

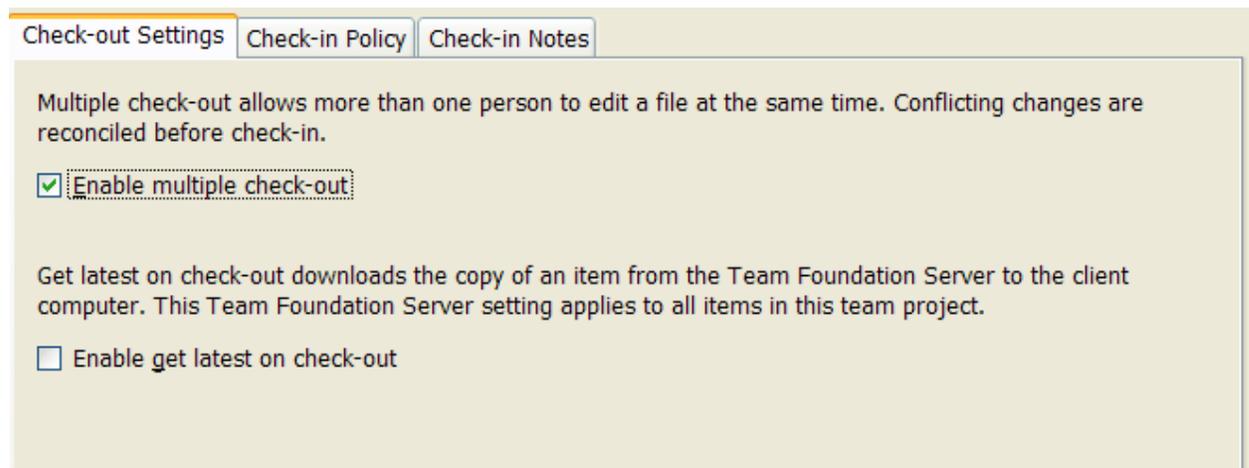
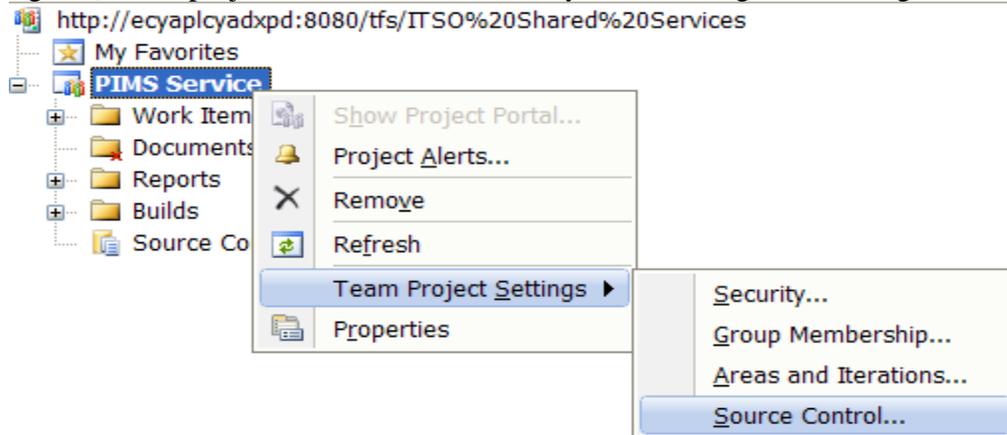
General TFS Guidelines – V2.1.2

Part 1 – TFS 2010 Source Control Setting Guidelines	P2-P6
Part 2 – Team project structure	P7-P8
Part 3 – Branching and Merging	P9
Part 4 – Storing project document.	P9

Part 1 - Team Foundation Server-2010 Source Control Setting Guidelines

General Information: Using the TFS as a source-code control without knowing the true meaning of the TFS project portal settings can lead to confusion, version conflicting, especially when there are multiple developers working on the same project. There are four TFS source-code control options. This document explains the two most common options that can be used depending on the nature of the development team and their projects.

The TFS default setting is “Enable multiple check out”. This setting can be changed by anyone who has access to the project. To change the setting, select Team Explorer, right click on the Project name, Team Project Setting then Source Control (see the below figures). The project lead should be the only one to change this setting.



1. The TFS default setting: Enable multiple check out

Each time the developer adds a new item, The TFS also checks out the project solution file (.csproj). This top level xml file is the one that keeps track and manages all items in the project. Other developers can also add new items and the projects solution is also checked out to them. Each developer will not see the new items or folders that were added by the others. The developers only see these items when they checked them in and performed the Get Latest Version.

