User Guide – Eclipse & Linux - TFS Integration

Version No.: 1.0

Date: 16-May-2014

Project Name: Boeing- Software Improvement Team

Project Code: C114572

**Copyright ©: The Boeing Company**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version No. | Date | Prepared by / Modified by | Significant Changes |
| 1.0 | 16-May-2014 | Narayanamurthy Balasubramaniyan | Initial Draft |
| 2.0 | 21-Aug-2014 | Narayanamurthy Balasubramaniyan | TFS integration with Linux |

Glossary

|  |  |
| --- | --- |
| Abbreviation | Description |
| TFS | Team Foundation Server |
| SSL | Secure Socket Layer |
| TEE | Team Explorer Everywhere |

Table of Contents

[1 Introduction 5](#_Toc398118604)

[1.1 Purpose 5](#_Toc398118605)

[2 TFS in eclipse 5](#_Toc398118606)

[2.1 Install TFS plugin in eclipse 5](#_Toc398118607)

[2.2 Enable SSL Certification for TFS 10](#_Toc398118608)

[2.3 Configure Boeing Certificates 13](#_Toc398118609)

[2.3.1 Steps to Configure Boeing Certificate: 13](#_Toc398118610)

[2.4 Connect to TFS Server 17](#_Toc398118611)

[2.5 Import project 21](#_Toc398118612)

[2.6 Create TFS Workspace 27](#_Toc398118613)

[2.7 Map working folder 29](#_Toc398118614)

[2.8 Remove mapping 31](#_Toc398118615)

[2.9 Get Latest 33](#_Toc398118616)

[2.10 Add Items to Folder 34](#_Toc398118617)

[2.11 Checkout 35](#_Toc398118618)

[2.12 Check-in 37](#_Toc398118619)

[2.12.1 Using Project Explorer 37](#_Toc398118620)

[2.12.2 Using Team Explorer 39](#_Toc398118621)

[2.13 Shelve 42](#_Toc398118622)

[2.14 Undo Pending changes 45](#_Toc398118623)

[2.15 Label 47](#_Toc398118624)

[2.16 Find Label 49](#_Toc398118625)

[2.17 History 52](#_Toc398118626)

[2.18 Diff 53](#_Toc398118627)

[2.19 Branch 56](#_Toc398118628)

[2.20 Merge 63](#_Toc398118629)

[2.21 Build definition 69](#_Toc398118630)

[2.22 Gated Check-in 83](#_Toc398118631)

[2.23 User story traceability 89](#_Toc398118632)

[2.24 Work Items 95](#_Toc398118633)

[2.24.1 Create a new Work Item 97](#_Toc398118634)

[2.25 Queries 99](#_Toc398118635)

[3 TFS in Linux 103](#_Toc398118636)

[3.1 Glossary 103](#_Toc398118637)

[3.2 Install TFS in Linux 103](#_Toc398118638)

[3.3 Accept License Agreement 105](#_Toc398118639)

[3.4 Create TFS Workspace 107](#_Toc398118640)

[3.5 View list of available workspaces 107](#_Toc398118641)

[3.6 Map TFS Working folder 107](#_Toc398118642)

[3.7 Get 108](#_Toc398118643)

[3.8 Add Items to folder 109](#_Toc398118644)

[3.9 Status 109](#_Toc398118645)

[3.10 Checkin 110](#_Toc398118646)

[3.11 Rename 110](#_Toc398118647)

[3.12 Delete 111](#_Toc398118648)

[3.13 Branch 111](#_Toc398118649)

[3.14 View Branches 112](#_Toc398118650)

[3.15 Label 112](#_Toc398118651)

[3.16 View Labels 113](#_Toc398118652)

[3.17 History 113](#_Toc398118653)

[3.18 Diff 113](#_Toc398118654)

[3.19 Merge 113](#_Toc398118655)

[3.20 Shelve 114](#_Toc398118656)

[3.21 Properties 114](#_Toc398118657)

[3.22 Undo 115](#_Toc398118658)

[3.23 Unmap TFS Working folder 115](#_Toc398118659)

[4 Reference 115](#_Toc398118660)

# Introduction

## Purpose

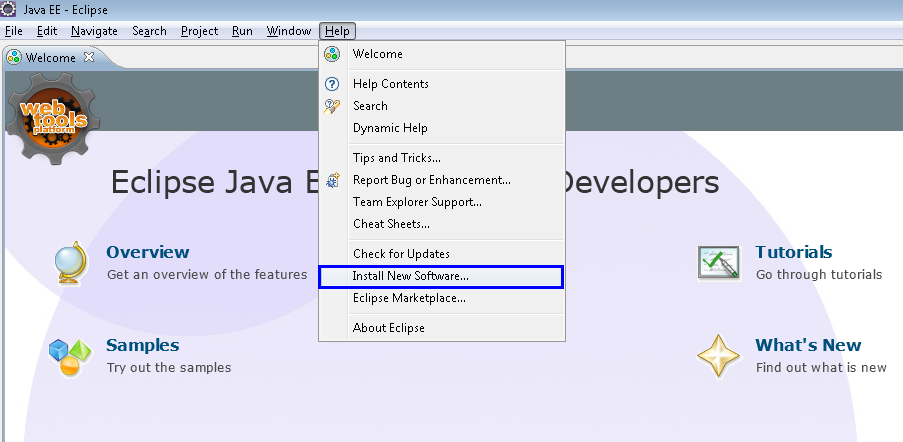
This document guides the user to configure and work with TFS in eclipse and Linux.

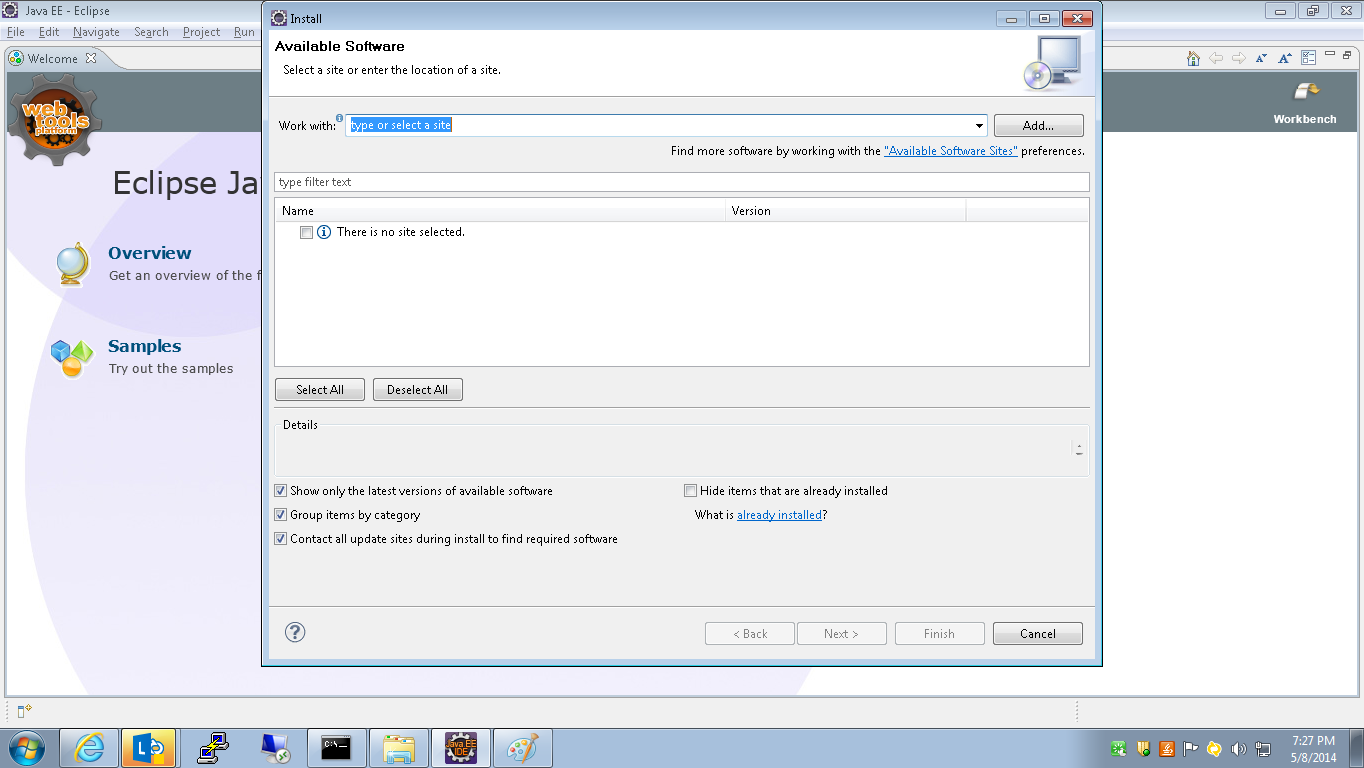
# TFS in eclipse

## Install TFS plugin in eclipse

1. Open Eclipse.



1. Go to Help 🡪Install New Software.
2. A new window opens to install the softwares.



1. Click Add.

Provide Name and Location URL of the software to be installed and Click OK.

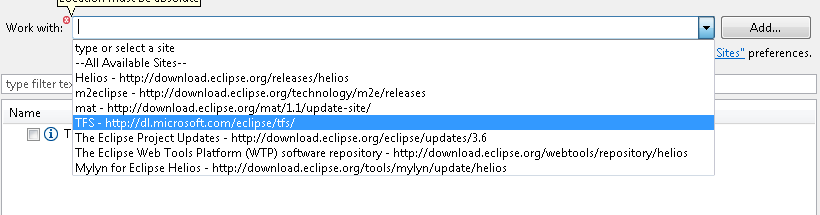
Herein case

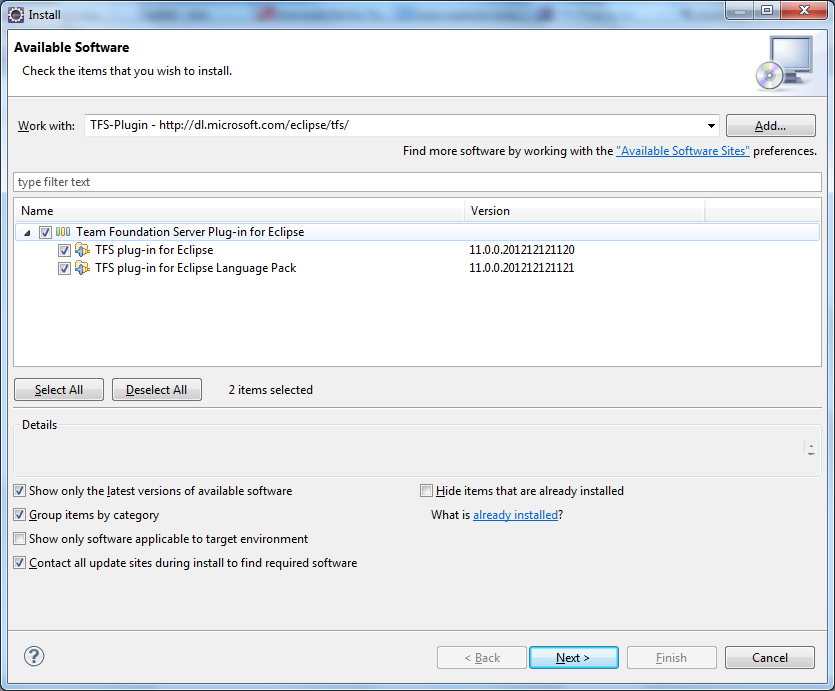
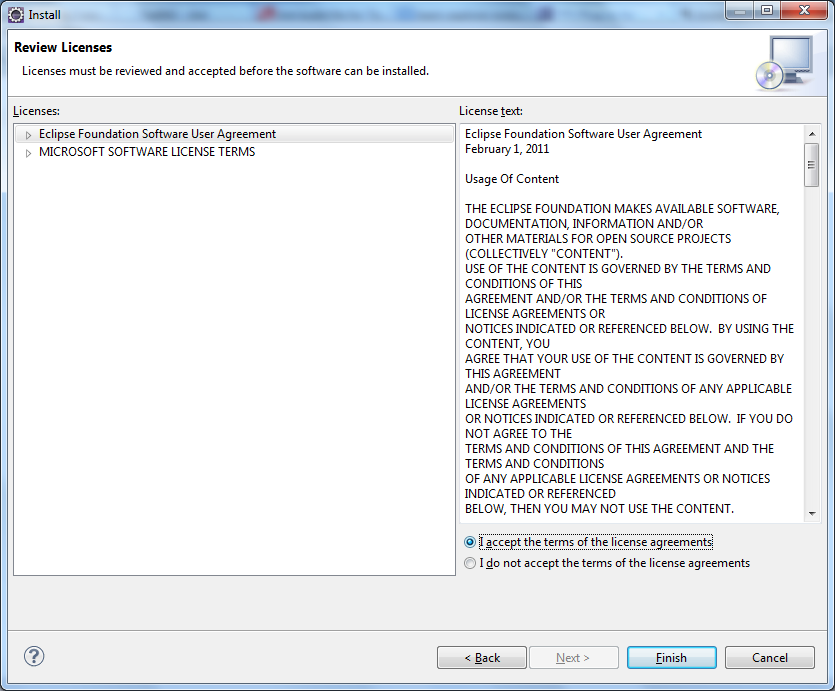
**Name** : TFS-Plugin

**Location**: <http://dl.microsoft.com/eclipse/tfs/>

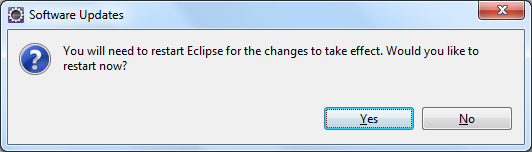


1. Select the added Repository for software to be installed from the dropdown as shown below.



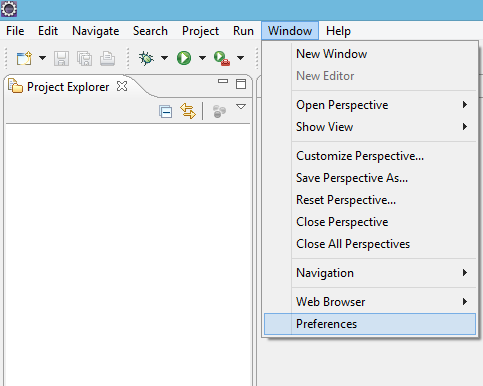
1. Check the required software package as shown below. Click Next.
2. Accept the license agreement and click “finish” button.
3. Once the license is accepted, it starts installing the “TFS-Eclipse Plugin”.

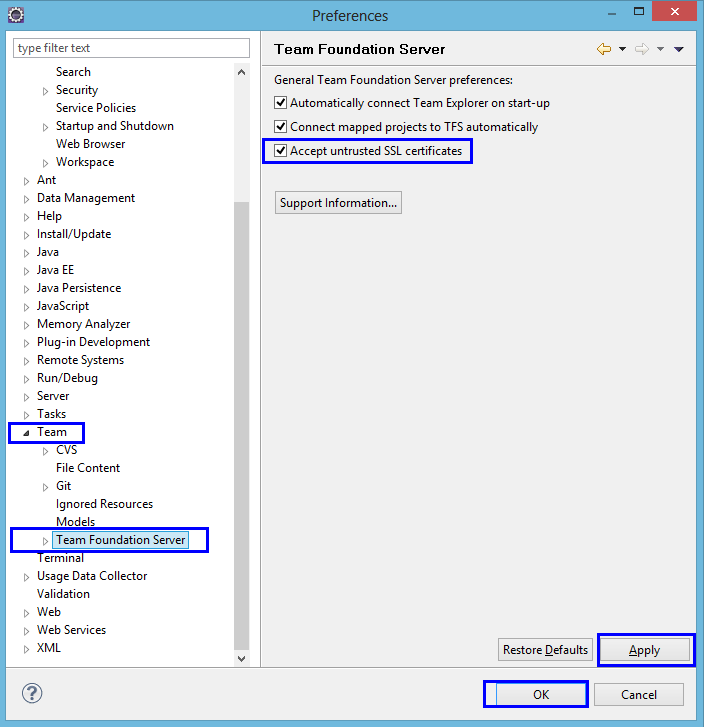
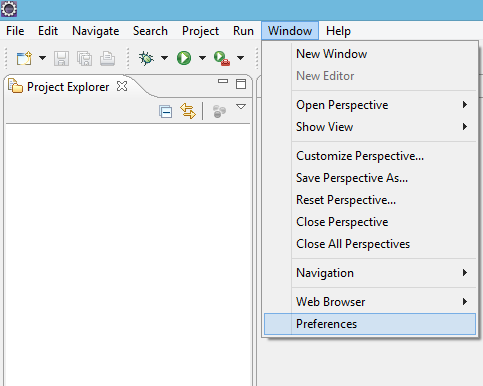


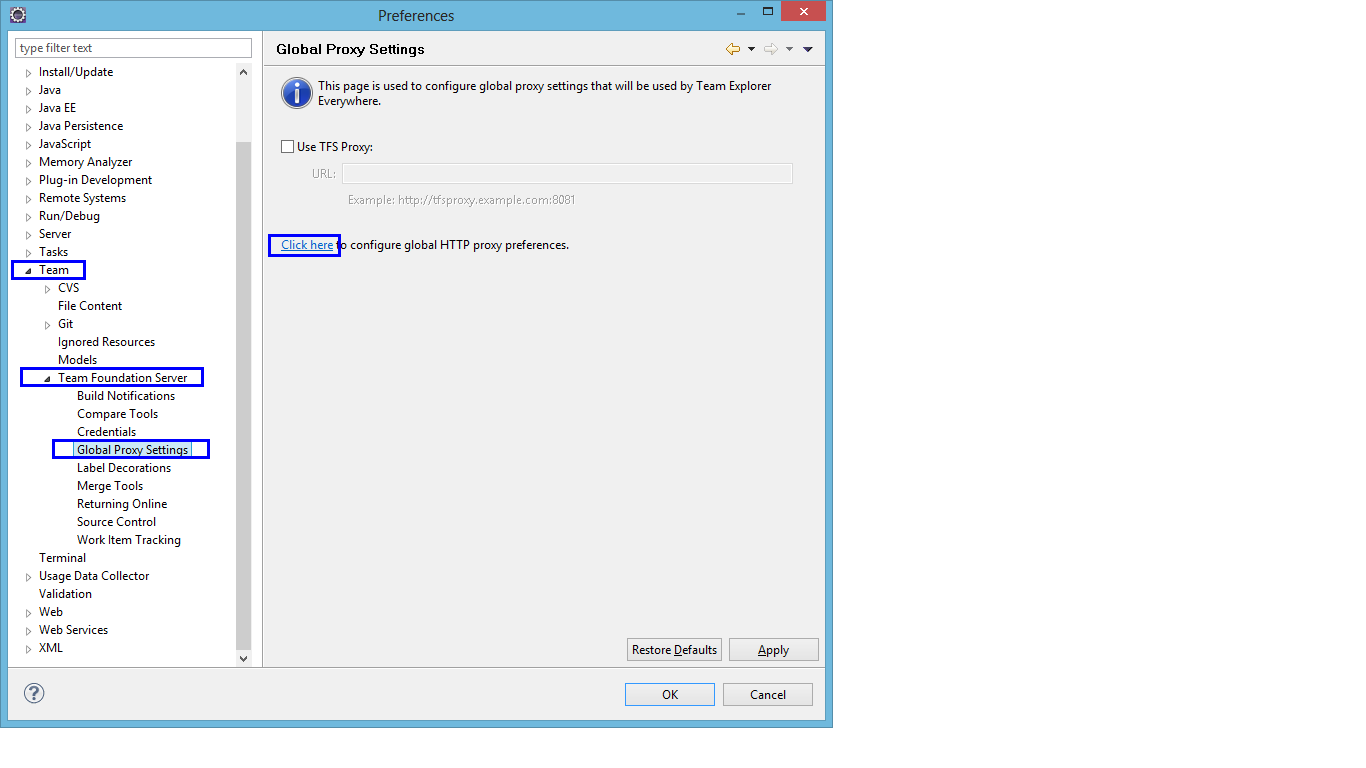
1. Once Installation is completed, Eclipse asks for “Restart”.
2. Click “Yes” to restart eclipse.

