<https://nvie.com/posts/a-successful-git-branching-model/>

Git stores history as a snapshot of the repository in time, while TFVC records the discrete operations that occurred on a file. Change types in TFVC like rename, undelete and rollback cannot be expressed in Git; instead of seeing that file A was renamed to file B, you may only see that file A was deleted and file B was added in the same commit

TfsMigrator assists with readying a Team Foundation Server collection for import to Azure DevOps Services.

-------------------------------------

Commands

-------------------------------------

1) Validate - Validates that a collection is eligible to be imported to Azure DevOps Services and verifying changes made to the identity mapping file generated from the prepare command.

> TfsMigrator Validate /collection:CollectionURI [/connectionString:sql connection string] [/skip:step1,step2,...] [/o

utput:path] [/tenantDomainName:name] [/region:region]

2) Prepare - Generates files required to import the specified collection. Before generating any files prepare runs a ful

l validate. If it fails then no files will be generated.

> TfsMigrator Prepare /collection:CollectionURI /tenantDomainName:name /region:region [/connectionString:sql connectio

n string] [/importFile:name] [/output:path]

3) Import - Queues an import to Azure DevOps Services. Can also be used to verify import before running.

> TfsMigrator Import /importFile:import.json [/validateOnly]

-------------------------------------

Command Help

-------------------------------------

Learn more about each command and see examples by viewing their help text:

> TfsMigrator [command] /help

For example, to view the help text for Validate:

> TfsMigrator Validate /help

-------------------------------------

Collection

-------------------------------------

TfsMigrator commands will be run against a specified collection that is online only.

/collection:http://ServerName:port/VirtualDirectoryName/CollectionName

-------------------------------------

Connection String

-------------------------------------

If you're unable to run TfsMigrator directly on the Application Tier, set this option to validate from a remote server b

y specifying a connection string to the configuration database.

/connectionString:"Data Source=sqlserver;Initial Catalog=Tfs\_Configuration;Integrated Security=True"

-------------------------------------

Guidance

-------------------------------------

If this is your first time using TfsMigrator to prep an import, it's recommended that you do the following:

1) Run a validate against the collection(s) you want to import

2) Validate will provide guidance, but errors will be need to fixed and validate re-run before moving on

3) After a successful validate you will want to run a prepare on an online collection to generate necessary import fil

es

4) Review the generated files, and if necessary, make changes to ready them for use in an import

5) If you made any changes to the generated files it's highly recommended that you run them through the validate comma

nd

6) Run validate one last time before taking a collection offline for import to ensure everything is OK