This setting allows multiple developers to check-in the same item back into TFS. Whoever checks-in last, the TFS will prompt a message to reconcile the conflicting changes that were made by the previous developer.

Example: Developer A and B perform the Get Latest Version for project TestMeTFS. Neither developer has checked out any items from their solution yet.

- Developer A adds a new item Class1.CS : Visual Studio checks out the project file TestMeTFS.csproj. (see figure 1)
- Developer B adds a new item EmployeeForm.aspx: Visual Studio also checks out the project file TestMeTFS.csproj for this developer.
- Developer A checks in the new item Class1.CS and TestMeTFS.csproj project file back into TFS: Everything checks-in fine.
- Developer B checks in the new item EmployeeForm.aspx and TestMeTFS.csproj project file back into TFS: Visual Studio prompts the Developer B to reconcile the TestMeTFS.csproj project file. (see the figure 2)

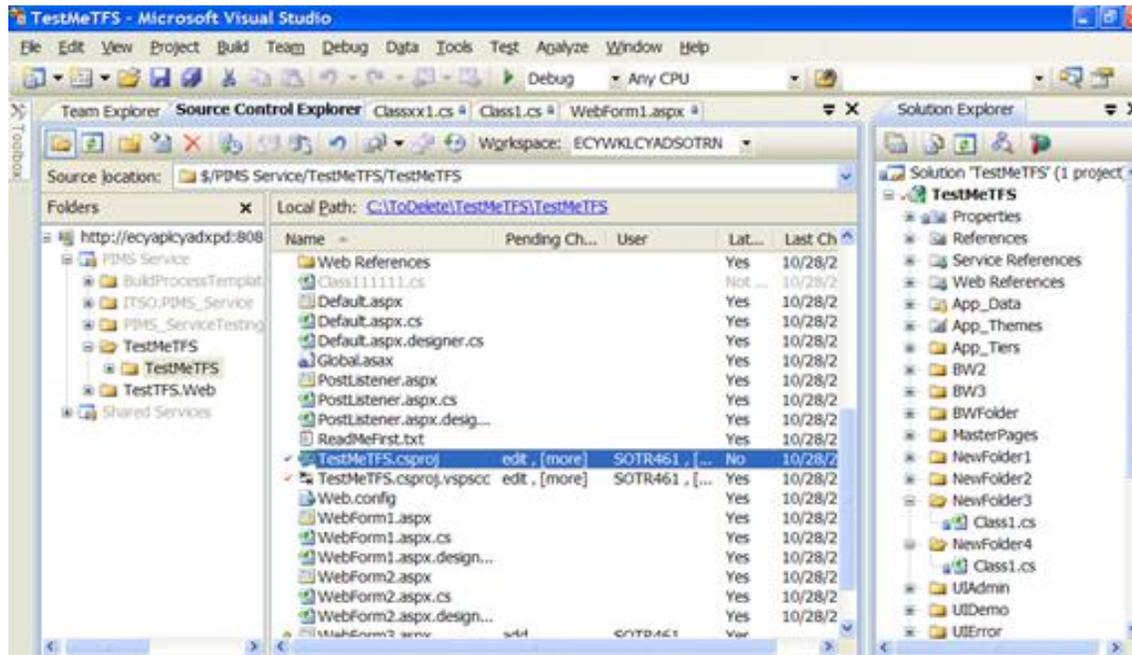


Figure 1: added a new item, VS checked out the project file.