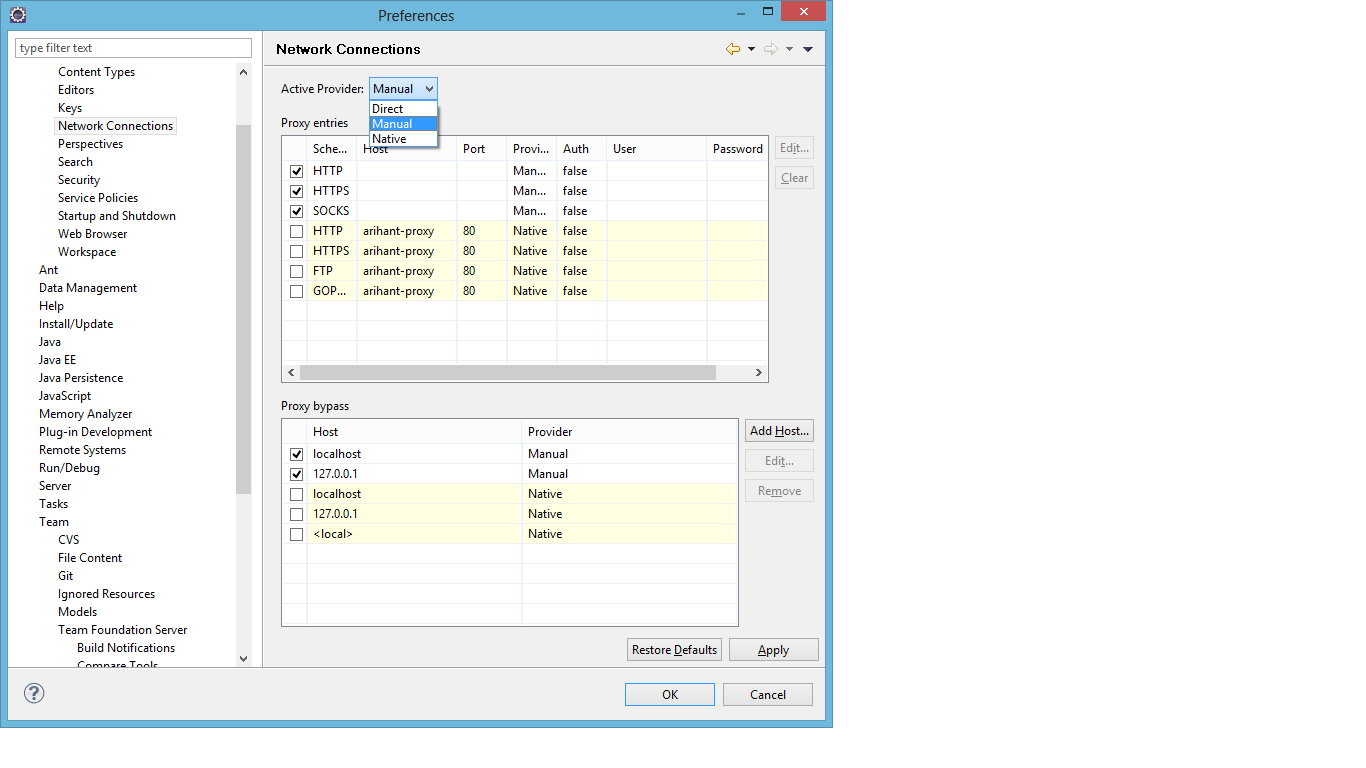
## Enable SSL Certification for TFS

1. Go to Window 🡪 Preferences.



1. Go to Team🡪Team Foundation Server. Enable the option “Accept untrusted SSL certificates”. Click “Apply”. Click OK.
2. Go to Window 🡪 Preferences.
3. Click Team 🡪 Team Foundation Server 🡪 Global Proxy Settings 🡪 Click here.



1. Select Manual from the Active Provider drop down menu and click OK.

## Configure Boeing Certificates

### Steps to Configure Boeing Certificate:

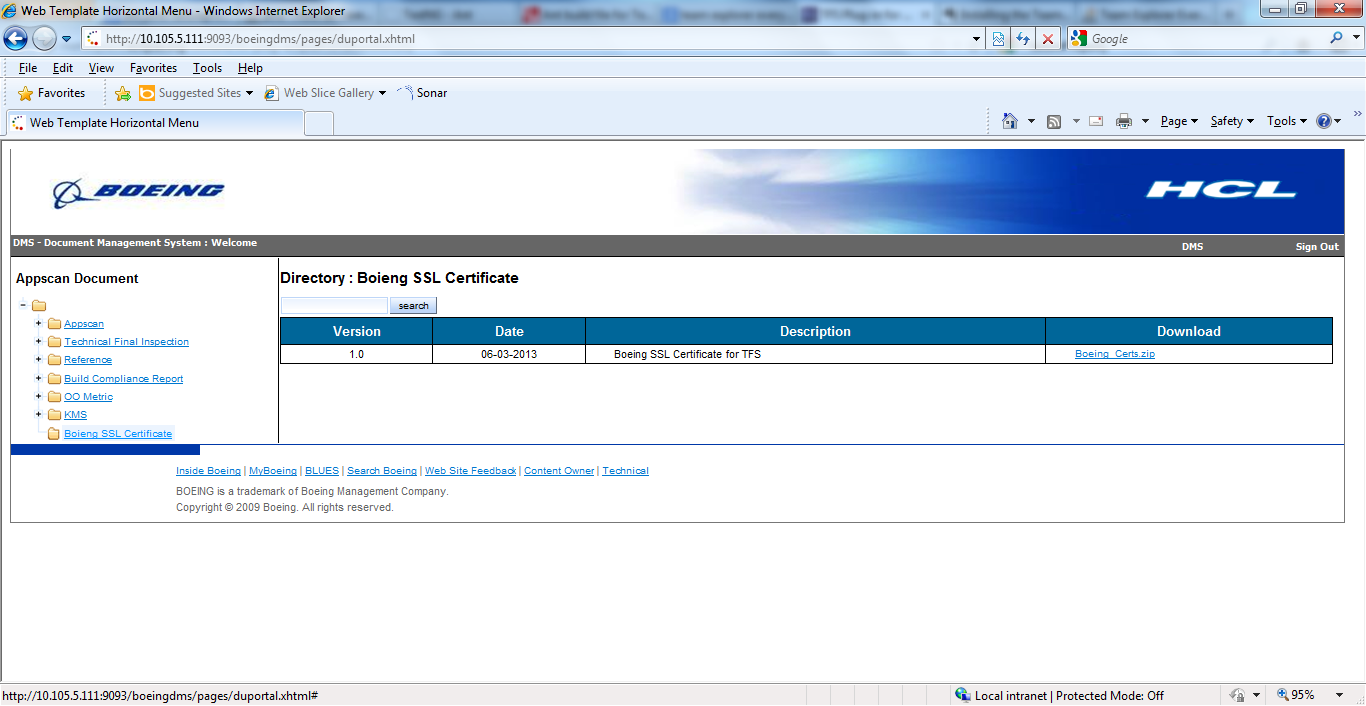
#### Onsite Users:

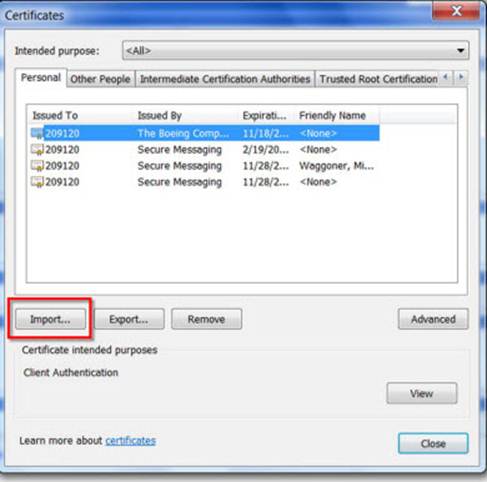
1. Kindly Refer Below Link For Installing Boeing\_Certificate:

<https://fec.web.boeing.com/SitePages/FAQ_Details.aspx?faq_id=621>

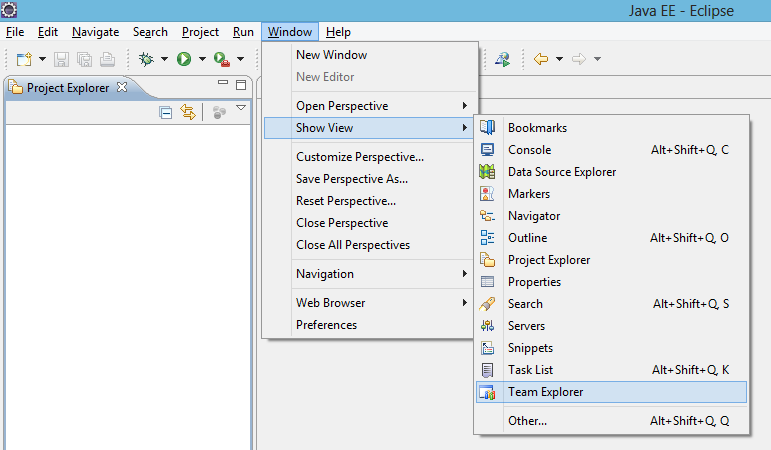
#### Offshore Users:

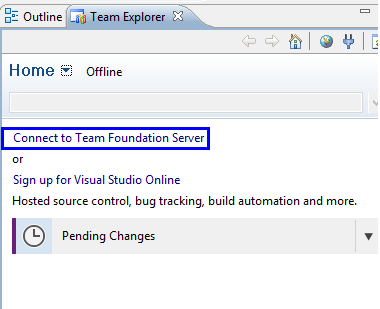
1. Goto: <Http://10.105.111:9093/boeingdms>
2. In the left side of the page select “Boeing SSL Certificate” and you will find the “Boeing\_Certs.zip”.
3. Click on “Boeing\_Certs.zip” to download it in local machine.
4. Unzip “Boeing\_Certs.zip”.

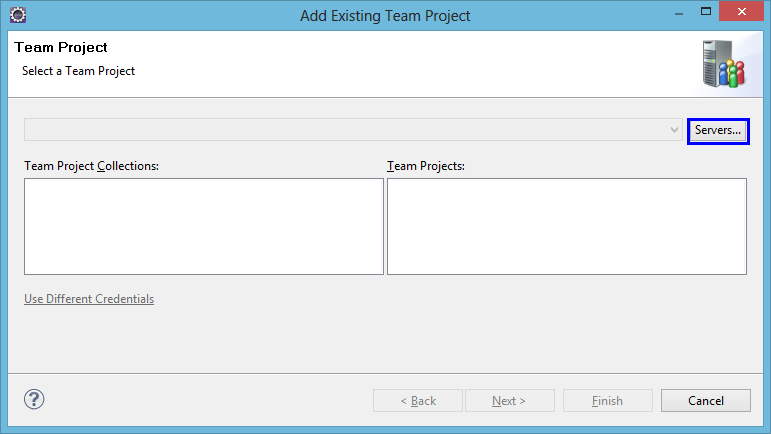


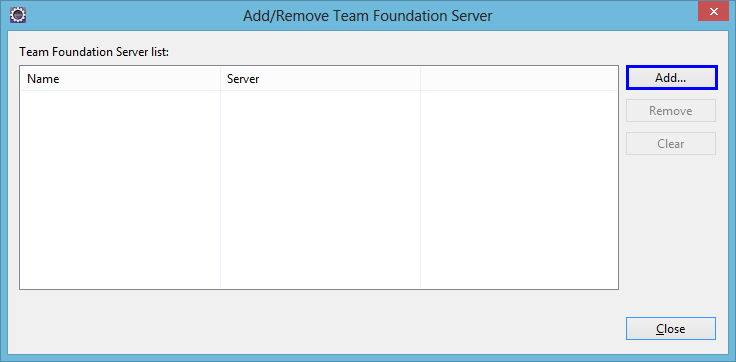
1. Connect to the Boeing VPN:  
   Insert access card into the PC and point web browser to <https://connect.boeing.com/tals>. Select either the Split Tunnel or Tunnel All option.   
   Note: If using Tunnel All access the proxy settings will need to be modified to access the Internet via Boeing VPN.
2. In Internet Explorer navigate to Tools, select Internet Options. Select the Content tab and click on the Certificates button.  
     
   
3. Select the Intermediate Certification Authorities tab and select the Import button.  
   
4. Select the Next button.
5. Browse to the correct location for Boeing provided intermediate certificates **(Downloaded from DMS)** and select next.
6. Follow the remaining prompts within the Certification Import Wizard.
7. Repeat the same process for the Trusted Root Certification Authorities tab to import trusted root certificates provided by Boeing into Internet Explorer

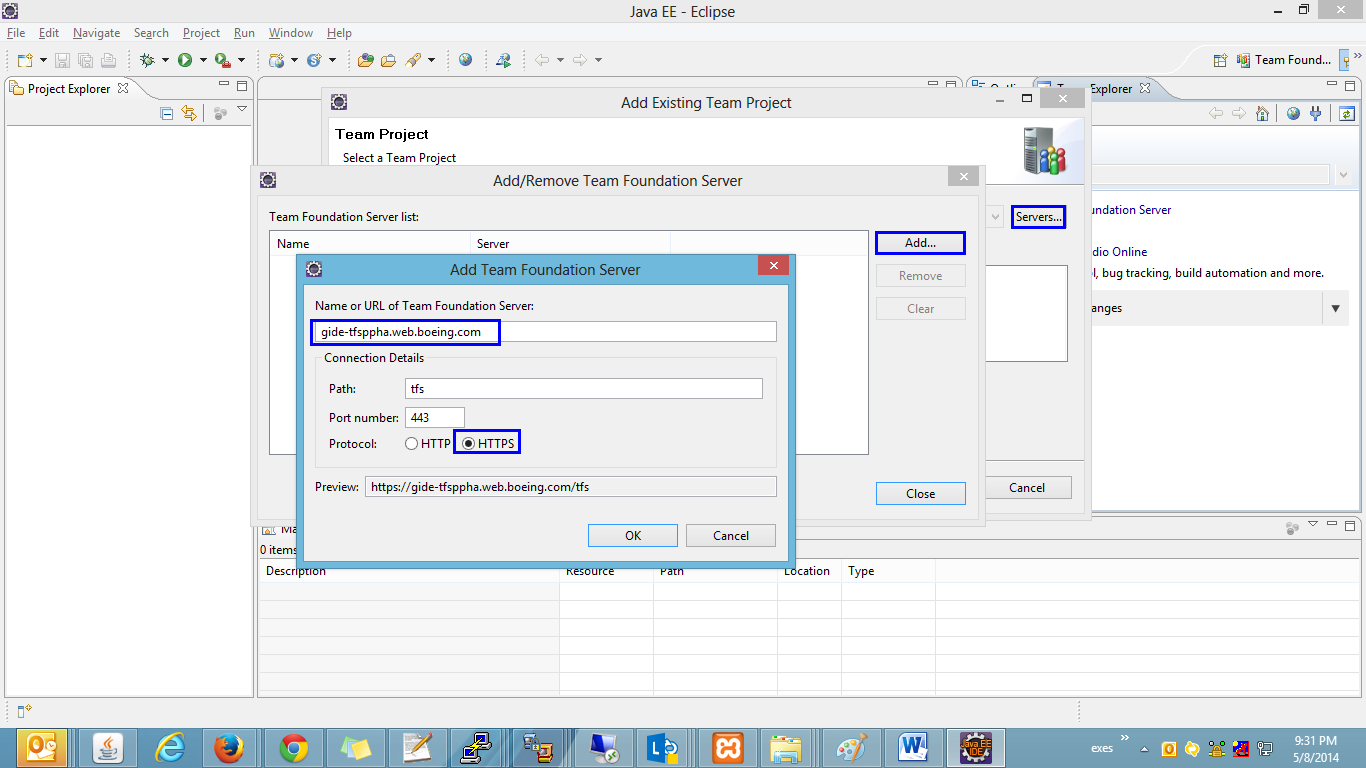
## Connect to TFS Server

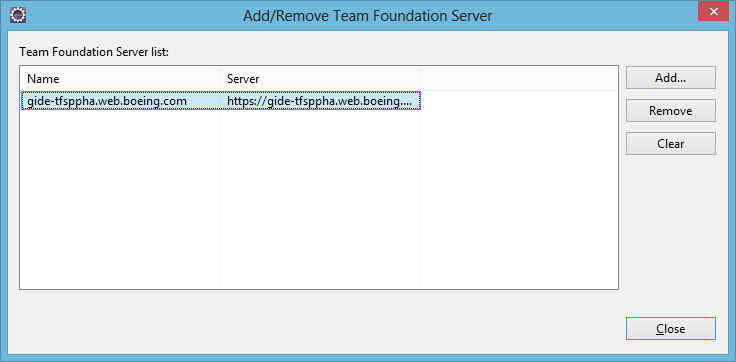
1. Once the TFS plugin is been installed in eclipse, Go to Window 🡪 Show View 🡪 Team Explorer.
2. Click Connect to Team Foundation Server

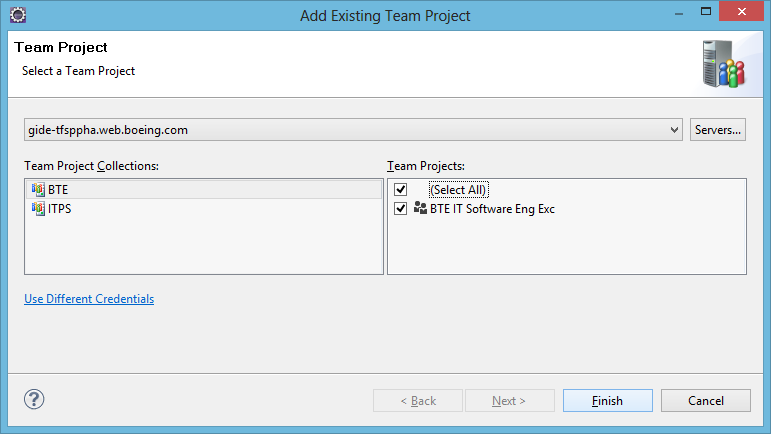


1. Click Servers on the window “Add Existing Team Project” that is popped up.
2. Click Add on the window “Add/Remove Team Foundation Server” that is popped up.

