Azure DevOps
Version Controlling with Git

Eng Teong Cheah & Eng Soon Cheah

Microsoft MVP for Developer Technologies

Source Control?

What is Source Control

A source control system, also called a version control system, allows developer to collaborate on code and track changes.

Source control is an essential tool for multi-developer projects.

Benefits of Source Control?

Benefits of Source Control

- Create workflows
- Work with versions
- Code together
- Keep a history
- Automate tasks

Types of Source Control Systems

Git: distributed version control

Git is the most commonly used version control system today and is quickly becoming the standard for version control.

Git is a distributed version control system, meaning that your local copy of code is a complete version control repository.

The fully functional local repositories make it is easy to work offline or remotely.

You commit your work locally, and then sync your copy of repository with the copy on the server.

Team Foundation Version Control (TFVC): centralized version control

Typically, team members have only one version of each file on their dev machines.

Historical data is maintained only on the server.

Branches are path-based and created on the server.

Migrate from TFVC to Git

Migrate from TFVC to Git

Before you try to migrate source code from a centralized version control system to Git, be sure that you familiarize yourself with the differences between centralized and distributed version control systems, and plan your team’s migration. After you’ve prepared, you can begin the migration

Requirements Migrate from TFVC to Git

In order to make migrations simple, there are a number of requirements on the TFVC Import tool:

1. Only a single branch is migrated.
2. A “tip migration”, importing only the latest version of the source code, is suggested.
3. You do not have binary assets like images, scientific data sets, or game models in your repository.
4. The imported repository cannot exceed 1GB in size.

Authenticate the Git Credential Manager

Authenticate with the Git Credential Manager

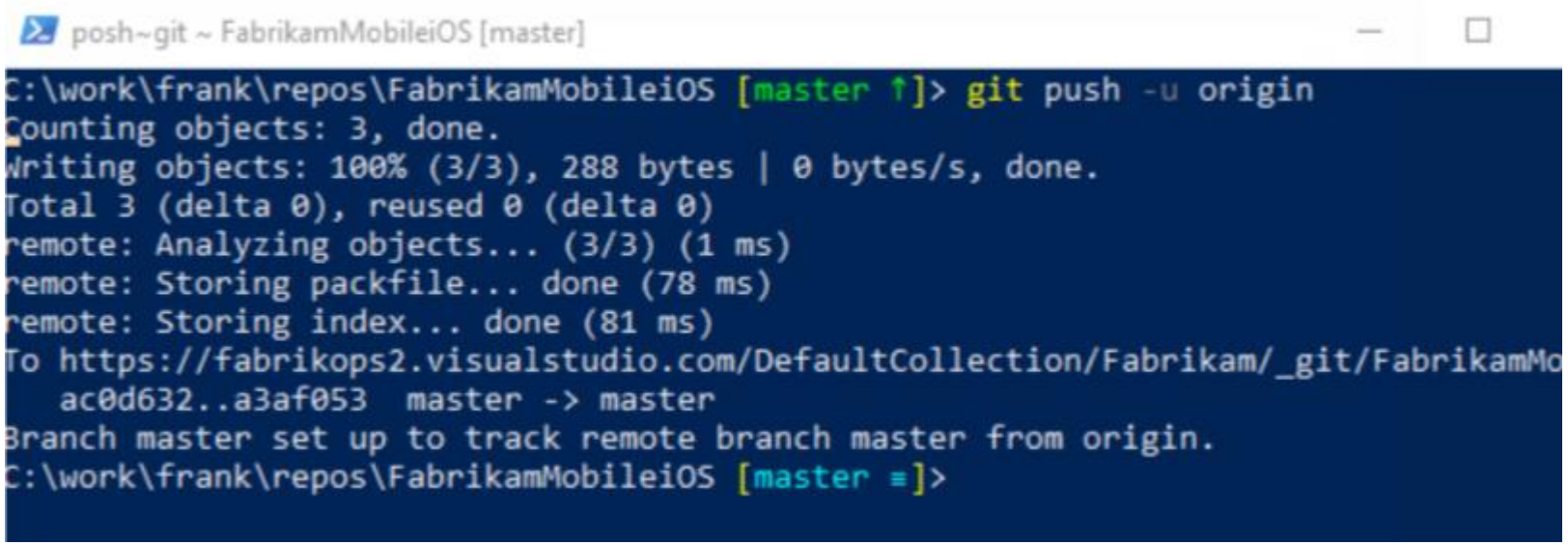
Git Credential Manager simplify authentication with your Azure DevOps Services/TFS Git repos.

