

# Importing a Team Foundation Version Control repository

## In this article

About imports from Team Foundation Version Control

Migrating with Azure Repos (recommended)

Migrating with git-tfs

You can import a repository from Team Foundation Version Control (TFVC) by converting the repository to Git, then pushing the Git repository to GitHub Enterprise Server.

## About imports from Team Foundation Version Control

We recommend using Azure Repos to convert your Team Foundation Version Control (TFVC) repository to Git. However, Azure Repos only migrates up to 180 days of history. If you wish to retain more history, you can use `git-tfs` instead.

## Migrating with Azure Repos (recommended)

**Note:** Azure Repos only migrates up to 180 days of history. The state of your repository before the 180-day threshold will be migrated in a single initial commit. To retain more history, see "[Migrating with `git-tfs`](#)."

To follow these steps, you must use a macOS or Linux system and have the following tools installed:

- [Git](#)
  - Git Large File Storage (Git LFS) (see "[Installing Git Large File Storage](#)")
- 1 Create a new repository on your GitHub Enterprise Server instance. To avoid errors, do not initialize the new repository with README, license, or gitignore files. You can add these files after your project has been pushed to GitHub Enterprise Server. For more information, see "[Creating a new repository](#)."
  - 2 To confirm that Git is installed on your machine, run `git --version`.  
The output should be similar to `git version 2.40.0`.
  - 3 To confirm that Git LFS is installed on your machine, run `git lfs --version`.  
The output should be similar to `git-lfs/3.1.4 (GitHub; darwin arm64; go 1.18.1)`.
  - 4 Convert your TFVC repository to Git using Azure Repos. For instructions, see [Import repositories from TFVC to Git](#) in Microsoft Learn.

- 5 To clone your Azure Repos repository to your local machine, run `git clone --mirror URL` , replacing URL with the clone URL for your Azure DevOps repository.

To identify the clone URL for your Azure DevOps repository, see [Get the clone URL of an Azure Repos Git repo](#) in Microsoft Learn.

- 6 To add your GitHub repository as a remote, run `git remote add origin URL` , replacing URL with the URL for the GitHub repository you created earlier, such as `https://github.com/octocat/example-repository.git` .

- 7 To push the repository to GitHub, run `git push --mirror origin` .

If your repository contains any files that are larger than GitHub Enterprise Server's file size limit, your push may fail. Move the large files to Git LFS by running `git lfs import` , then try again.

## Migrating with `git-tfs`

If you migrate with `git-tfs` , you will retain the full history of your TFVC repository.

To follow these steps, you must use Windows and have the following tools installed:

- [Visual Studio Team Explorer](#)
- [git-tfs](#) , installed using Chocolatey or by downloading the binary release manually
- [Git](#)
- Git Large File Storage (Git LFS) (see "[Installing Git Large File Storage](#)")

- 1 Create a new repository on your GitHub Enterprise Server instance. To avoid errors, do not initialize the new repository with README, license, or gitignore files. You can add these files after your project has been pushed to GitHub Enterprise Server. For more information, see "[Creating a new repository](#)."

- 2 To confirm that Git is installed on your machine, run `git --version` .

The output should be similar to `git version 2.40.0` .

- 3 To confirm that `git-tfs` is installed on your machine, run `git tfs --version` .

The output should start with something similar to `git-tfs version 0.32.0.0 (TFS client library 16.0.0.0 (MS))` .

If you get an output similar to `Unable to load DLL`

`'Microsoft.VisualStudio.Setup.Configuration.Native.dll': The specified module could not be found. (Exception from HRESULT: 0x8007007E)` , make sure that Visual Studio Team Explorer is correctly installed.

- 4 To confirm that Git LFS is installed on your machine, run `git lfs --version` .

The output should be similar to `git-lfs/3.1.4 (GitHub; darwin arm64; go 1.18.1)` .

- 5 If you haven't already, configure Git with your name and email address by running `git config --global user.name "NAME"` and `git config --global user.email EMAIL` , replacing NAME with your name and EMAIL with your email address.

- 6 Run `git tfs clone` , passing your TFVC repository's URL and repository path as arguments. For example, to convert the `example` repository from `https://dev.azure.com/octocat` into a Git repository stored in the `/example` directory, run `git tfs clone --branches=all https://dev.azure.com/octocat $/example` .

- 7 Move into the directory for the newly-created Git repository.
- 8 To add your GitHub repository as a remote, run `git remote add origin URL` , replacing `URL` with the URL for the GitHub repository you created earlier, such as `https://github.com/octocat/example-repository.git` .
- 9 To push the repository to GitHub, run `git push --mirror origin` .

If your repository contains any files that are larger than GitHub Enterprise Server's file size limit, your push may fail. Move the large files to Git LFS by running `git lfs import` , then try again.

## Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)