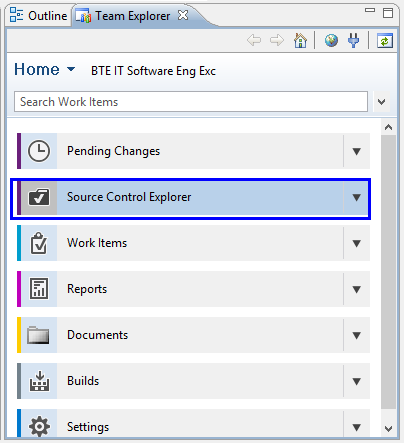
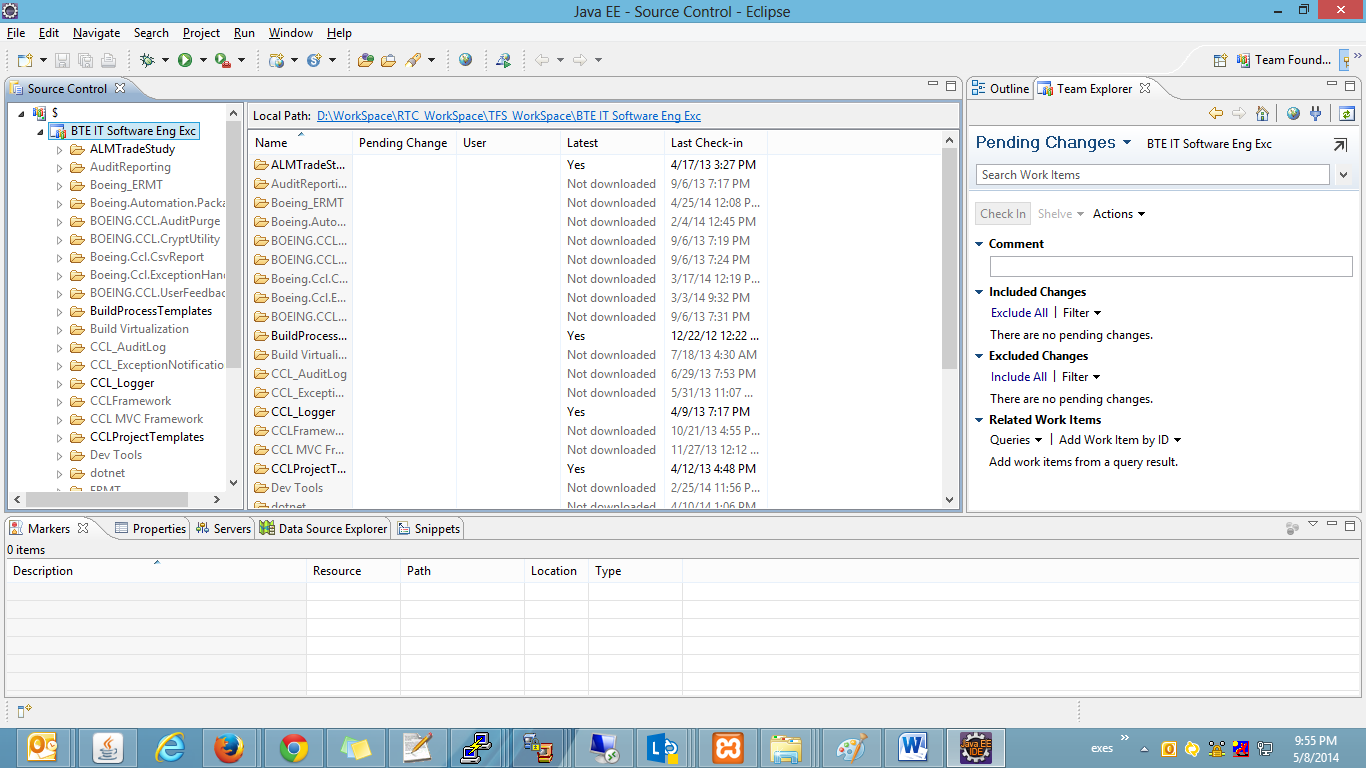


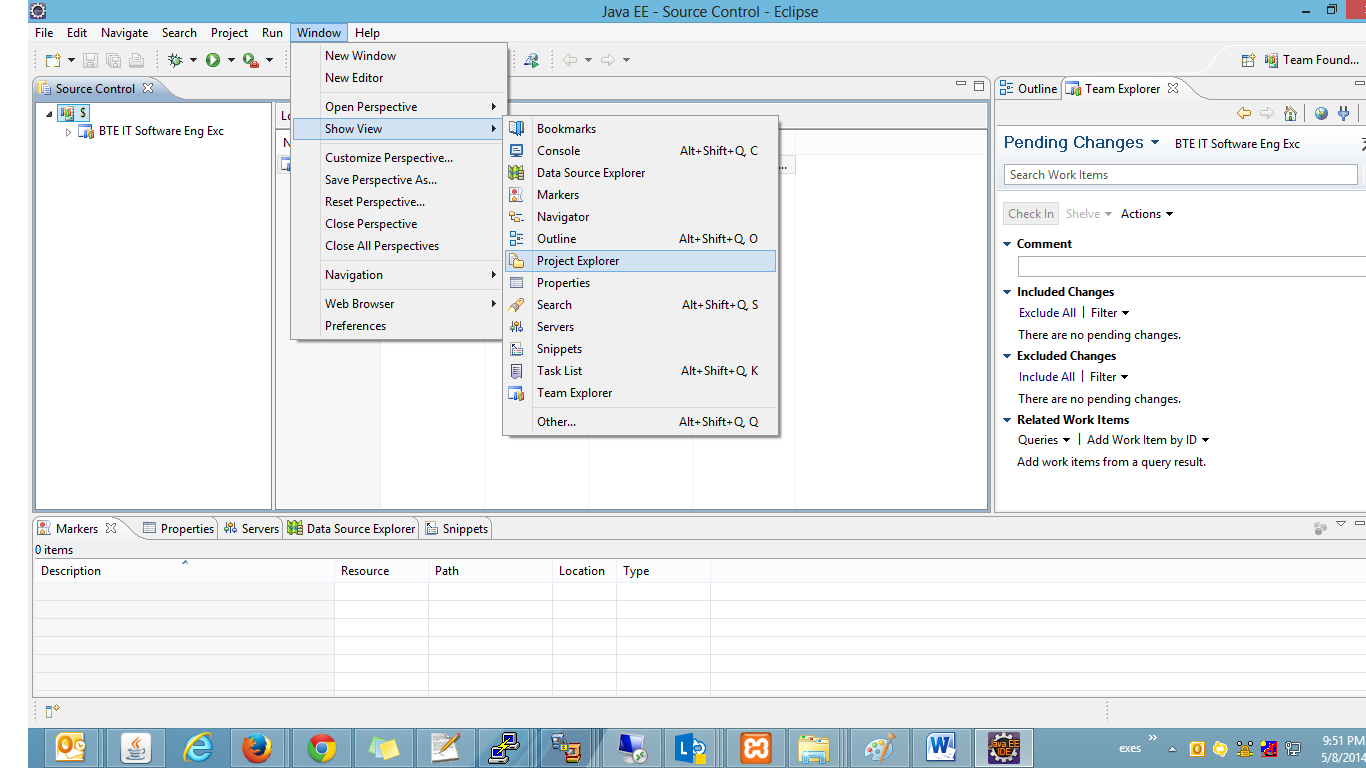
1. Enter the TFS Server details to connect and Click OK.
2. You will be prompted to enter the login credentials. Provide your credentials and click OK.
3. You could see the TFS Server being added as below. Click Close.

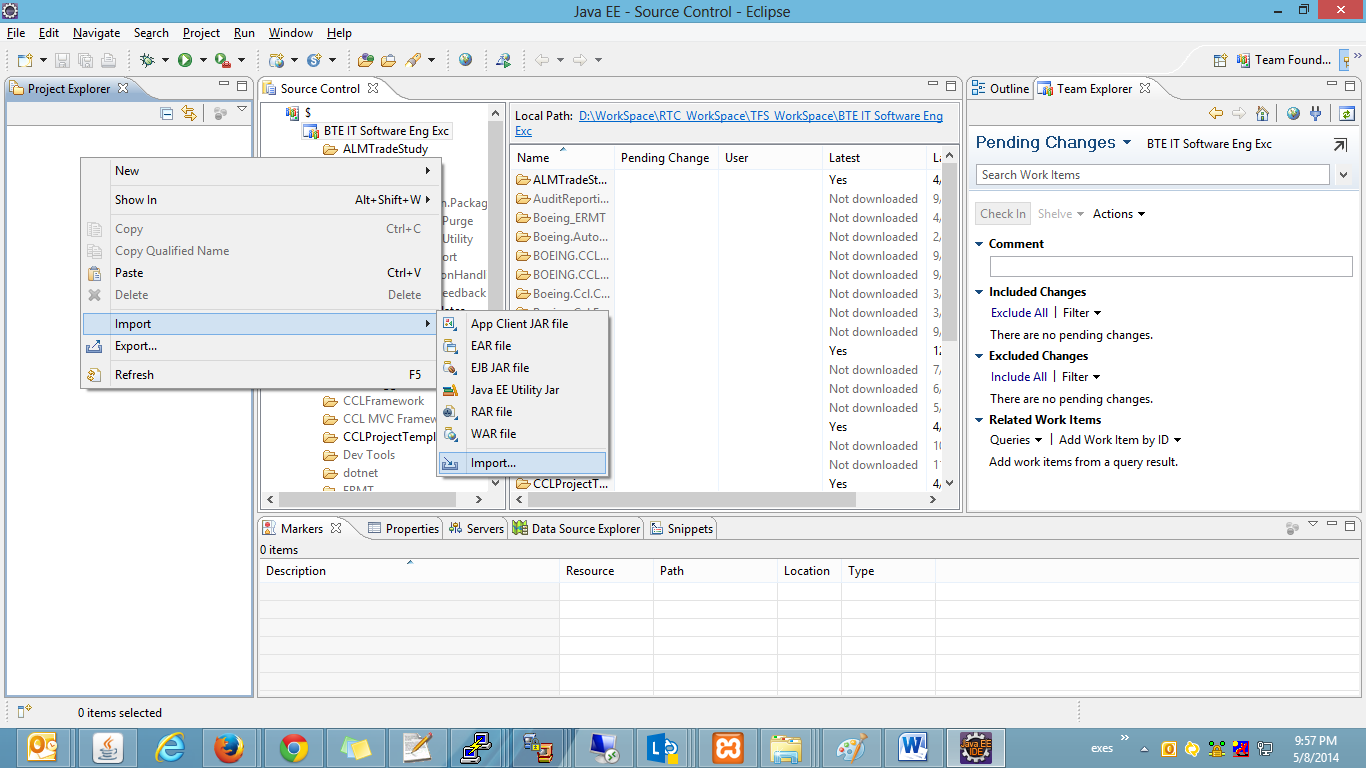
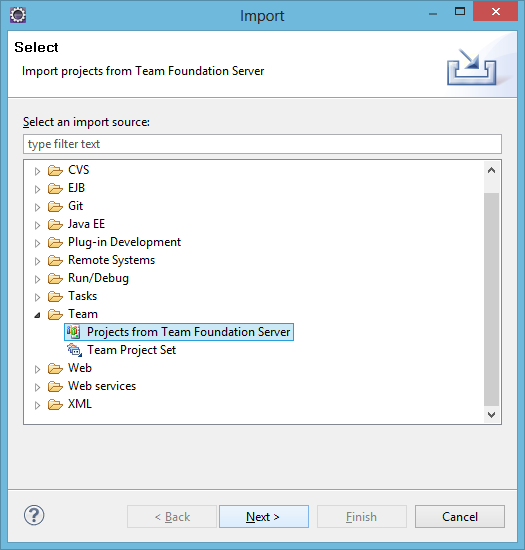
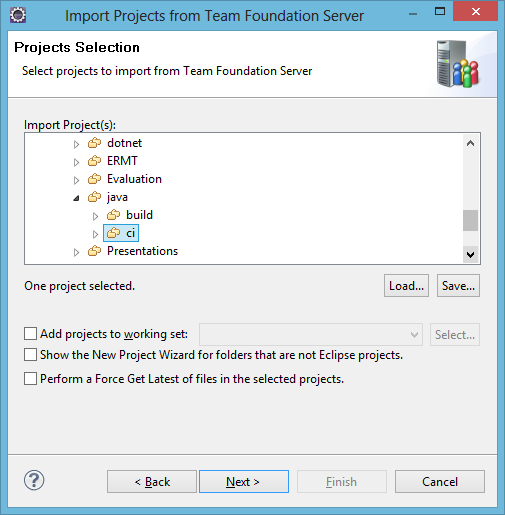


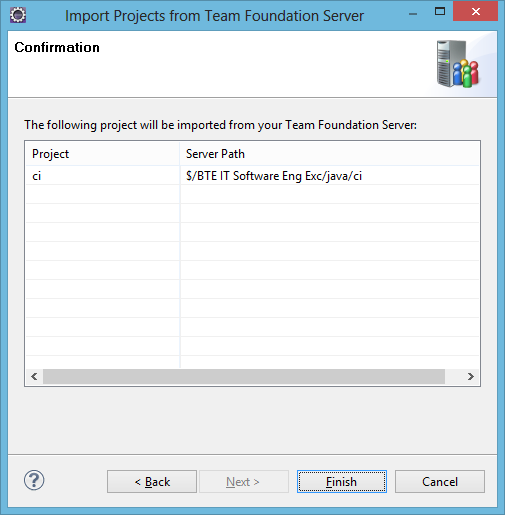
1. Select the Team Project Collections and Team Projects and Click Finish.

## Import project

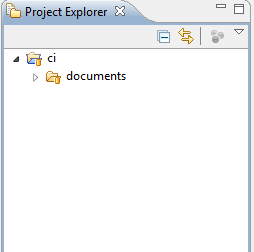
1. Click Source Control Explorer as shown below.
2. A New window appears which shows the contents of connected TFS.
3. Go to Window 🡪 Show View 🡪 Project Explorer.



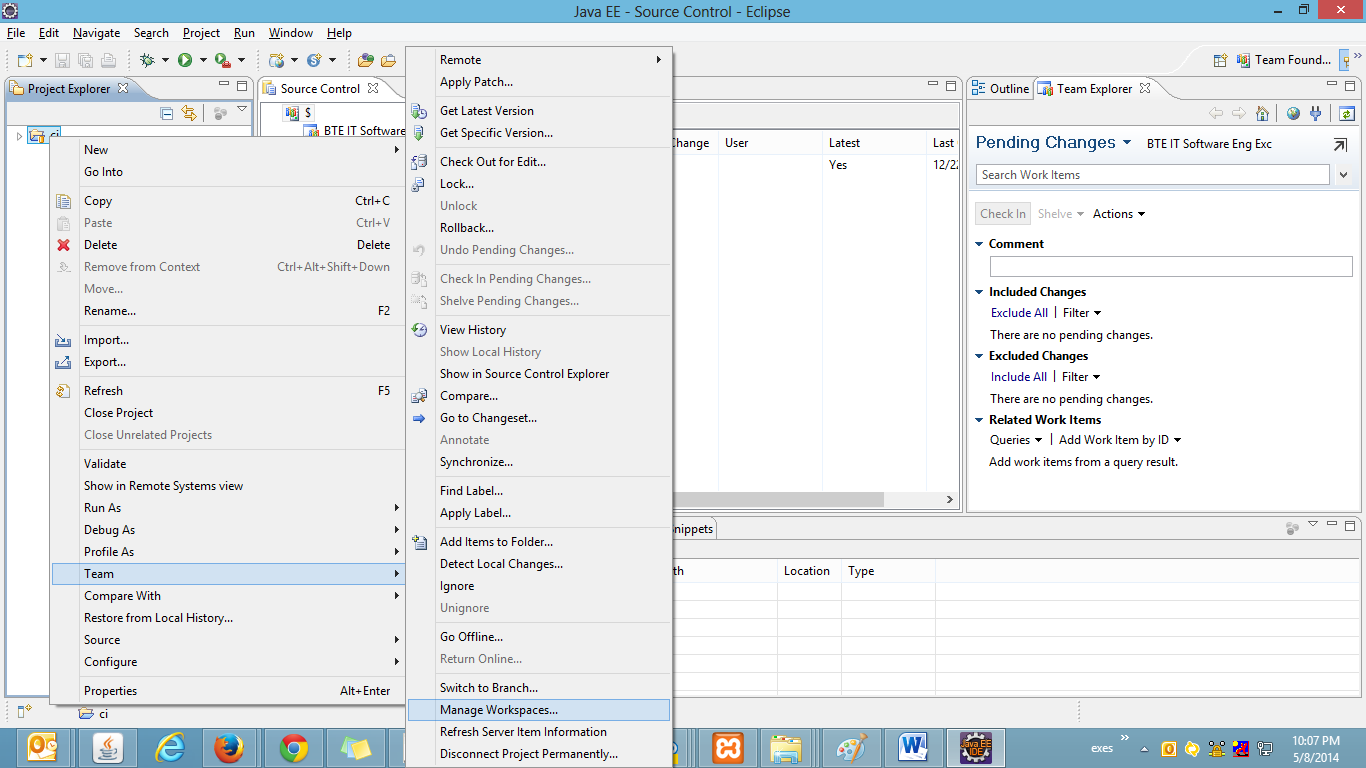
1. Right click on Project Explorer 🡪 Import 🡪 Import as shown below.
2. Select Projects from Team Foundation Server and click Next.
3. Select the projects to import and click Next. Selected **$/BTE IT Software Eng Exc/java/ci** in this case. Click Next.
4. Click Finish.

