



FREE eBook

LEARNING

Team Foundation

Server

Free unaffiliated eBook created from
Stack Overflow contributors.

#tfs

Table of Contents

About.....	1
Chapter 1: Getting started with Team Foundation Server.....	2
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
What is TFS and how Data is stored in it?.....	2
Chapter 2: TF GET (command-line).....	5
Parameters.....	5
Remarks.....	5
Examples.....	6
Getting Files.....	6
Get files, even if TFS believes the workspace is upto date.....	6
Credits.....	7

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [team-foundation-server](#)

It is an unofficial and free Team Foundation Server ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Team Foundation Server.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with Team Foundation Server

Remarks

This section provides an overview of what tfs is, and why a developer might want to use it.

It should also mention any large subjects within tfs, and link out to the related topics. Since the Documentation for tfs is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

Detailed instructions on getting tfs set up or installed.

What is TFS and how Data is stored in it?

Team Foundation Server (commonly abbreviated to TFS) is a Microsoft product that provides source code management (either via Team Foundation Version Control or Git), reporting, requirements management, project management (for both agile software development and waterfall teams), automated builds and lab management, testing and release management capabilities. It covers the entire application lifecycle. TFS can be used as a back end to numerous integrated development environments but is tailored for Microsoft Visual Studio and Eclipse.

ADVANTAGES:

1. Team Foundation Server provides a set of collaboration tools that work with your existing IDE or editor, so your team can work effectively on software projects of all shapes and sizes.
2. Store and collaborate on code with unlimited private repositories. Use Git for distributed version control to maximize collaboration or use Team Foundation version control (TFVC) for centralized version control.
3. Support for AGILE methodology
4. Support for multiple languages and IDE
5. Allows third party plugin integration

TYPES of TFS:

1. Online
2. On-premises

Online is backed by Microsoft's cloud platform, Windows Azure and it does not require any setup. A user logs in using a Microsoft Account to begin setting up their account, creating projects and adding team members. New features developed in three-week development cycles are added to the online version first. These features migrate to the on-premises version as updates, at

approximately three-month intervals.

Team Foundation Server stores all the changeset data in a SQL Server Database. It stores the code from the most recent changeset in its entirety. It then stores a diff to the previous version. One of the benefits of storing it all in SQL Server is that it gains the "all or none" saving capability that is provided by transactions. The architecture of TFS is centralized. This means all source code is maintained at a single location. In a centralized architecture the TFS server itself can be considered a single point of failure, but with high-availability solutions available in the Windows Server operating system, this does not need to be so. Similarly, the SQL Server database storing the actual source code bits can be mirrored on multiple servers. TFS control has been designed to integrate seamlessly with the latest versions of Microsoft Visual Studio. However, this does not mean that you could not use TFS version control with other software development products. The functionality in TFS can be divided into the following areas. Basic functionality - checking files in and out Locking - limiting concurrent edits Branching and merging - work with different versions of the source code Security - decide who can access the version control data and how

The basic functionality in any version control system includes checking file in and out. To support concurrency, TFS allows multiple checkouts of the same file, but this can be disabled should the need arise. Items can also be exclusively locked so that nobody else can check in or out a file while it is locked. If concurrent checkouts are disabled in team project settings, then a lock is automatically placed on the file upon checkout. Branching and merging can be considered advanced functions in TFS, but nonetheless, they are highly useful. The main idea of branching is to take a set of source code files and create a distinct version from those files. The branched code can live a life of its own separate from the original source files. For instance, if you are developing a generic application but need to make a specialized version for a certain customer, you could branch the customer customizations from the main source control tree (the "trunk"). Should the need arise later, you can again combine the customization code with the original source control line. This is called merging. Everything in TFS (except Active Directory user rights version control) are stored in a central SQL Server database. This includes team project settings, file data, changeset details, and so on. Because almost everything is in a central location, it is imperative to make sure you take regular backups of the SQL database(s) and have a disaster recovery plan. To understand how version control is integrated into Microsoft Visual Studio, you need to be aware of three separate windows (or panes, depending on your point of view): the Source Control Explorer, Solution Explorer and Pending Changes windows. When you have connected to a Team Foundation Server instance, Solution Explorer will allow you to check out and check in files directly from the window by right-clicking the items. However, by default a check out occurs simply when you start editing a file in Visual Studio. For instance, if you open a C# source code file in the editor and start typing, the file is checked out. This is often the most convenient way to check out files. Small icons shown by the Solution Explorer window help you distinguish between locked, checked out and added files, and so forth. A blue lock icon indicates that a file is part of source control but is not currently checked out. A red check mark indicates that the file has been checked out, and a yellow plus sign indicates that a file has been added to the project. In TFS, a check out operation and the operation to get the latest version are separate from each other. In practice, this means that before checking out a file, you should execute a "Get Latest" command on the file(s) you wish to check out. This can be done by simply right-clicking an item in Solution Explorer, and choosing the Get Latest menu item. To change this default behavior, you can choose Microsoft Visual Studio's Tools/Options menu command, and navigate to the section Source Control/Visual Studio

