

Explore how to move git repositories between Team Projects with full-fidelity history

04/26/2018 • 3 minutes to read • 

In this article

[What's the scenario?](#)

[How do I move?](#)



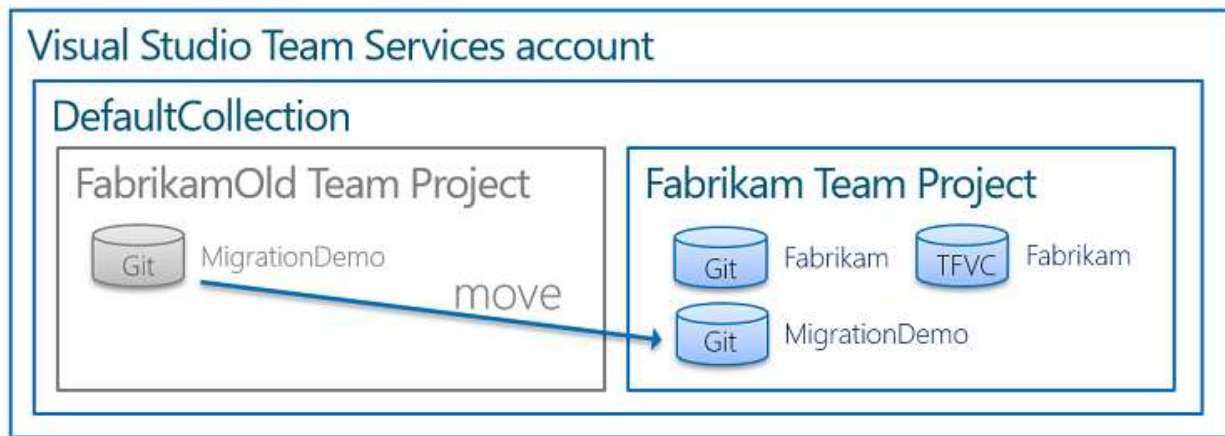
If you're planning to consolidate multiple Team Projects into one, you're probably wondering:

- What to do with all the repositories?
- Move or merge them?
- Keep history or just the tip of the iceberg?

In this article you will learn how to move your Git repositories to another team project, with full-fidelity history.

What's the scenario?

As shown, you need to move the MigrationDemo repo, from the FabrikamOld to the new Fabrikam team project.



How do I move?

You have two options as outlined below. Import functionality is easier, but is only available in Azure DevOps Services and TFS 2017 Update 1 and above.

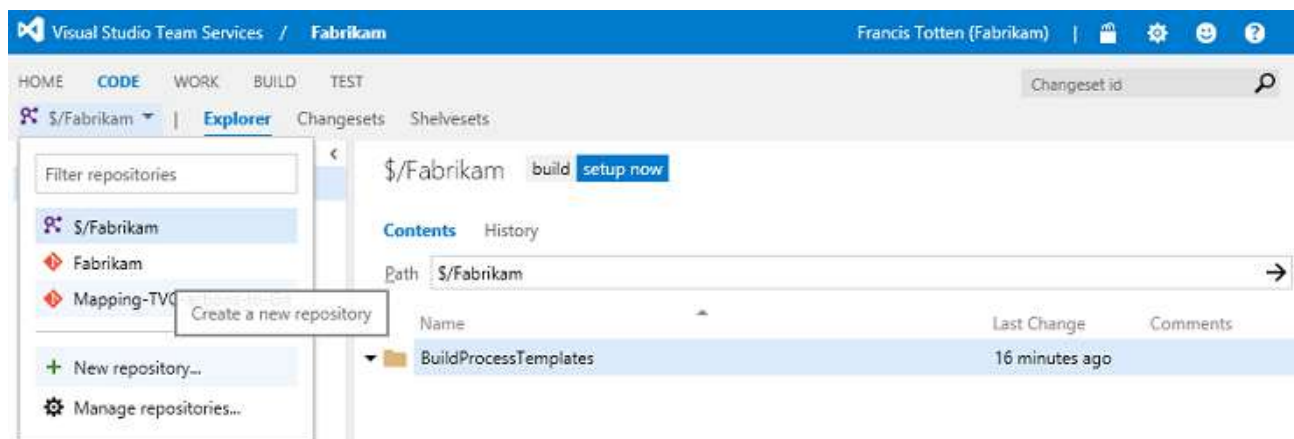
Use Import Git repository functionality

Using the Import Repository feature, you can import a Git repository to your team project from Team Foundation Server (TFS), Azure Repos or any other Git source code provider like GitHub. Review the [import repository documentation](#) for more details.