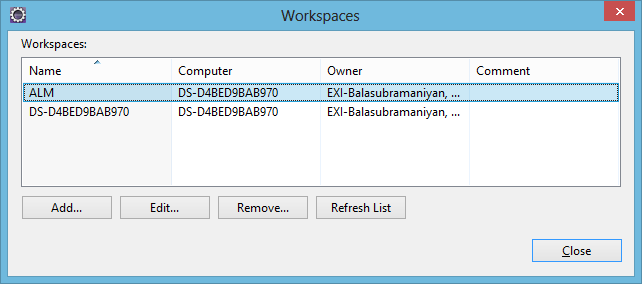


1. You could see the imported project in project explorer.

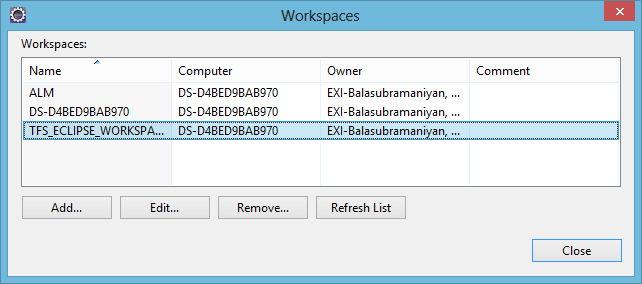


## Create TFS Workspace

1. Right Click on the project in project explorer 🡪 project name 🡪 Team 🡪 Manage Workspaces…
2. List of workspaces associated with the logged in user appears in a separate window.

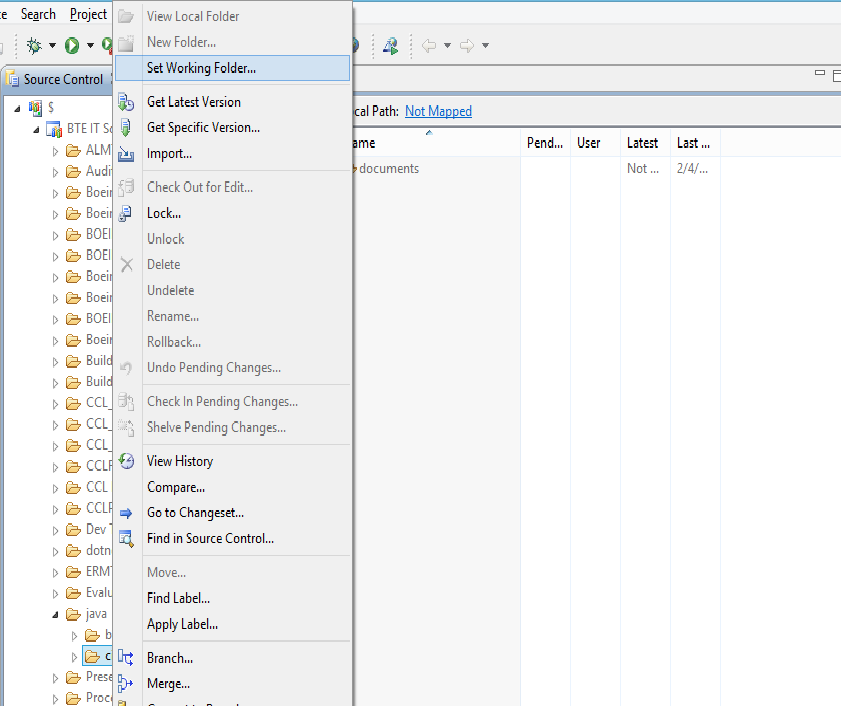
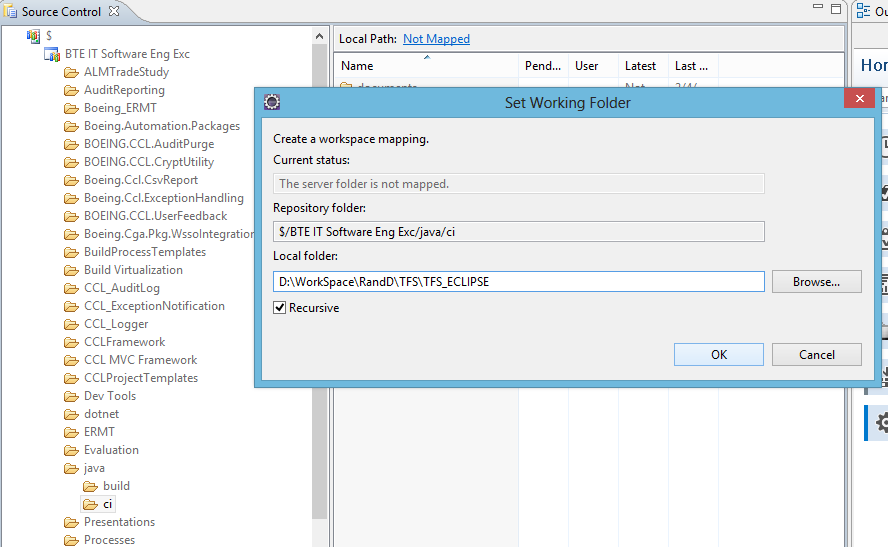


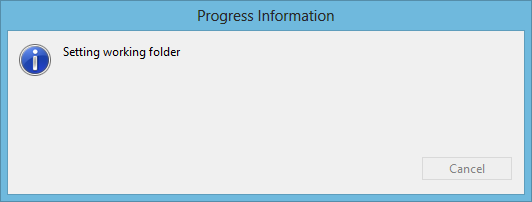
1. Click Add to create a new workspace.
2. Provide the workspace name to be created and Click OK.



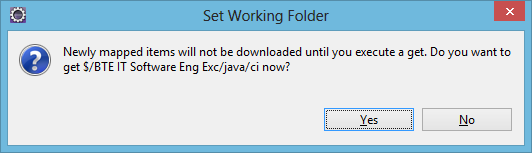
1. Click Close.

## Map working folder

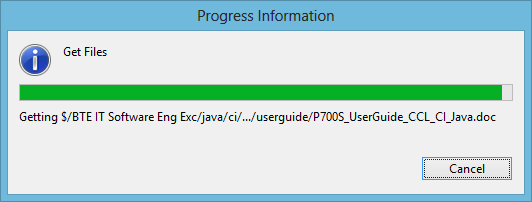
1. Go to Source Control Explorer 🡪 Right click project name 🡪 Set Working Folder.
2. Browse the path of the working folder in “Local Folder” and map it to the repository folder.
3. Progress window.



1. Working Folder has been mapped successfully to the Repository folder and it prompts the user whether to get the mapped items to the working folder.

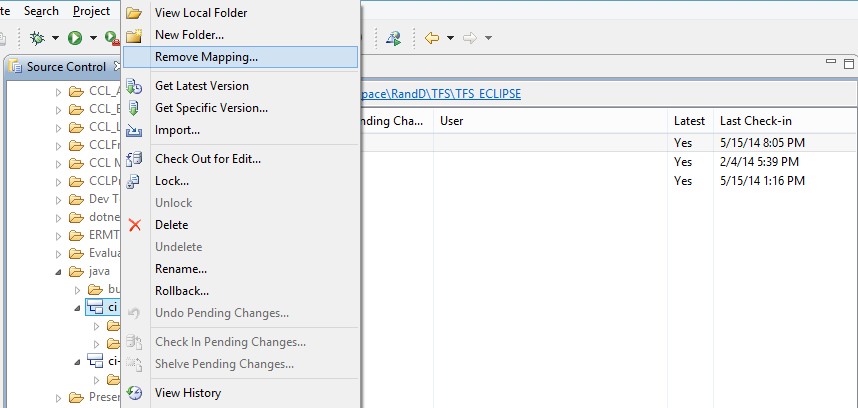


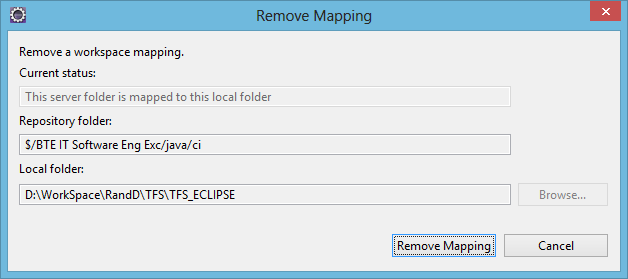
1. Get Files Progress window.



## Remove mapping

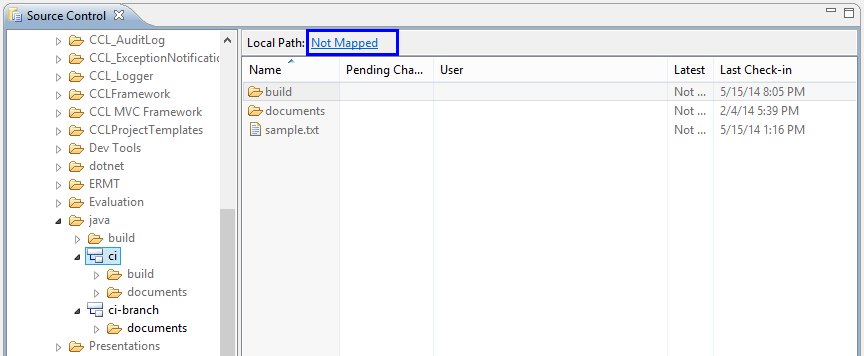
1. Go to Source Control Explorer 🡪 Right click project name 🡪 Remove mapping.



1. Select Remove mapping.
2. Deleting working folder mapping.

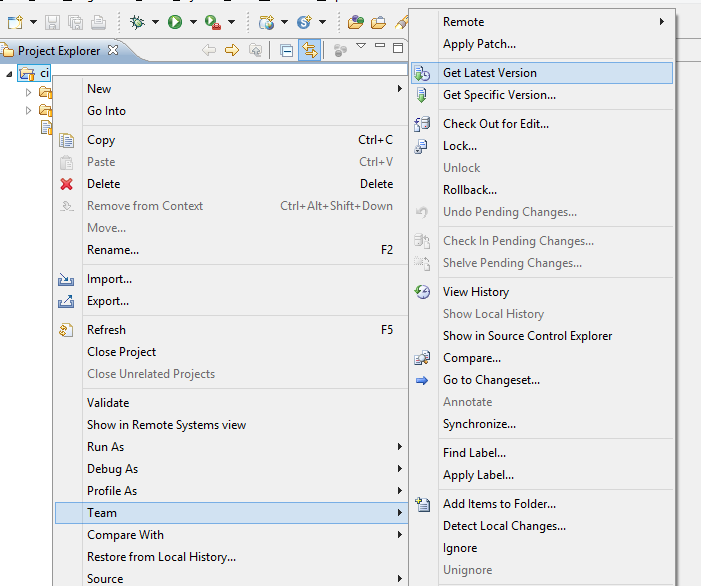


1. Thus the below window shows that there are no mapping to the Repository Folder and the Local Path.

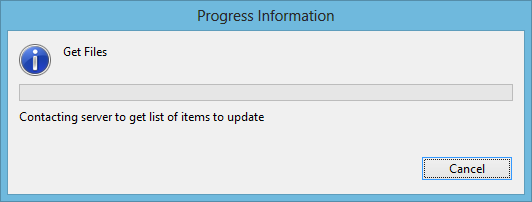


## Get Latest

1. Go to Project Explorer 🡪 Right click project name 🡪 Team 🡪 Get Latest Version.



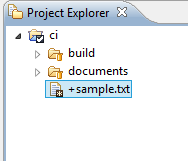
1. Progress window



1. Latest files would be replaced with the local folder that is being mapped.

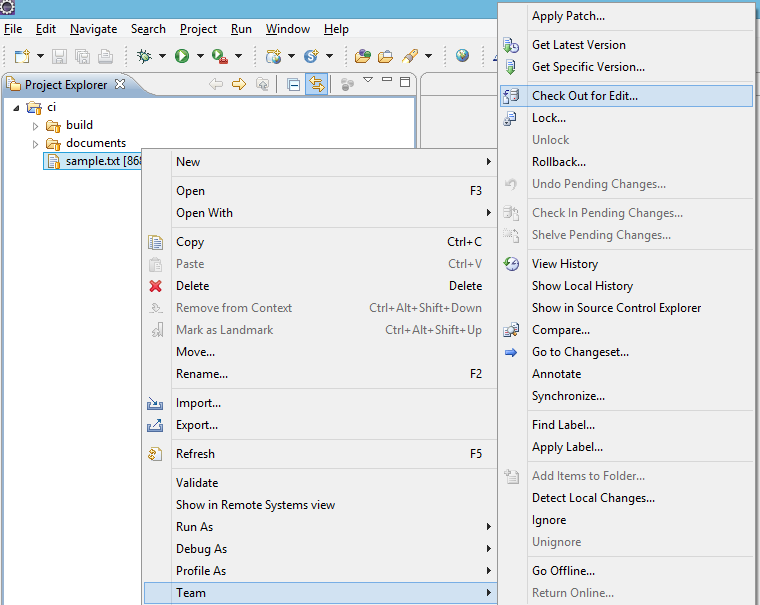
## Add Items to Folder

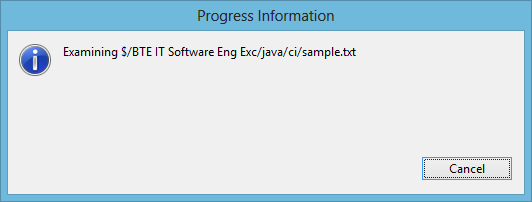
1. Go to the Project Explorer.
2. Add a new file named sample.txt to the imported project.



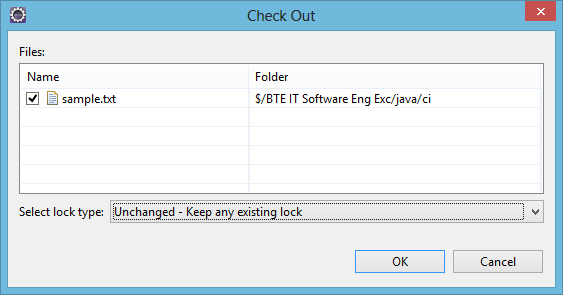
1. You could see a + symbol for the newly added file/folder.
2. To add permanently, we must check-in the added file to TFS.

## Checkout

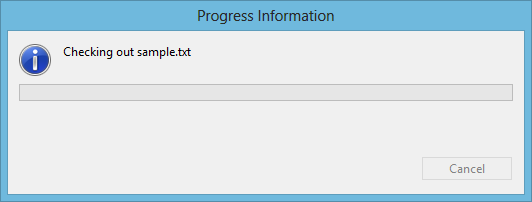
1. Right click Project name 🡪 Team 🡪 Check Out for Edit…
2. Progress Window.



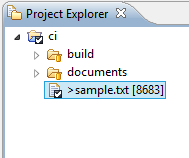
1. Check the list of Items to Check out.



1. Progress Window.



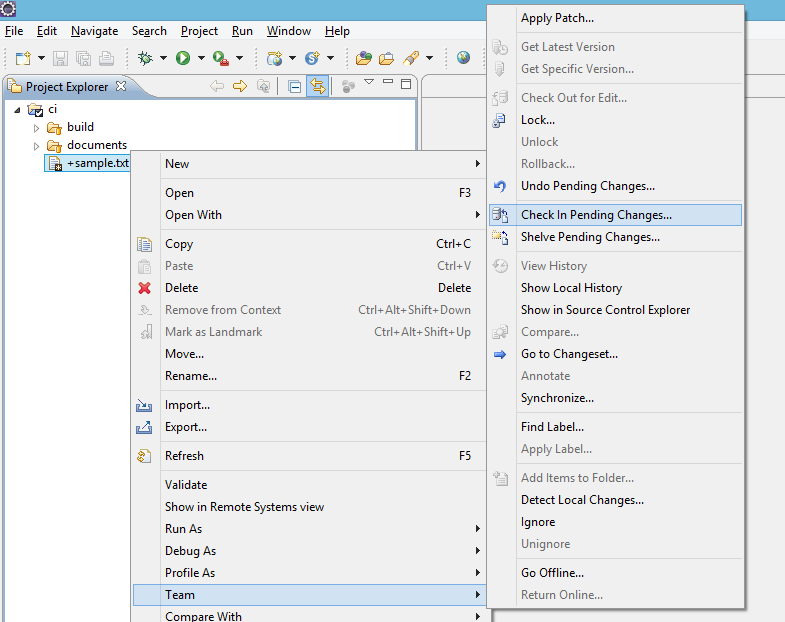
1. Once the Checkout Progress has been completed, the checked out item displays the status as shown below.

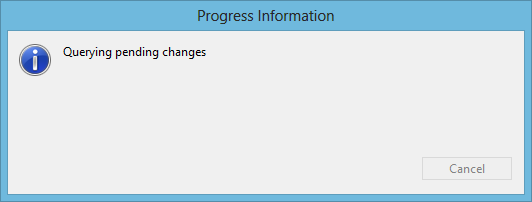


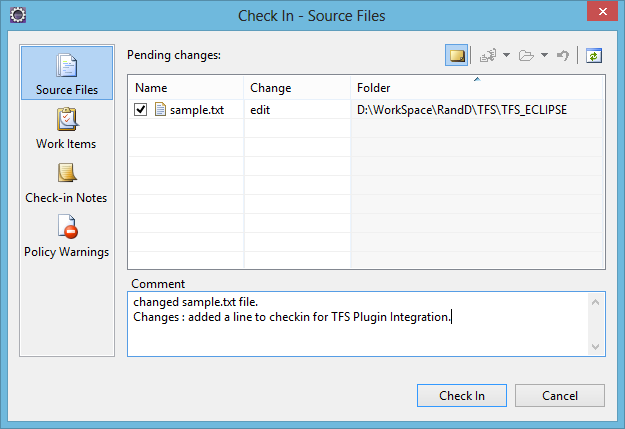
## Check-in

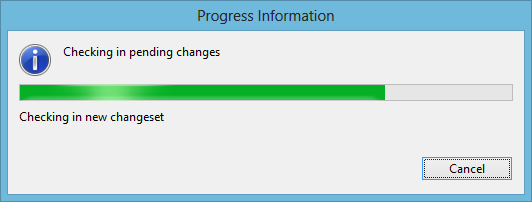
### Using Project Explorer

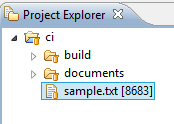
1. Right click on project name 🡪 Team 🡪 Check In Pending Changes... Make sure there is a change for the item that you are going to check-in.