Credential Managers let you use the same credentials that you use for the Azure DevOps Services/TFS web portal and support multi-factor authentication through Microsoft Account (MSA) or Azure Active Directory(Azure AD).

Authenticate with the Git Credential Manager

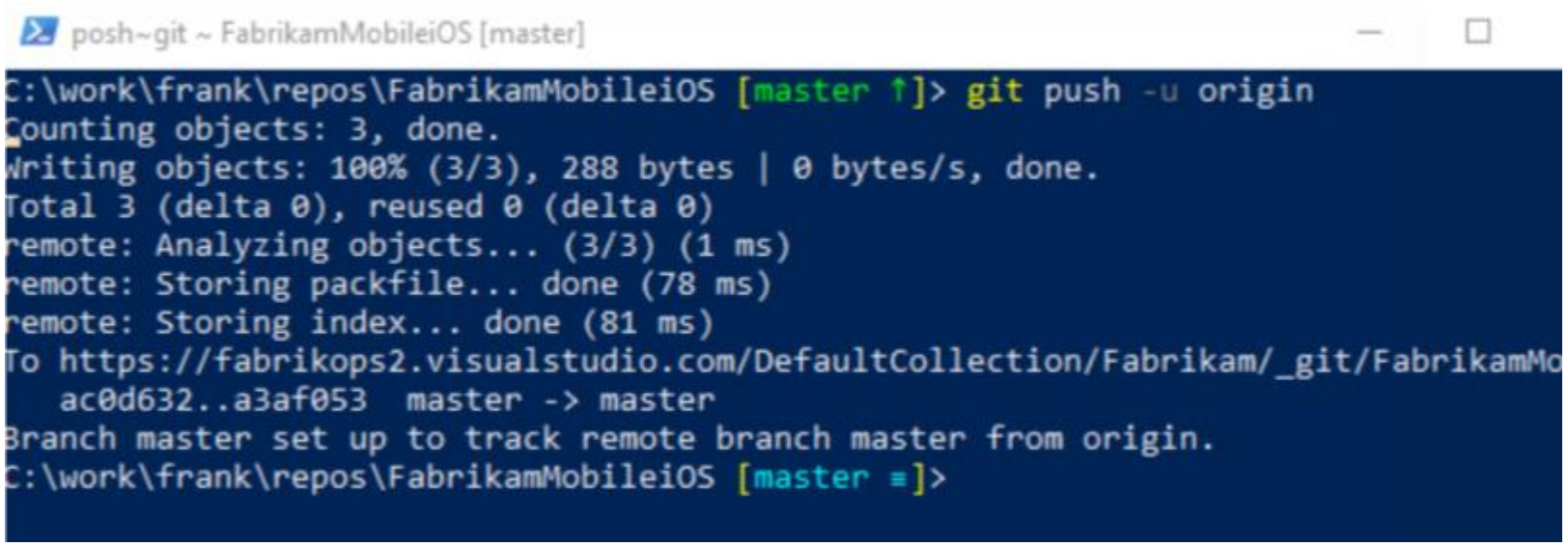
In addition to supporting multi-factor authentication with Azure DevOps Services, the credential managers also provide support two-factor authentication with GitHub repositories.

Authenticate with the Git Credential Manager



When you connect to a Git repo in Azure Repos from your Git client for the first time, the credential manager prompts for your Microsoft Account or Azure Active Directory credentials.

Authenticate with the Git Credential Manager



Once authenticated, the credential manager creates and caches a personal access token for future connections to the repo.

Demo: Version Controlling with Git

Reference

- docs.microsoft.com