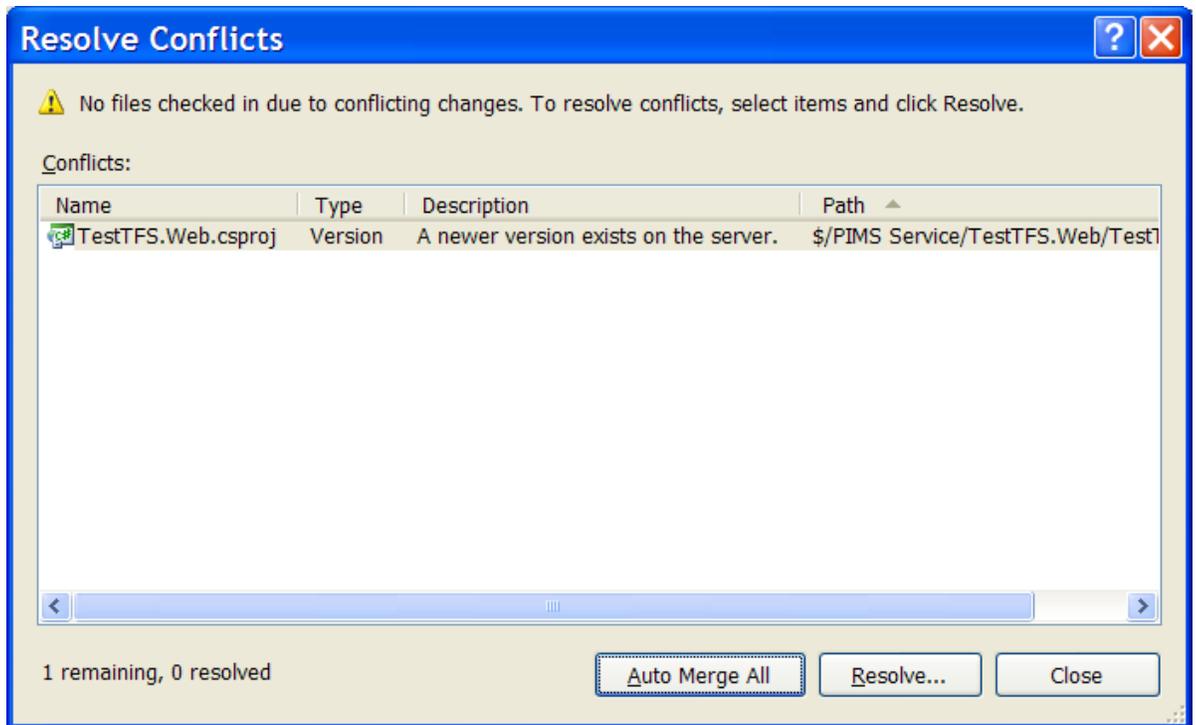
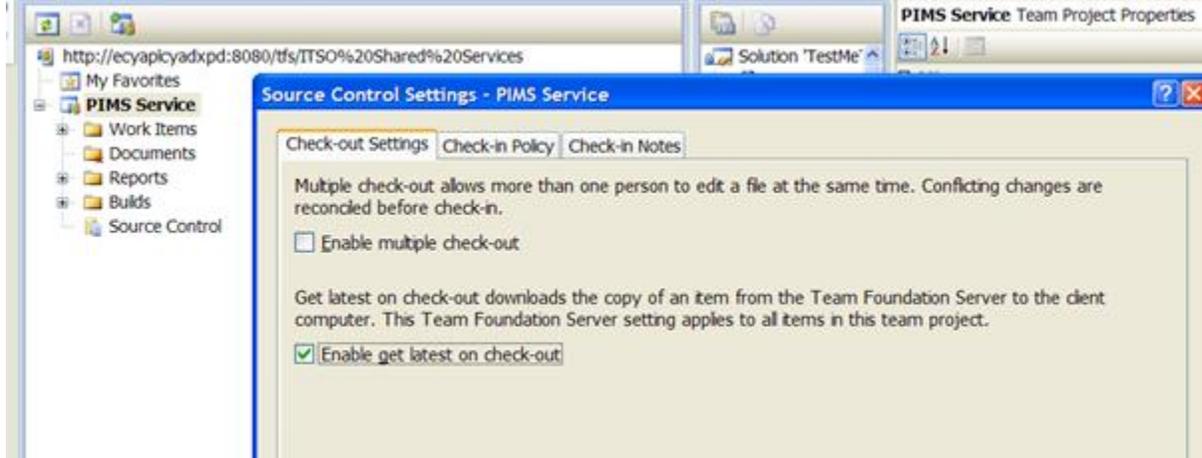