1. Progress Window.
2. Check the Items to Check In. Provide valuable comment for each check-in.

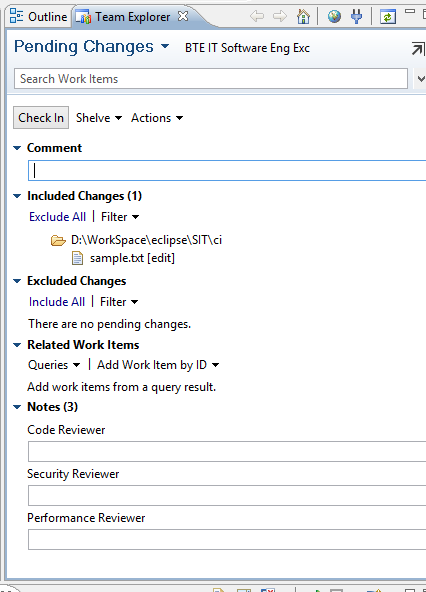


1. Progress window.
2. Items checked in as shown below with the changeset number as shown below. E.g. 8683 is the Changeset number.

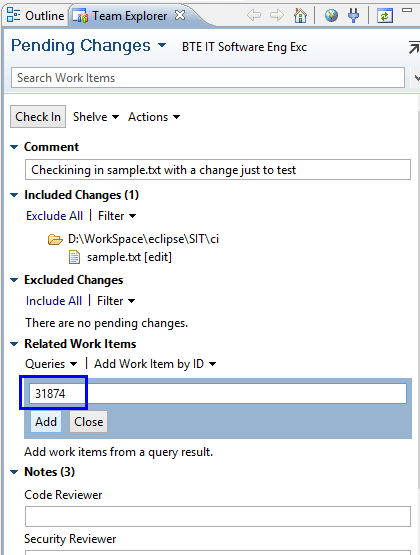


### Using Team Explorer

1. Once you have a file/folder ready to check-in. Go to Team Explorer.



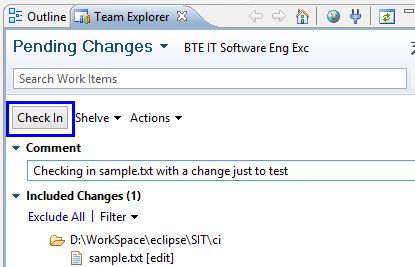
1. Provide a valuable comment at the comment section.
2. Provide the work item ID to relate to the work item.



1. We could see the work item getting related as shown below.



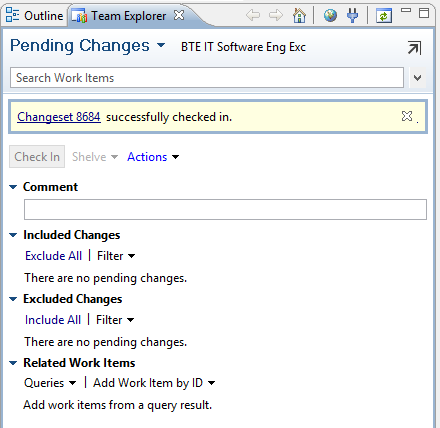
1. Finally click check in.



1. Progress window.

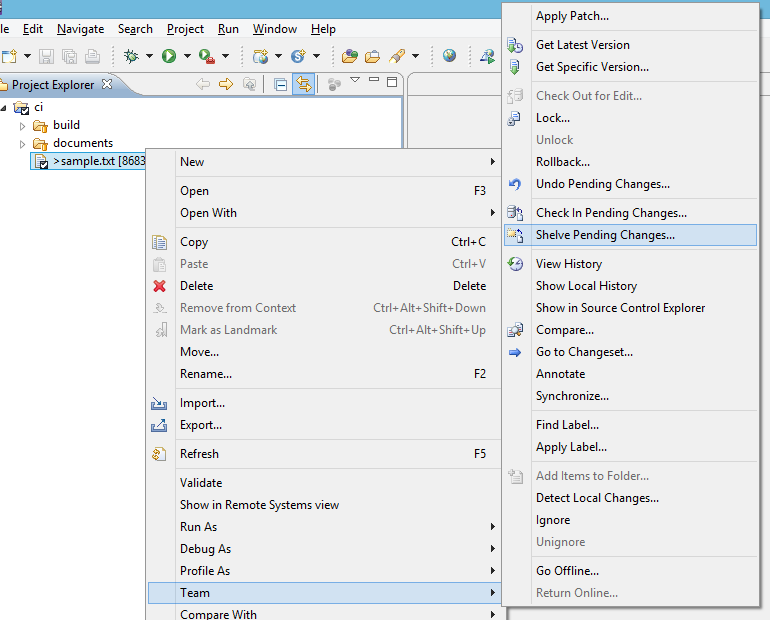


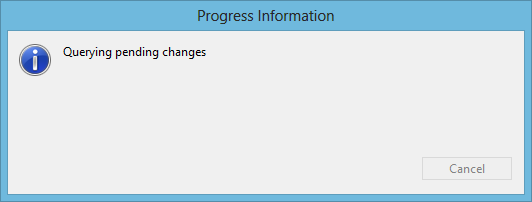
1. After check-in completes you could see the changeset number which will be generated for each operation performed in TFS.

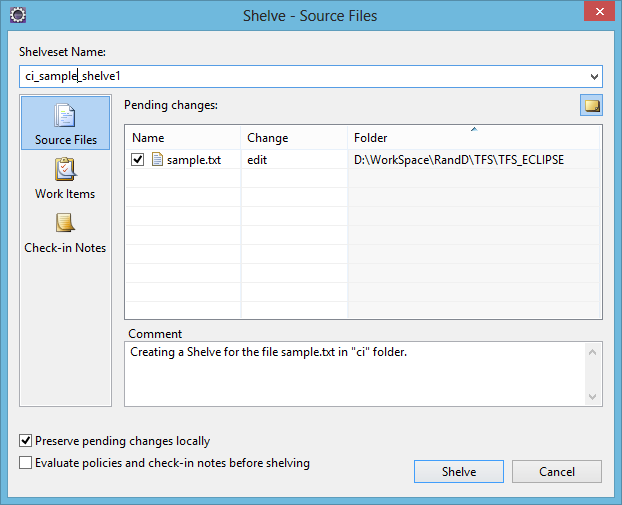


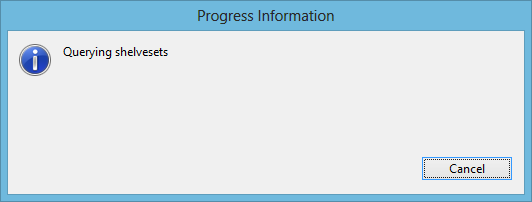
## Shelve

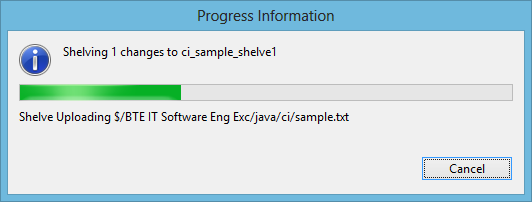
1. Right click Project Name 🡪 Team 🡪 Shelve Pending Changes…



1. Progress Window.
2. Provide the Shelveset Name and check the suitable file/folder and provide meaningful comment.



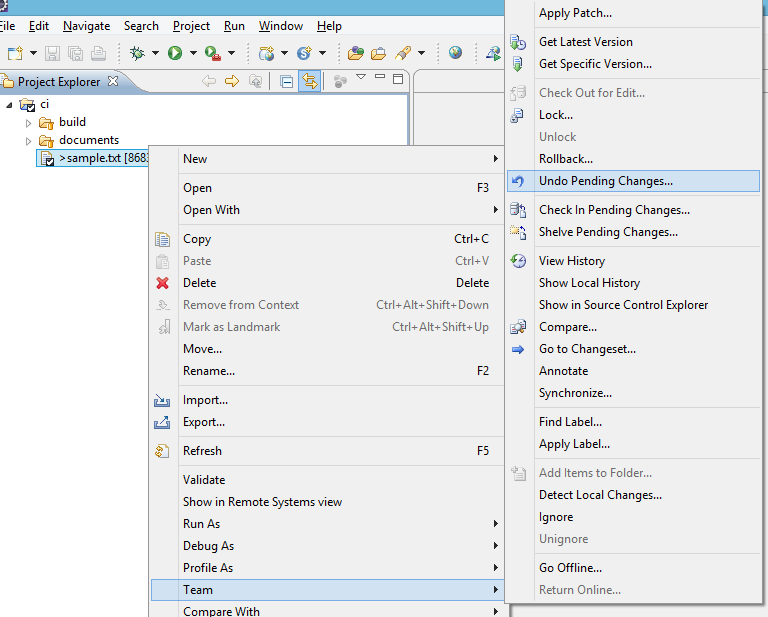
1. Progress Window.



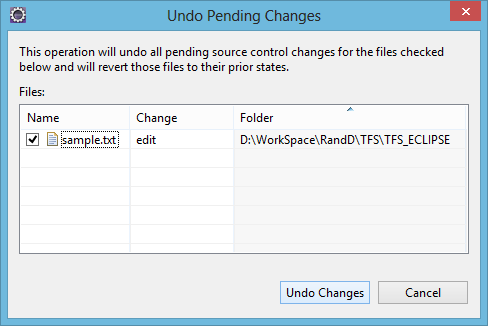
1. Once the Shelveset has been created as you can view in the console window. You can see the file/folder with status of Pending Change as “edit”. So need to [Check-in](#_Check-in) the file/folder to shelve permanently to the Repository.
2. You can also perform Shelve operation [Using Team Explorer](#_Using_Team_Explorer)

## Undo Pending changes

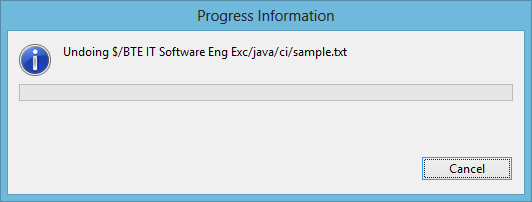
1. Right click Project Name 🡪 Team 🡪 Undo Pending Changes.



1. Check the files/folders that you want to undo the Pending changes.

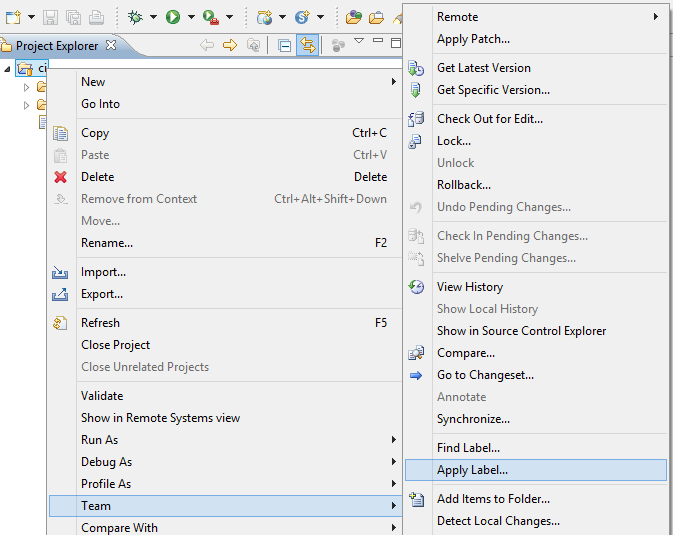


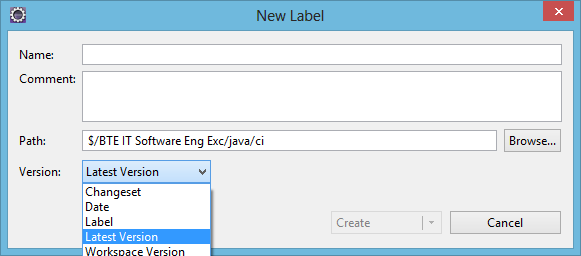
1. Progress Window.

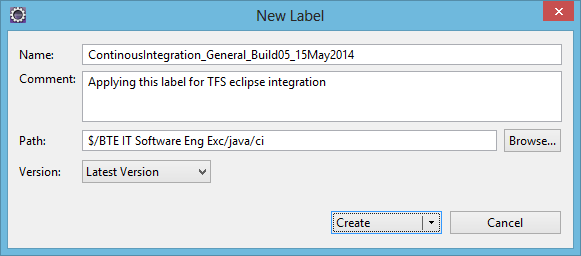


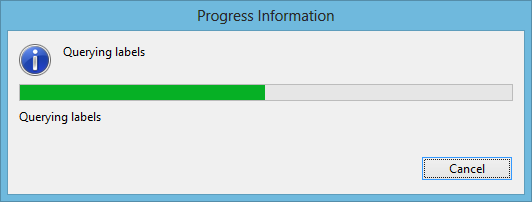
1. Once the Progress completes, the last made changes would be undone.

## Label

1. Right click Project Name 🡪 Team 🡪 Apply Label...
2. Select the option Version (Latest Version in this case).



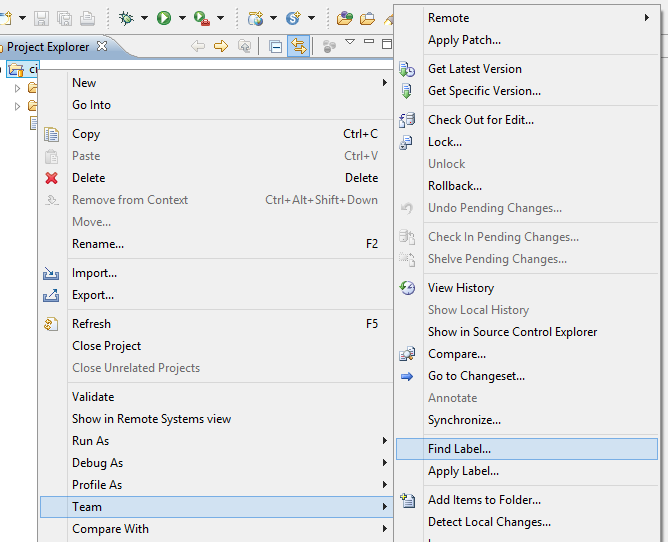
1. Provide the name for new label and its Target location and valuable comment.
2. Progress Window.



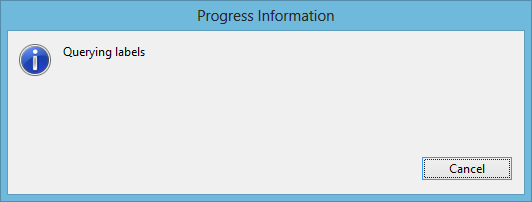
1. Label has been applied.

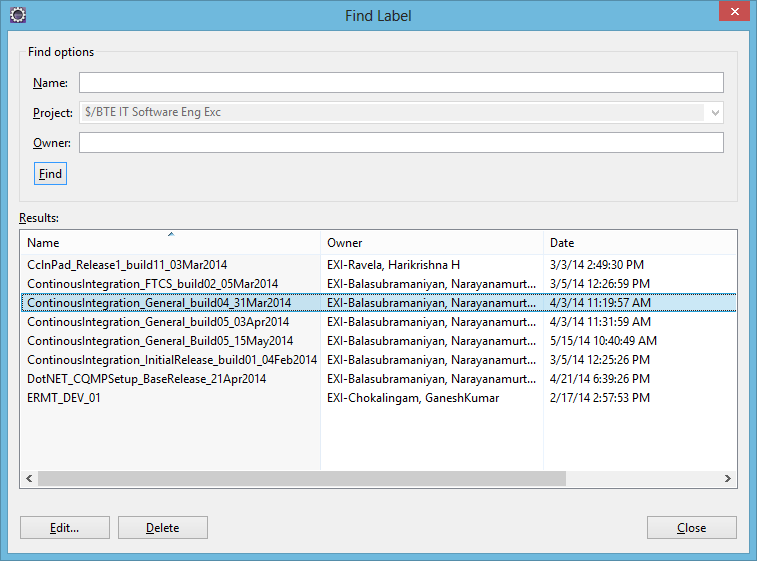
## Find Label

1. Right click Project Name 🡪 Team 🡪 Find Label…

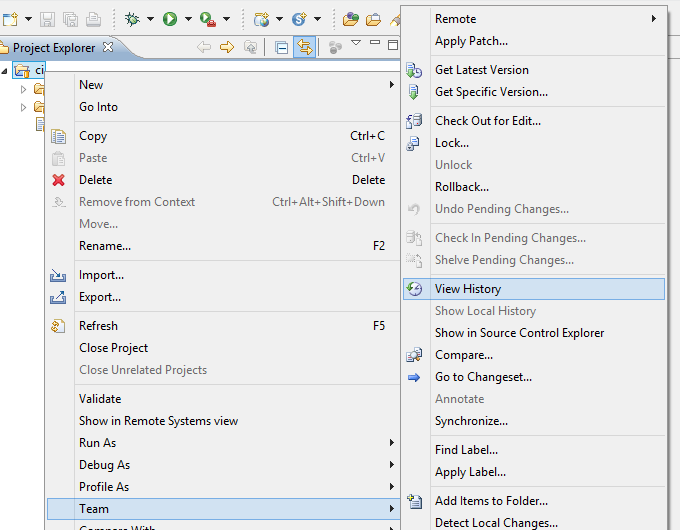


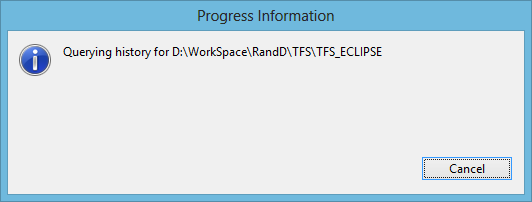
1. Click Find.
2. Progress Window.