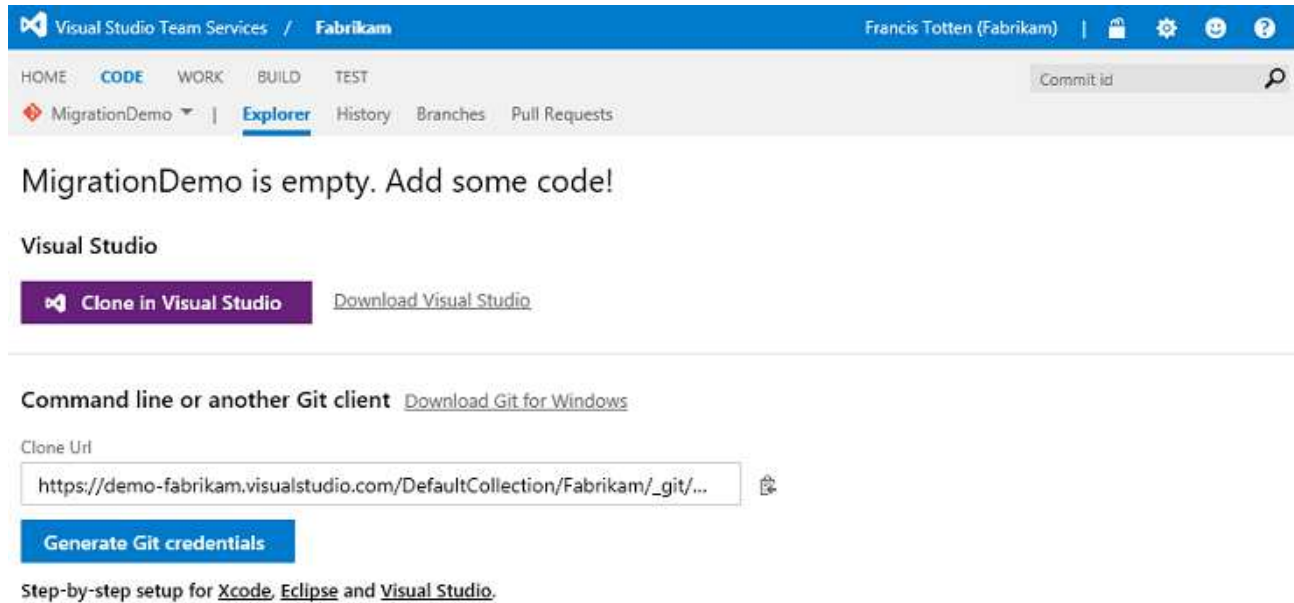
Manually migrate the Git repo in five easy steps:

Create an empty Git repo.

From the code explorer, click on the repo name. Choose **New Repository** from the list, select Git as the type, and give it a name.



Once the repo is created, you are presented with step-by-step instructions to quickly get started. Copy the `clone` URL to your clipboard.



Important - Clear the **Automatically create links for work items mentioned in a commit comment** if you are importing from a different project collection or a foreign git repository. Azure DevOps will otherwise associate the commits to existing work items of unrelated team projects in the team project collection.

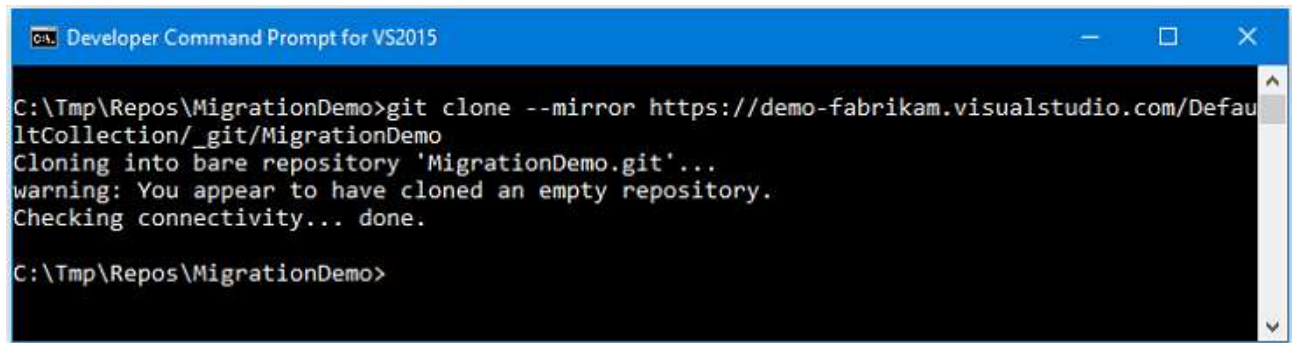


Mirror the repository

Switch to a Developer Command Prompt and path to your local (source) repository for the MigrationDemo repo in FabrikamOld. Run the `git clone --mirror` command, using the Clone URL from above.

Command Line: `git clone --mirror https://demo-fabrikam.visualstudio.com/DefaultCollection/Fabrikam/_git/MigrationDemo`

As shown, the `clone --mirror` is redundant in this case, as the remote repository is bare. It is used here as a safe and easy way to set up the remote.