Team Foundation Server. From here, you can find an option named "Get latest version of item on check out". The source control window allows you to get a more holistic view of your version control tree. Below is the source control explorer window (currently disconnected).

To open the window, choose the View/Other Windows/Source Control Explorer menu command, or double-click the Source Control node in Visual Studio's Team Explorer window. The Source Control Explorer window allows you to view and manipulate files in your version control tree. You can do all the same operations through this window than you could do in Solution Explorer: for instance, you can check in and out files, rename them, delete them, and so on. The difference is that using the Source Control Explorer window, the files you work with do not need to be part of a Visual Studio development project, such as a C# project. For example, you could add an Excel spreadsheet to version control; this is something that you might not want to do through Solution Explorer. Whenever you work with files under version control and edit, add or delete them, Visual Studio will not immediately commit the changes back to version control. Instead, this is done only when you check the changes in. In the meantime, all your changes are by default stored in a pending changes list, which can be seen through the Pending Changes window.

The Pending Changes window shows a list of files that have not been checked in. The window also indicates the operation (add, edit, delete or rename) requested. Usually, you do your check-ins through this window, since it allows you to conveniently check in multiple files in a single operation. You can also write a comment to accompany the checked in files, and you can link to a Team Foundation Server work item with the files. Overall, one or more source files, optional comments and work item associations collectively form a changeset. A changeset in TFS is always checked in atomically, which means that the complete set either succeeds or fails in the check in. A changeset is associated with a unique ID, and can be later viewed for example through the Source Control Explorer window. In TFS version control, a changeset is an essential concept because it is the smallest set of changes that the system can process. A changeset can contain a single file, or a set of files. Furthermore, it is the basis of reporting, especially when used together with work items.

Read [Getting started with Team Foundation Server](https://riptutorial.com/tfs/topic/2104/getting-started-with-team-foundation-server) online:

<https://riptutorial.com/tfs/topic/2104/getting-started-with-team-foundation-server>

Chapter 2: TF GET (command-line)

Parameters

Parameter	Details
/version	The version to retrieve. View remarks .
/all	Forces all files to be retrieved, not just those that are out-of-date.
/overwrite	Overwrites writable files that are not checked out
/force	Combines <i>/all</i> and <i>/overwrite</i>
/preview	Displays what would occur, without actually performing the Get operation.
/recursive	Recursively retrieves all items that match your itemspec.
/noprompt	Suppresses any dialog boxes that would otherwise be displayed during this operation.
/remap	Updates local mapping on the server and re-downloads the file faster if you already have downloaded the same file from another branch.
/login	Specifies the user name and password to authenticate the user with Team Foundation Server.

Remarks

1. The parameter **/version** assumes **Latest** if no parameter provided and accepts
 - a.Changesets ("C<ChangeSetID>")
 - b.Date/Time ("D<DateTime_UTC>")
 - c.Label ("L<Label>")
 - d.Latest Version ("T")
 - e.Workspace Version ("W<workspacename;owner>").
2. Parameters as of **TFS 2010**
3. Retrieved and edited from [https://msdn.microsoft.com/en-us/library/fx7sdeyf\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/fx7sdeyf(v=vs.100).aspx) at 2016-07-22 14:48UTC

Examples

Getting Files

```
tf get /all /recursive $/
```

Gets *and* replaces all files and directories from the last version available

Get files, even if TFS believes the workspace is upto date

```
tf.exe get /all /recursive /force /overwrite $/
```

Gets all files and directories, and will replace any files that are not currently checked out in the workspace (including files that have been edited).

Read TF GET (command-line) online: <https://riptutorial.com/tfs/topic/3054/tf-get--command-line->

Credits

S. No	Chapters	Contributors
1	Getting started with Team Foundation Server	Community , Jared Hooper , Nagaraj Raveendran
2	TF GET (command-line)	Eric Wu , pm_2