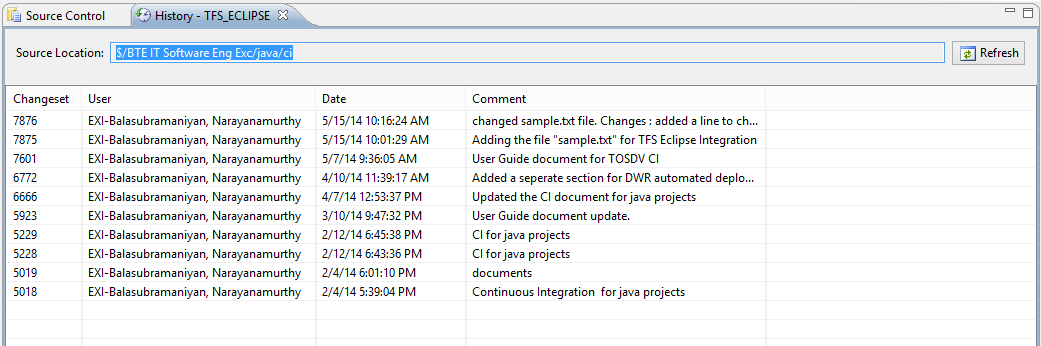
1. It displays the list of labels, its owner and the date when it is created.

## History

1. Right click Project Name 🡪 Team 🡪 View History
2. Progress Window

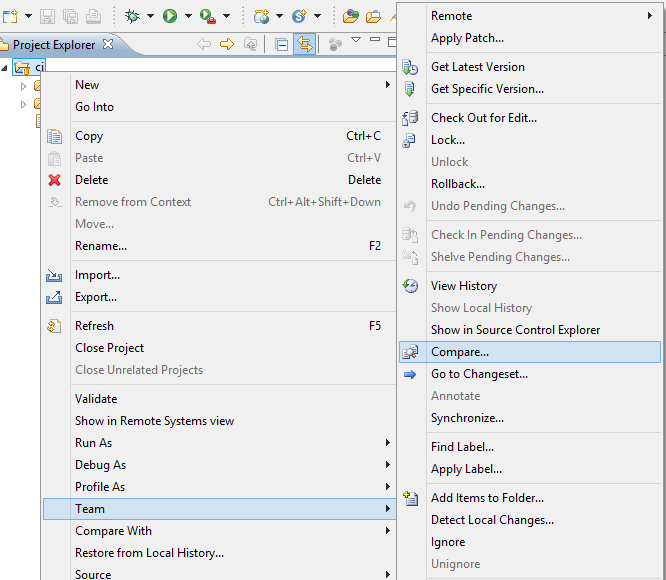


1. It displays the history of the particular project.

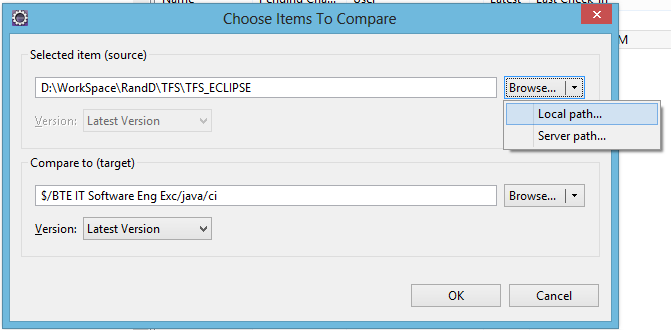


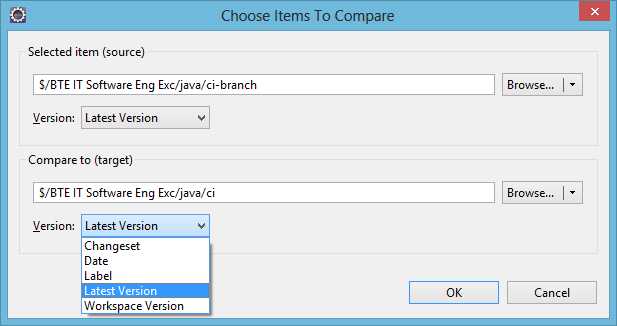
## Diff

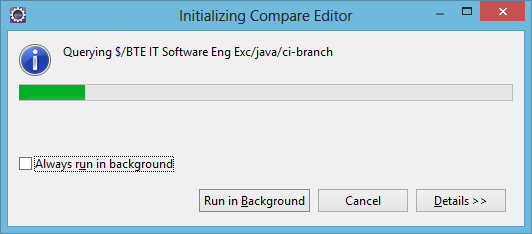
1. Right click Project Name 🡪 Team 🡪 Compare...



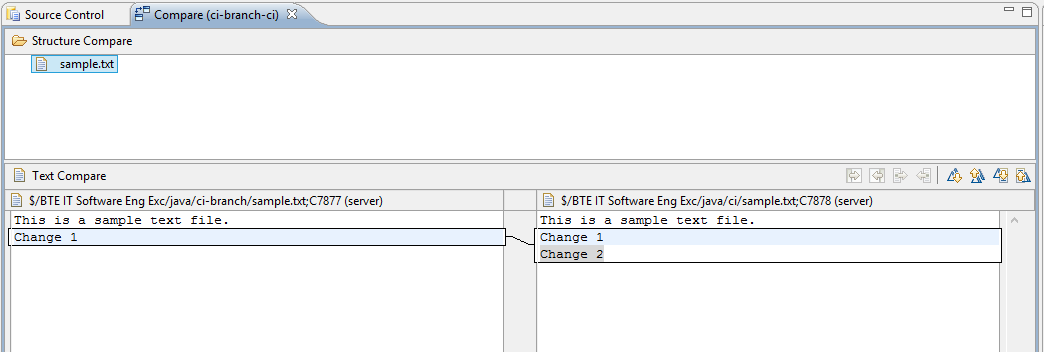
1. Click Browse to provide the path for source and target to be compared.



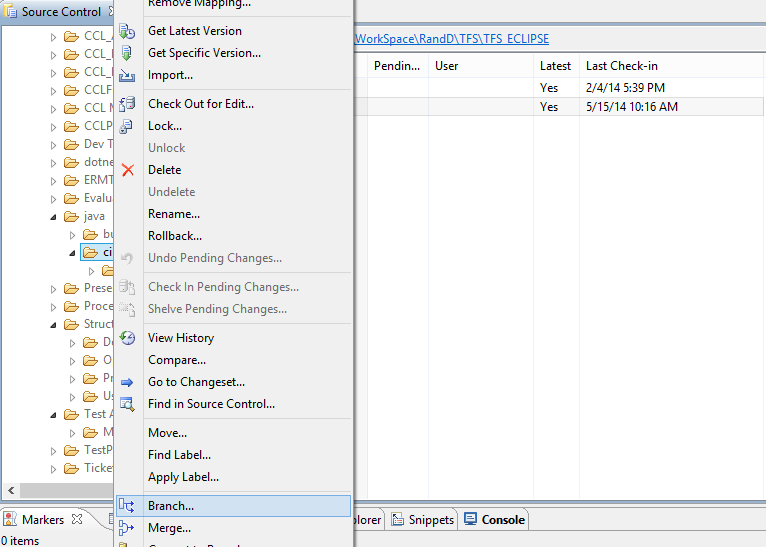
1. Provide the target location of the source and target. Select the option Version (Latest Version in this case).
2. Progress window.

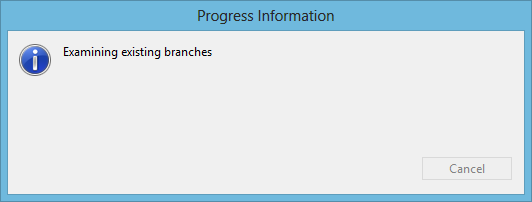


1. Diff page appears. Left side indicates the Source and Right side indicates the Target. You can see the difference between the source and target as shown below.

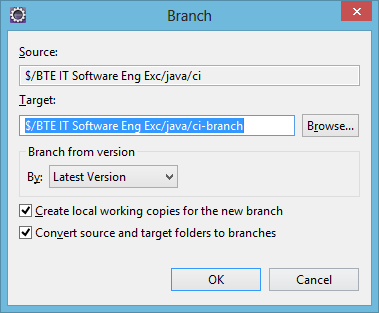


## Branch

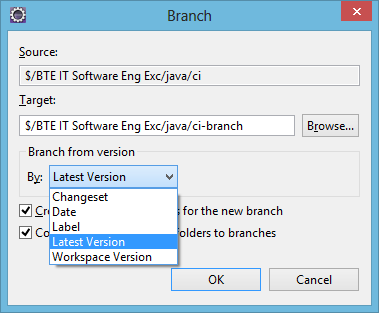
1. Go to Source Control Explorer 🡪 Right click Project Name 🡪 Branch.
2. Progress Window.

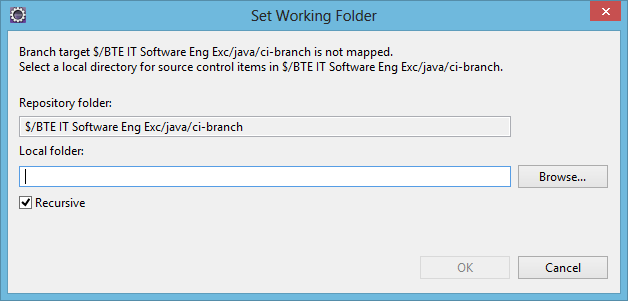


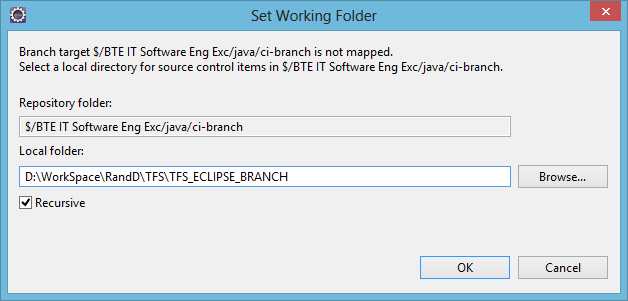
1. Provide the name for new branch (ci-branch) and its Target location.

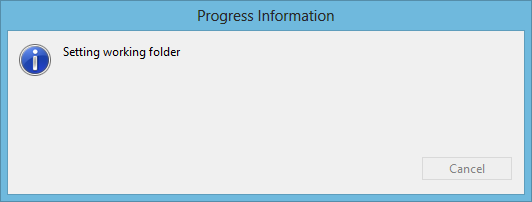


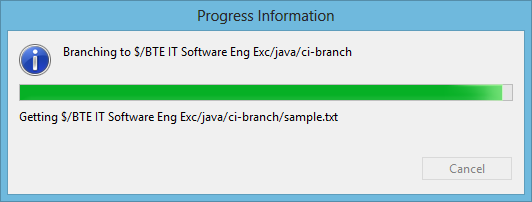
1. Select the option Branch by Version (Latest Version in this case).

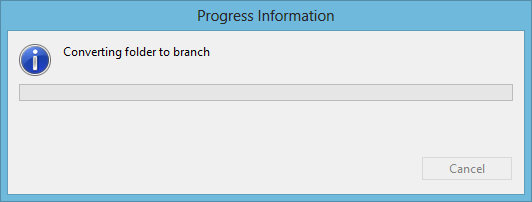


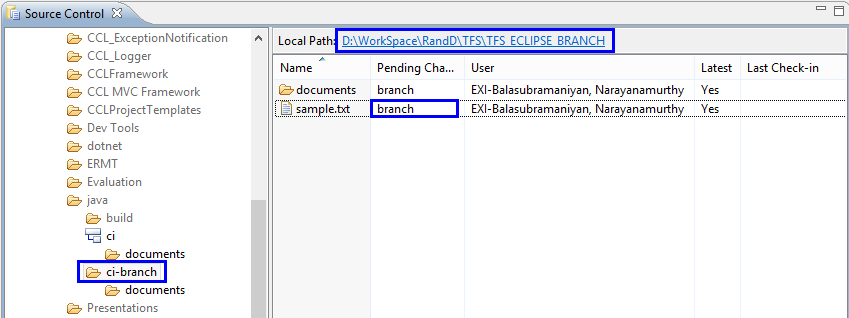
1. It prompts to set the Working Folder.
2. Browse the path of the Working Folder in “Local Folder” and map it to the repository folder.

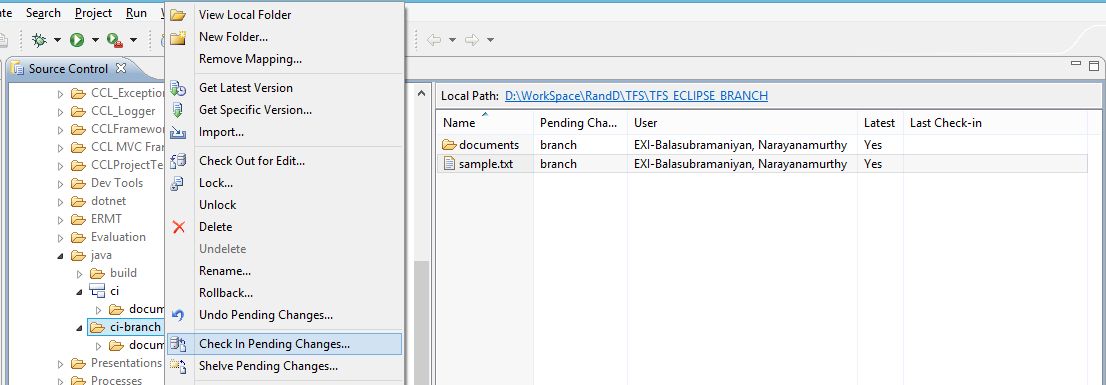


1. Progress Windo

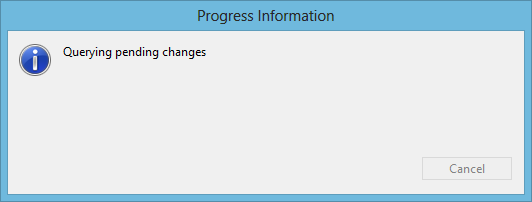




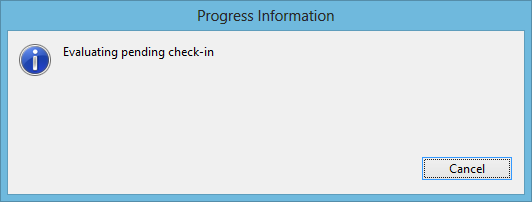
1. Go to Source Control Explorer 🡪 Project Name. The New Branch created displays the status of Pending Change as “branch”. You have to check-in the newly created branch to add permanently to the Repository Folder.
2. Go to Source control Explorer 🡪 Right click Branch Name 🡪 Check In Pending Changes…