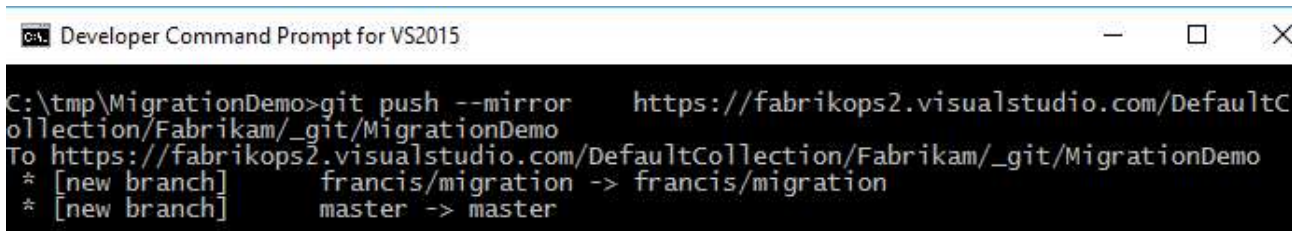


```
C:\Tmp\Repos\MigrationDemo>git clone --mirror https://demo-fabrikam.visualstudio.com/DefaultCollection/_git/MigrationDemo
Cloning into bare repository 'MigrationDemo.git'...
warning: You appear to have cloned an empty repository.
Checking connectivity... done.

C:\Tmp\Repos\MigrationDemo>
```

Push the repo

Run the `git push` command to push the local changes to the remote (target) repo.

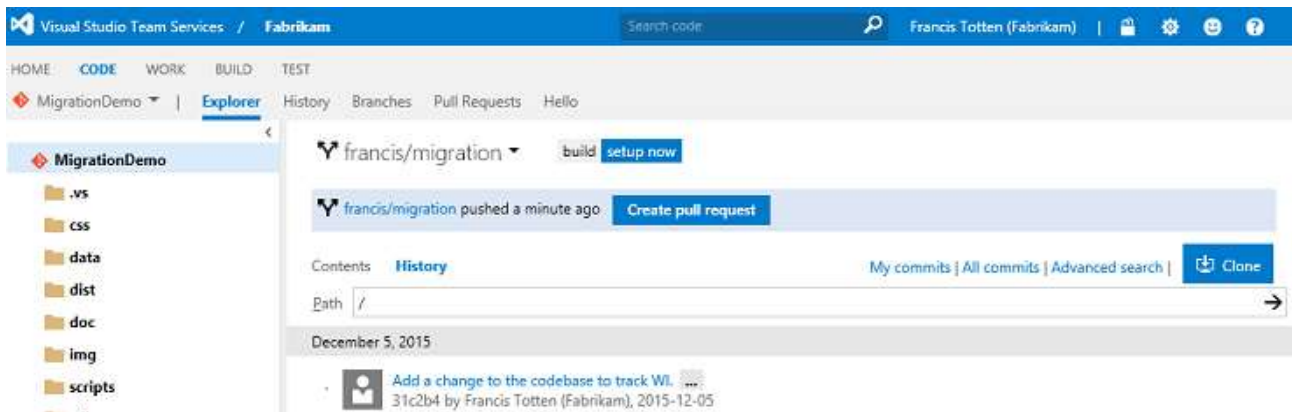


```
C:\tmp\MigrationDemo>git push --mirror https://fabrikops2.visualstudio.com/DefaultCollection/Fabrikam/_git/MigrationDemo
To https://fabrikops2.visualstudio.com/DefaultCollection/Fabrikam/_git/MigrationDemo
 * [new branch]      francis/migration -> francis/migration
 * [new branch]      master -> master
```

The `--mirror` option is used with both the clone and push command. The option ensures that all branches and other attributes are replicated in the new repo.

Validate the new repository

Switch to the Azure DevOps web portal and validate the new repository and the history in the **CODE** hub.



Verify that all your branches were moved over to the new repo.

Configure the new repo

Verify that the permissions and policies are correctly configured for the new repo. You can configure the security after step 1, or at this stage. Reconfigure your builds to connect with the new repo. Lastly, notify users of the original repo to update their remotes in Visual Studio, or running the `git remote set-url origin` command.

```
Command line: git remote set-url origin https://demo-  
fabrikam.visualstudio.com/DefaultCollection/Fabrikam/_git/MigrationDemo
```

Important - Remember to clean up the original project by either deleting the repo (be careful, there's no undo) or locking the branches so that no one accidentally keeps updating it.

For detailed information on planning your team project collections and team projects, refer to the [TFS Planning, Disaster Avoidance and Recovery, and TFS on Azure Iaas Guide](#).

Authors: Jesse Houwing, Mike Fourie, and Willy Schaub | Connect with the authors and ALM | DevOps Rangers [here](#)

(c) 2016 Microsoft Corporation. All rights reserved. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

Is this page helpful?

 Yes  No