Figure 2 - Project Solution file conflict needs to resolve

2. Disable “multiple check-outs.” Enable “get latest on check out.”

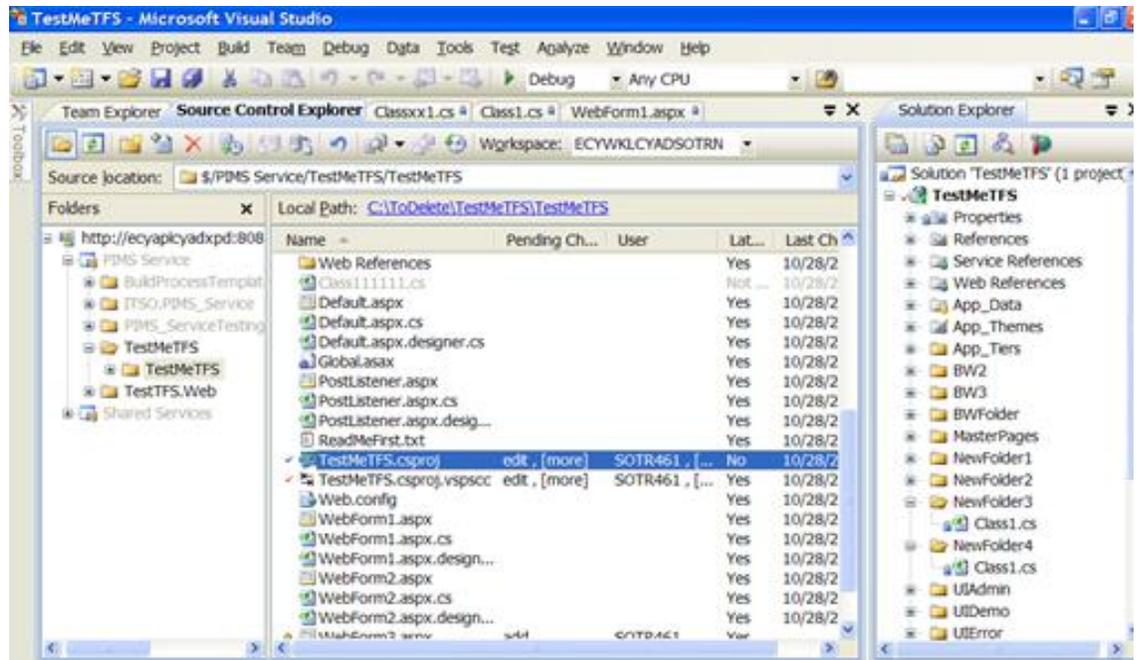


This option prevents developers from checking out the same item. The developers will not have to go through the reconciling process to solve conflicting changes. However, each time there is a new item added to the project, the TFS also checks the solution file out. The Visual Studio uses this file to manage all the items within the project. By checking out this solution file, the TFS prevents other developers from adding new items. To avoid this obstacle, all developers should check-in the solution file right after adding the new items.

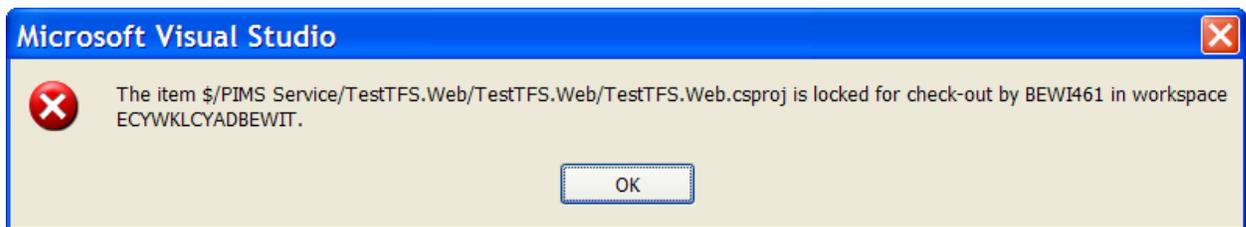
Note: When there is a new item added by one developer even if it is already checked-in, other developers will not see it until they close the project and open it again or perform the Get Latest Version.

Example: Developer A and B perform the Get Latest Version for project TestMeTFS. Neither developer has checked out any items from their solution yet.

- Developer A adds a new item Class1.CS : Visual Studio checks out the project file TestMeTFS.csproj and adds a new item Class1.CS (see figure 3)
- Developer B performs “add a new item” WebForm5.aspx: Visual Studio displays the locked message (see figure 4).
- Developer A checks in the TestMeTFS.csproj project file.
- Developer B performs “add a new item” again for WebForm5.aspx: Visual Studio checks out the project file TestMeTFS.csproj and allows WebForm5.aspx file to be added to the project.



• Figure 3: added a new item, VS checked out the project file.



• Figure 4: Locked message preventing new item from added.

Guidelines:

Use **Enable multiple check out** if the project team is comfortable with resolving conflicts for the project solution file and team members are unlikely to be working on the same item. Team leaders should establish:

- A routine TFS check-in schedule
- Rules for which items/folders are required to be checked-in regularly
- Schedule for developers to perform the "Get latest version" in order to have a full picture of the project.

Use **Disable multiple check out / enable Get Latest Version** if team members can check-in the project file each time they add new items, otherwise other team members will not be able to create new items. To avoid the burden of checking-in the project file all the time, the team lead can add a set of folders and empty items as a holding place to work on later then check-in the project file.

Part 2 - TFS - Team project Structure Guideline –Adoption Date: 09/14/2008

This guideline is based on Microsoft book: Patterns & Practices - Team Development with Visual Studio Team Foundation Server. This book can be found at:

<http://teams/sites/AS/it/SAT/Helpful%20Reading/TFSGuide.pdf> *

The developers should read this book prior using the TFS. This book will serve as guidelines for items not addressing in this document. On page 23 to 35, it is addressing the project structure and we encourage all TFS users read at least these pages.