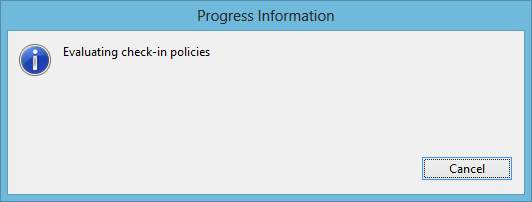


1. Progress Window

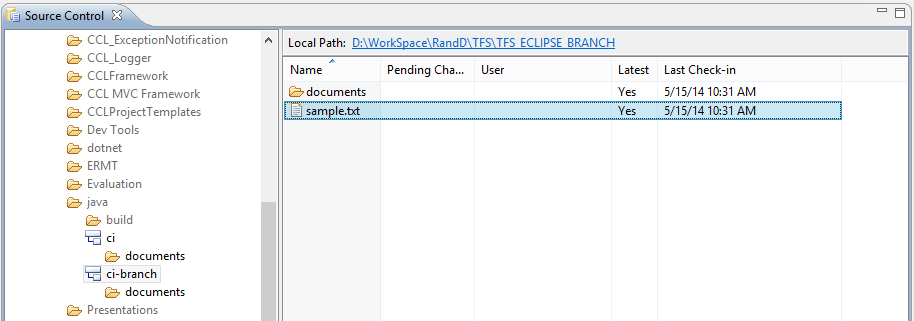


1. Check the Items to Check In. Provide valuable comment for each check-in
2. Progress Window

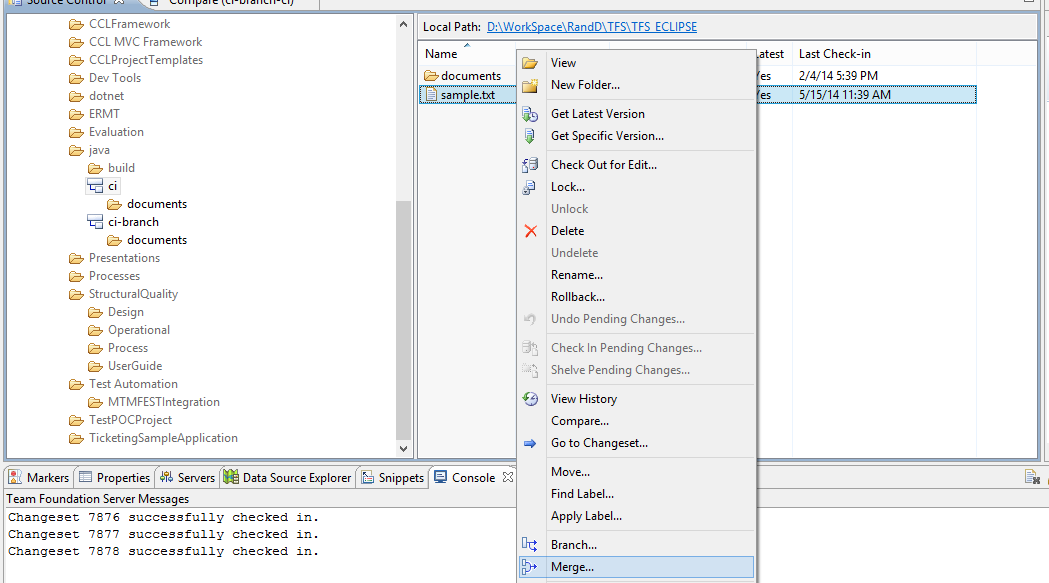
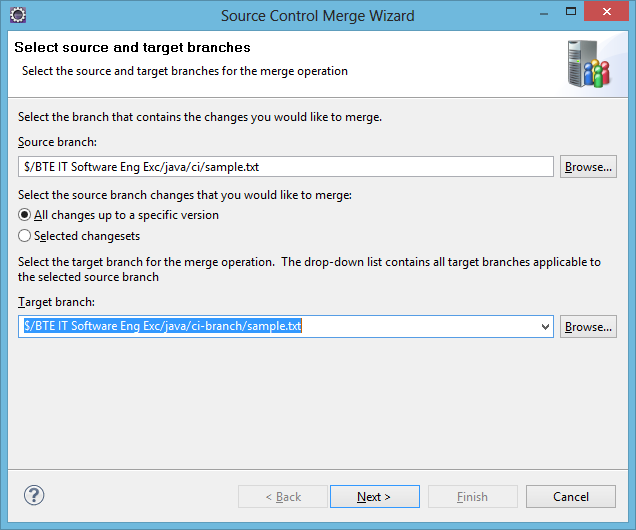
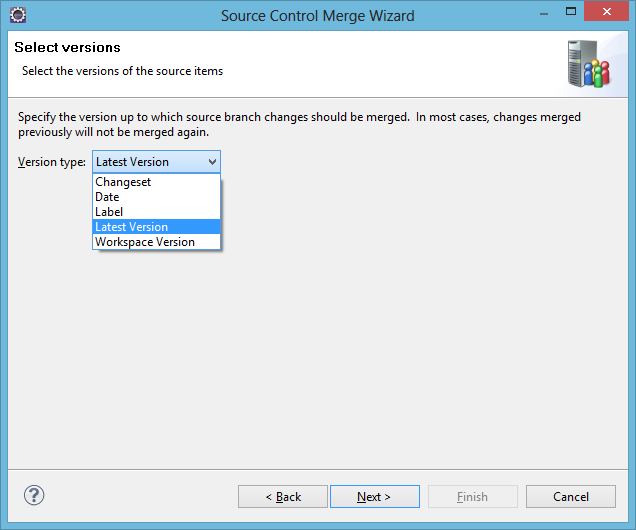
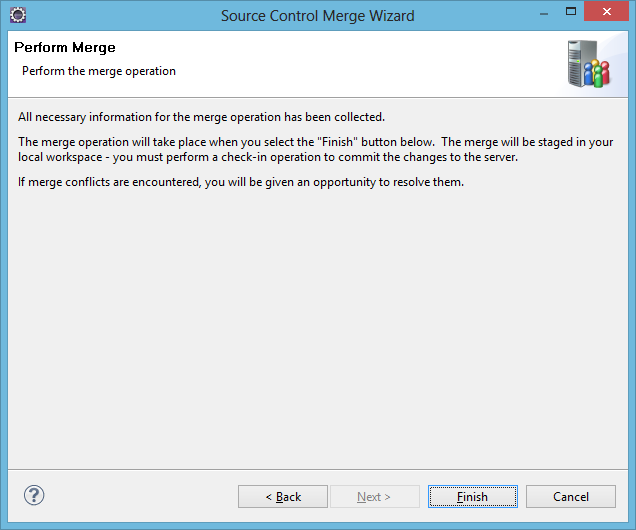


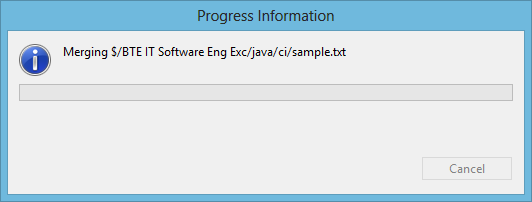


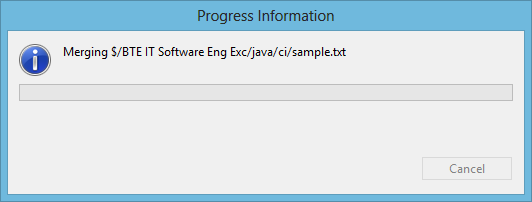
1. Items checked in as shown below.



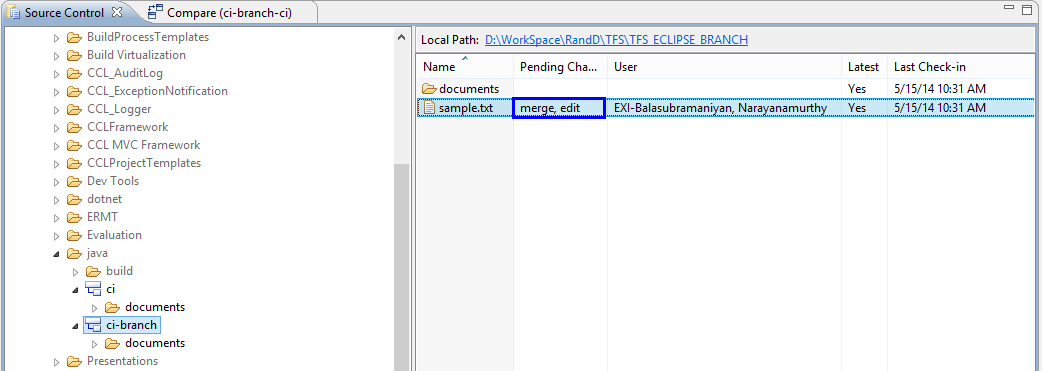
## Merge

1. Go to Source Control Explorer 🡪 Project Name 🡪 Right click file/folder name 🡪 Merge…
2. Browse the path of the Target Branch/Folder and click Next.
3. Select the option Version type (Latest Version in this case).
4. Click Finish
5. Progress window

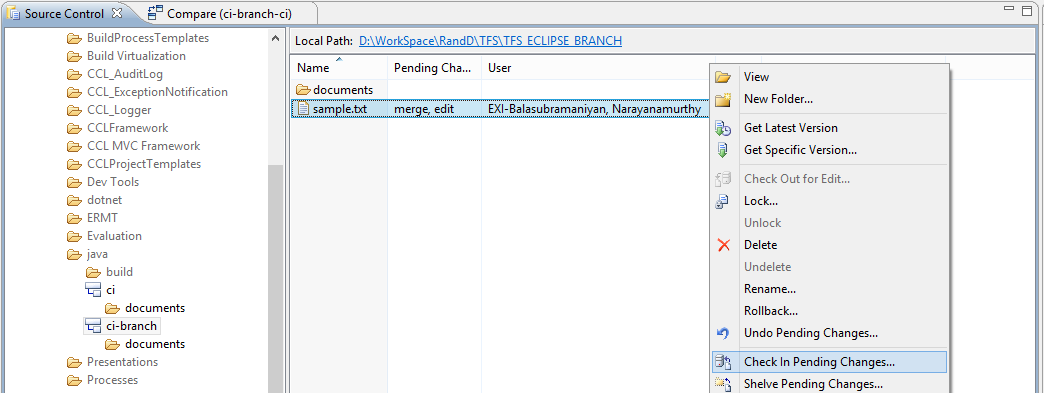


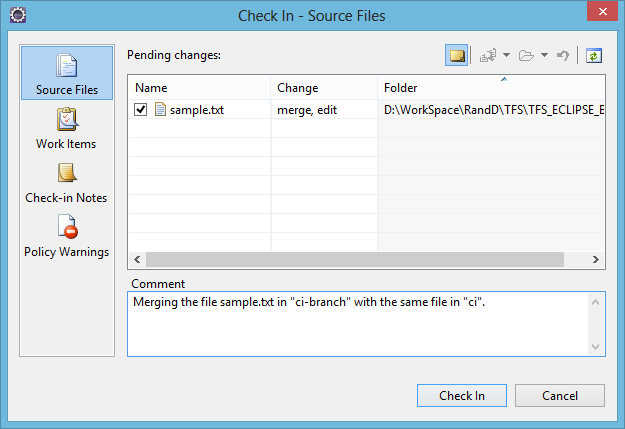


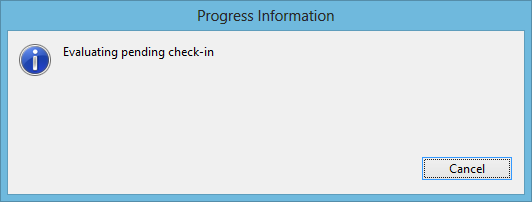
1. Once the progress bar is completed, the Pending change column changes to “merge, edit” as shown below. You have to check-in the merged file/folder to merge permanently to the Repository Folder.

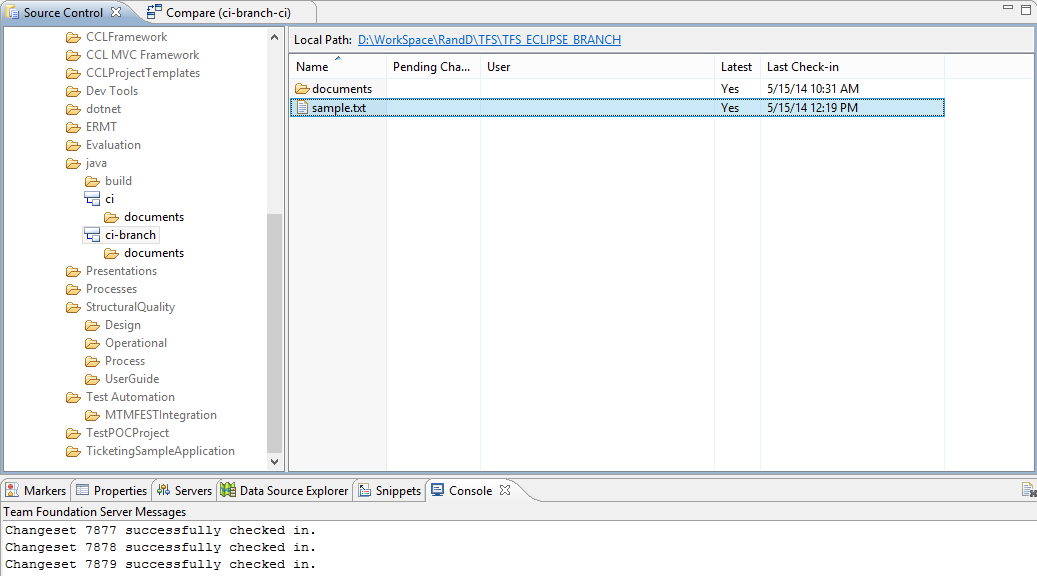


1. Go to Source Control Explorer 🡪 Project Name 🡪Right click file/folder name 🡪 Check In Pending Changes…



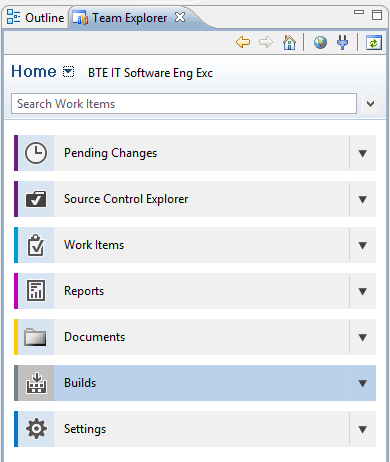
1. Check the suitable merged file/folder for check-in. Provide valuable comment.
2. Progress Window.

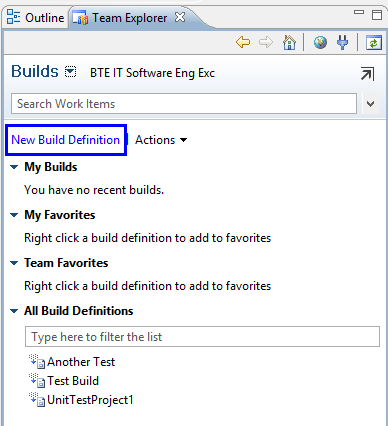


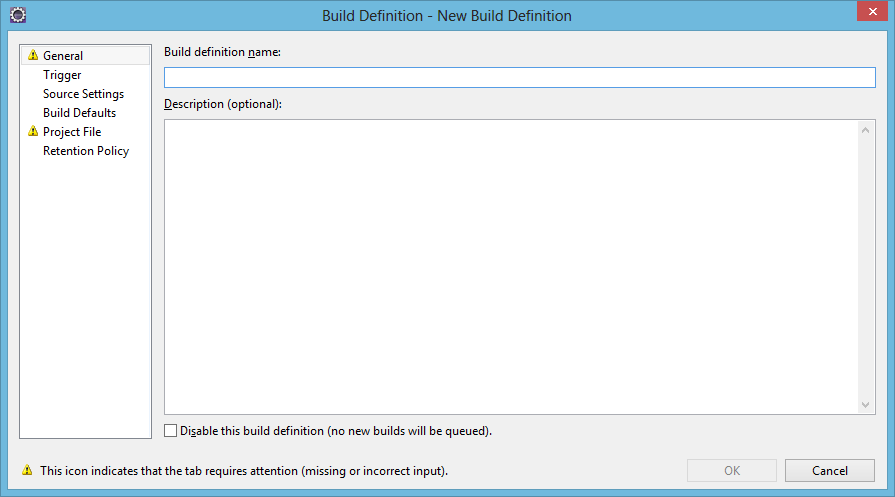
1. Thus the file/folder merged as shown below.

## Build definition

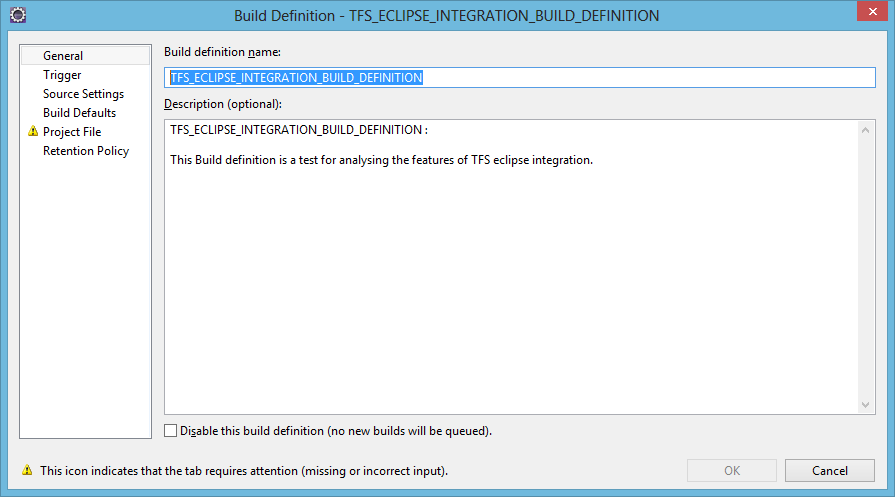
1. Go to Team Explorer 🡪 Builds.



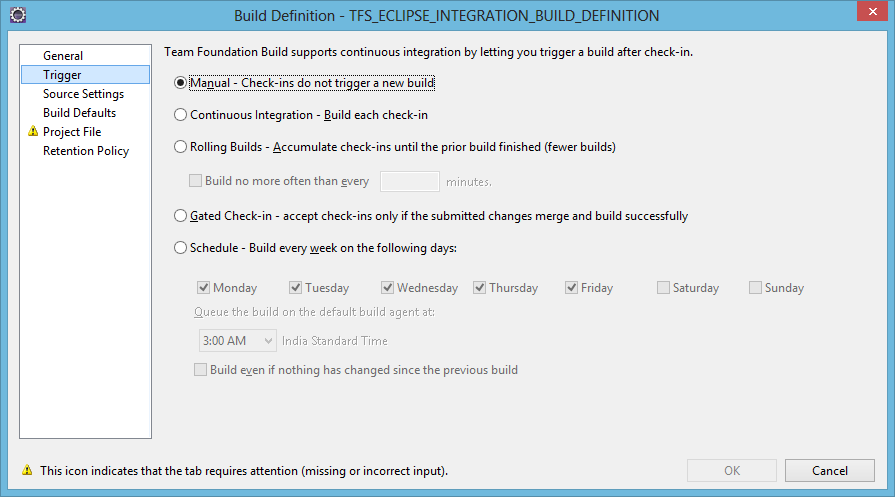
1. Go to New Build Definition as shown below.
2. A New Build Definition Window appears.

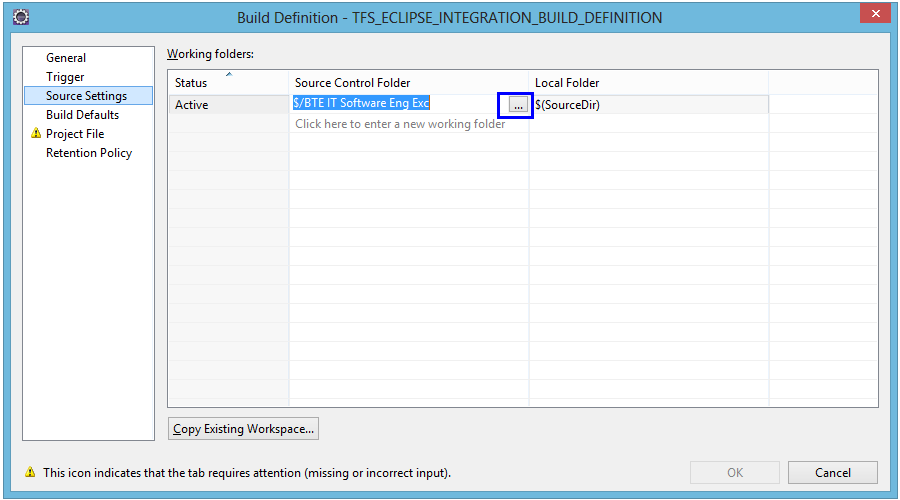


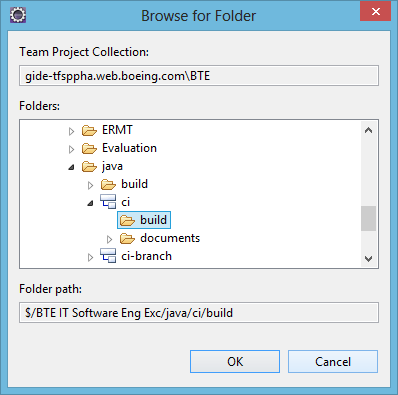
1. Go to General Tab. Provide the name of the Build Definition and the meaningful description for the Build Definition.

