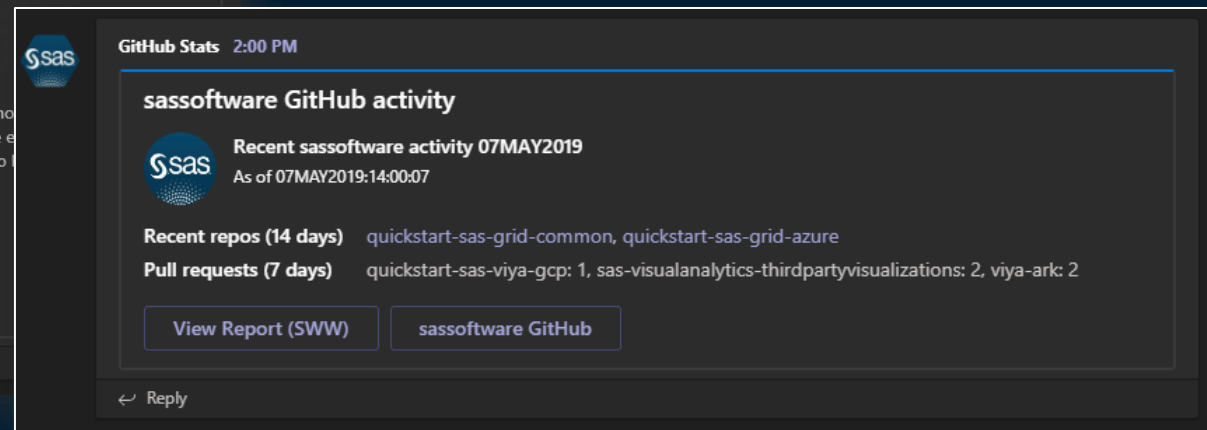
 kesmit13
5/1/2019 12:36 PM (UTC-05:00)

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

Repository: sassoftware/python-dlpy
Pull request #: 133


[Add a comment](#) [View in GitHub](#)

← Reply



GitHub Stats 2:00 PM

sassoftware GitHub activity

 **Recent sassoftware activity 07MAY2019**
As of 07MAY2019:14:00:07

Recent repos (14 days) quickstart-sas-grid-common, quickstart-sas-grid-azure
Pull requests (7 days) quickstart-sas-viya-gcp: 1, sas-visualanalytics-thirdpartyvisualizations: 2, viya-ark: 2

[View Report \(SWW\)](#) [sassoftware GitHub](#)

← Reply

Learn more

- [Using Git with SAS](#) (SAS Users YouTube)
- [Git & SAS Workshop](#) (GitHub project)
- [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



Q & A

sas.com