Server-Side Structure for TFS (TFS Server): After /Main/Source folder, the sub folders are optional depending on the size and nature of the project.

\$ PName.TeamProjectName (e.g. \$ADS.SoftCenter or \$TCP.MYEIM)

- /Main → can contain solution .sln files
- /Source → can contain 1 to many apps related to the Team-Project)
 - /AppName1 → contains AppName1 .sln file
 - /Source → contains folder for all source
 - /AppName.Web* → contains Default.aspx
 - / AppName.ClassLib1* → ClassLib1 project
(ClassLib1 = DataAccess, Business..etc.)
 - ...
 - /UnitTests
 - / AppName.WebTest → Test project and codes
 - / AppName.ClassLib1Test → Test project and codes
- /AppName2
- ...
- /DatabaseCode
 - /BuildScripts
 - /ChangeScripts
- ...
- /SharedBinaries → Shared binaries e.g. libraries
- /SharedSource → Shared Source Code
- /VendorTools → 3rd parties components or codes
- ...
- /Build → Build output for deployment
- /Documents → Production documentation
- /BuildScripts → build scripts
- /Tests
 - /FunctionalTests
 - /PerformanceTests

* Please review the [C# naming guideline](#)

Client-Side Structure (Developers' PC)

Each development workstations should have one root directory to contain all Team projects as in this example:

```
C:\DevProjects           → Root folder for all team projects
  \ PName.TeamProjectName1 → Container folder for TeamProject 1
  \ PName .TeamProjectName2 → Container folder for TeamProject 2
```

Beneath each team project folder, use a copy of the application folder structure used on the source control server.

```
\PName.TeamProjectName1
  /Main           → can contain solution .sln files
  /Source         → can contain 1 to many apps related to the Team-Project)
    /AppName1    → contains AppName1 .sln file
      /Source    → contains folder for all source
        /AppName.Web* → contains Default.aspx
        /AppName.ClassLib1* → ClassLib1 project
          (ClassLib1 = DataAccess, Business..etc.)
          ...
      /UnitTests
        /AppName.WebTest → Test project and codes
        /AppName.ClassLib1Test → Test project and codes

    /AppName2
    ...
    /DatabaseCode
      /BuildScripts
      /ChangeScripts
    ...
    /SharedBinaries → Shared binaries e.g. libraries
    /SharedSource → Shared Source Code
    /VendorTools → 3rd parties components or codes
    ...
  /Build → Build output for deployment
  /Docs → Production documentation
  /BuildScripts → build scripts
  /Tests
    /FunctionalTests
    /PerformanceTests
```

* Please review the [C# naming guideline](#)

Part 3 – Branching and Merging (per MS guidance)

Do not branch unless it becomes necessary for your development team. Branching introduces additional source tree maintenance and merging tasks. Most development teams such as those building line of business applications, working on short release cycles do not need to branch. Development teams working on longer release cycles are more likely to need branching as part of the development process.

If you have one stream of development, or are performing incremental and continuous releases, you might not need to create branches unless you frequently experience breaking changes that are destabilizing your development efforts.

Before branching, developers need to read page 40 for setting up Branching and page 45 to 53 on chapter Defining Your Branching and Merging Strategy from this [TFS Microsoft Best Practices](#).^{*} For folder and element naming standard, the sample in the previous chapter is applied for project, application and package name.

Part 4 - Storing Documents (per MS guidance)

The Docs folder is for product related documentation. To help determine what documents to store in TFS source control and what to store in a document library on your SharePoint team site, consider the following:

- Use SharePoint for internal team documents such as use cases, scenario and requirements documentation, and design documentation.
- Use TFS source control for product-related documentation that you ship to your customers. This could include installation and deployment guides, operation guides, and help files.
- Link is available for Ecology intranet only

Document History

Date	Version	Editor	Change
June 29, 2007	Pilot	Son Tran	Part 2 and Part 4 -Reviewed by WR (Jeremiah), TCP(Wayne), ADS(Balaji, Son, Randy) - Adopted to use as pilot
8/25/2008 9/02/2008 9/14/2008	1.0	Son Tran	Part 2 and Part 4 - Review by Strategic Architecture team -Review by Architecture Work Group -Approved to release by Enterprise Manager (Debbie Stewart)
12/23/2010	2.0	Son Tran	Insert Part 1
10/04/2011	2.1	Son Tran	Insert Part 3 and reformat the document