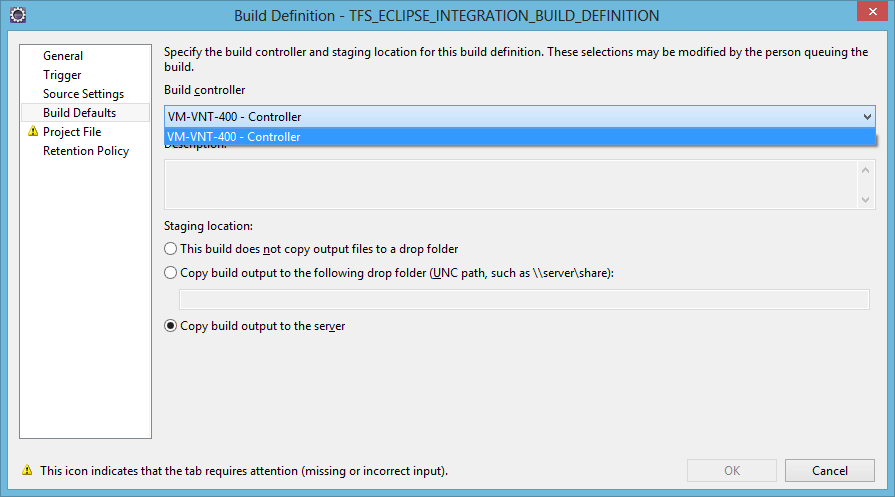


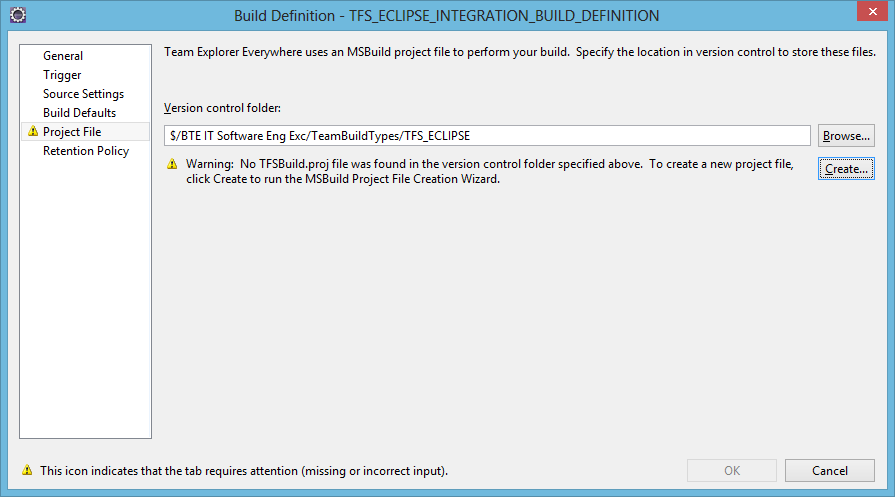
1. Go to Trigger Tab. Check the “Manual – Check-ins do not trigger new build”.

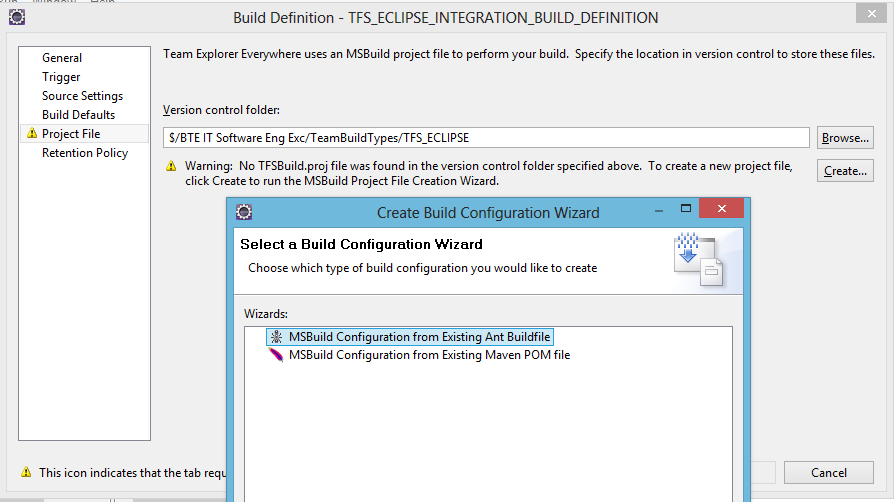


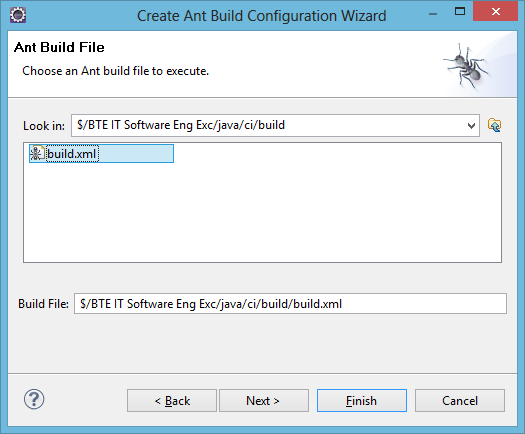
1. Go to Source Settings Tab. Click the area under “Source Control Folder” as shown below.
2. Browse the required folder from the Team project Folder path as shown below.



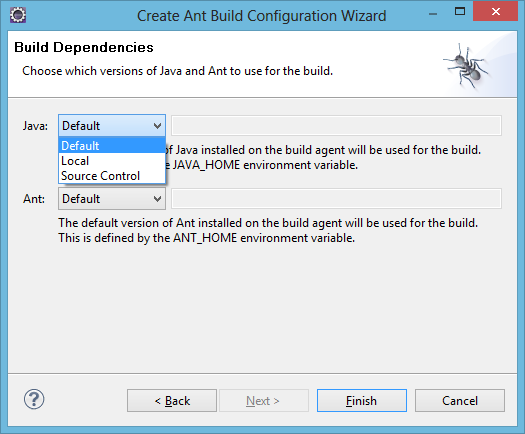
1. Go to Build Defaults Tab. Select the suitable Build Controller from the drop down menu. Select the suitable Staging location.
2. Go to Project File Tab. Click Create if you don’t have TFSBuild.proj file in the version control folder as shown below.

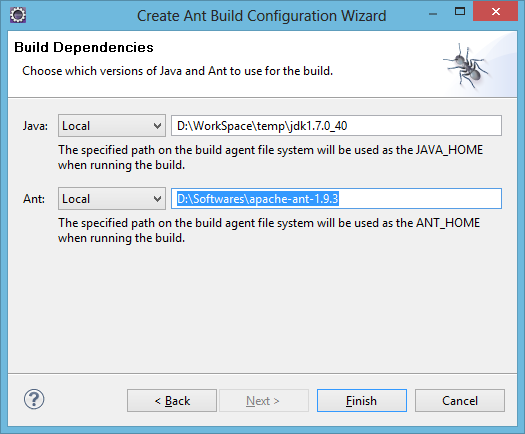
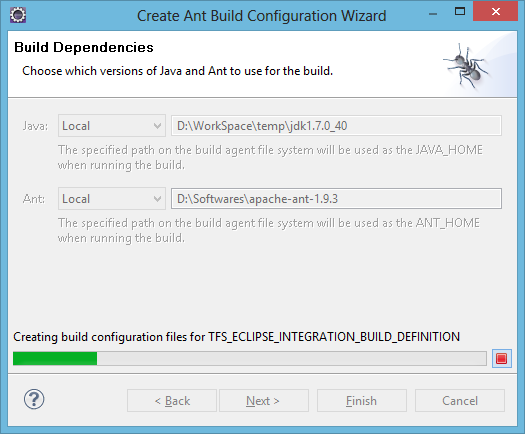


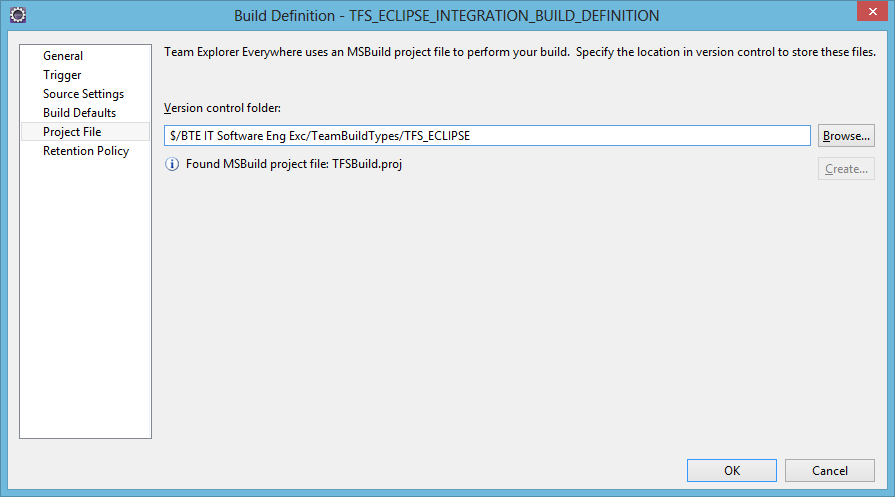
1. Select the suitable Build Configuration from the “Create a Build Configuration Wizard” that appears. “MSBuild Configuration from Existing Ant Buildfile” in this case.
2. Choose the required file as shown below. Build.xml selected in this case. Click Next.



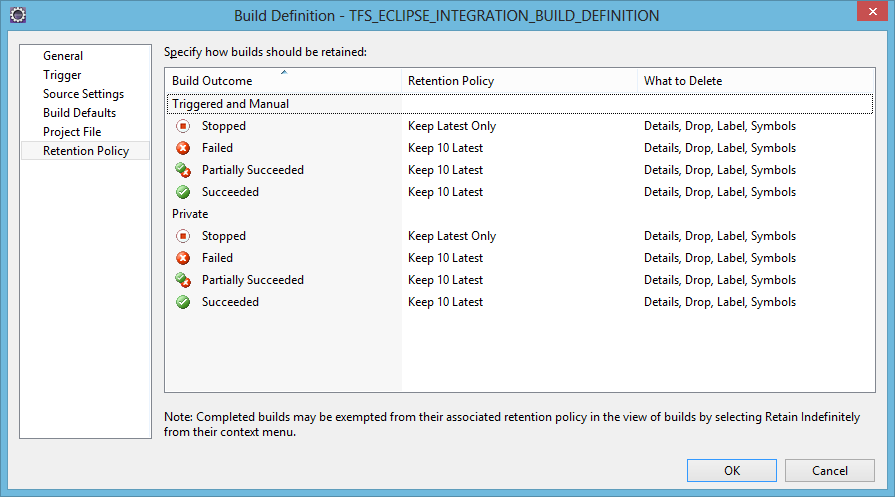
1. Build Dependencies Window appears. Select the suitable option for Java, ANT, MAVEN , etc., as shown below.



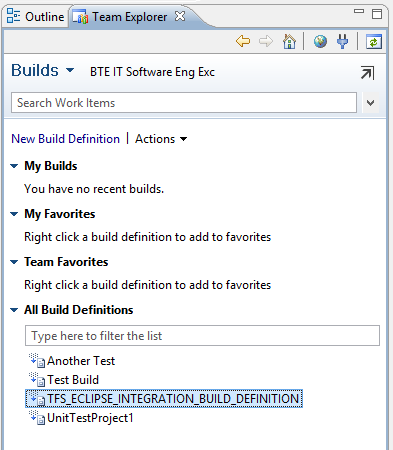
1. Selected Local for Java and Ant and its Installation folder path has been specified as below.
2. Progress Window.
3. TFSBuild.proj has been created as shown below.



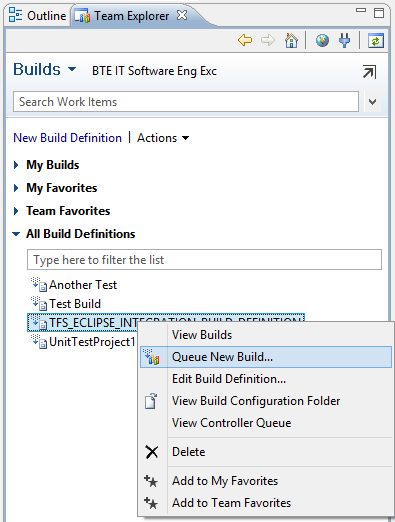
1. Go to retention policy tab. It shows how builds should be retained.

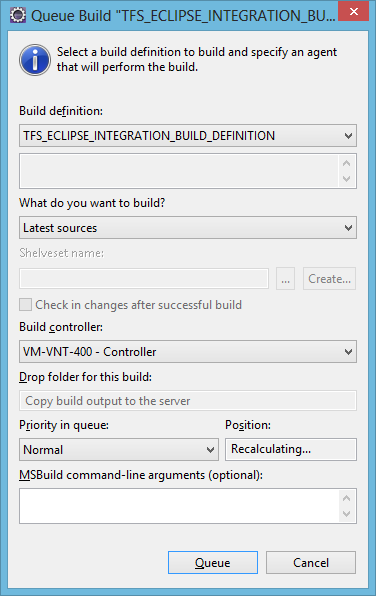


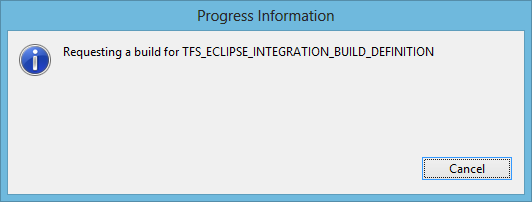
1. Go to Team Explorer. Newly created build definitions appears as shown below.



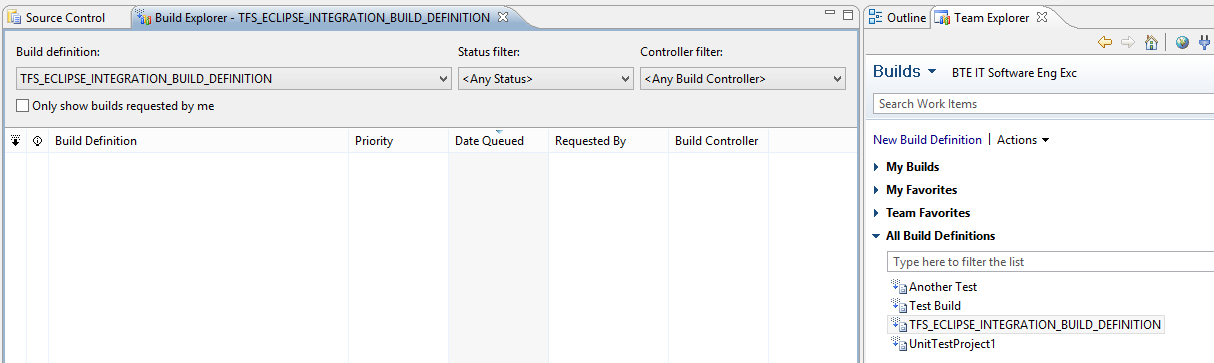
1. Go to Team Explorer 🡪 Build Definition Name 🡪Queue New Build…



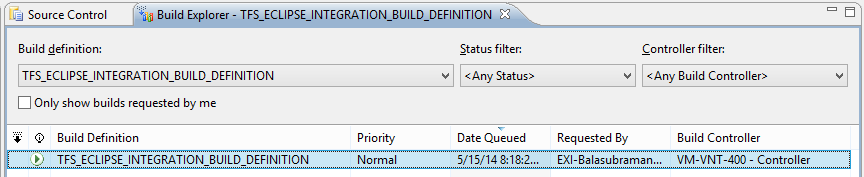
1. A Queue Build window appears. Click Queue.
2. Progress Window.



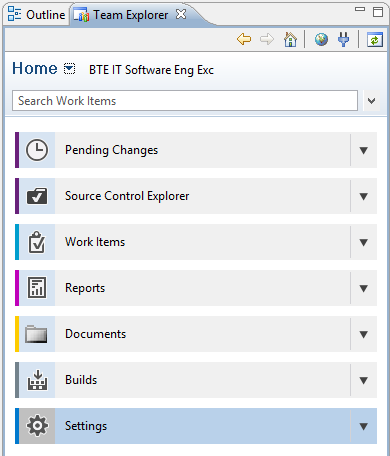
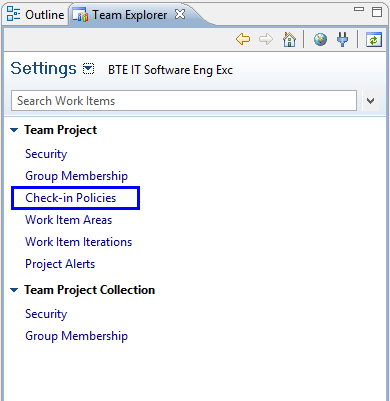
1. The Build Explorer window appears as shown below.

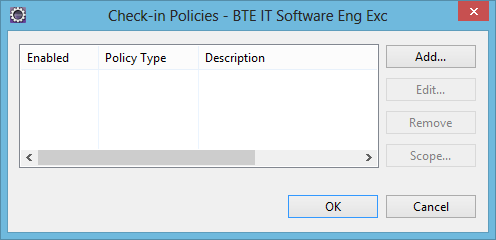


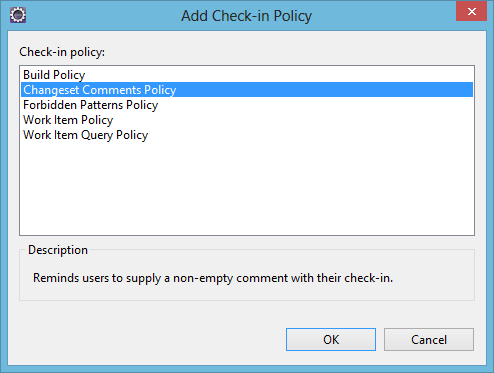
1. Status of Build Definition and log can be seen by double clicking the Build Definition Name as shown below.

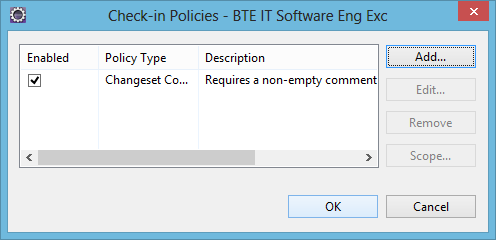


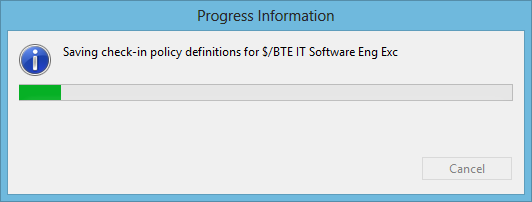
## Gated Check-in

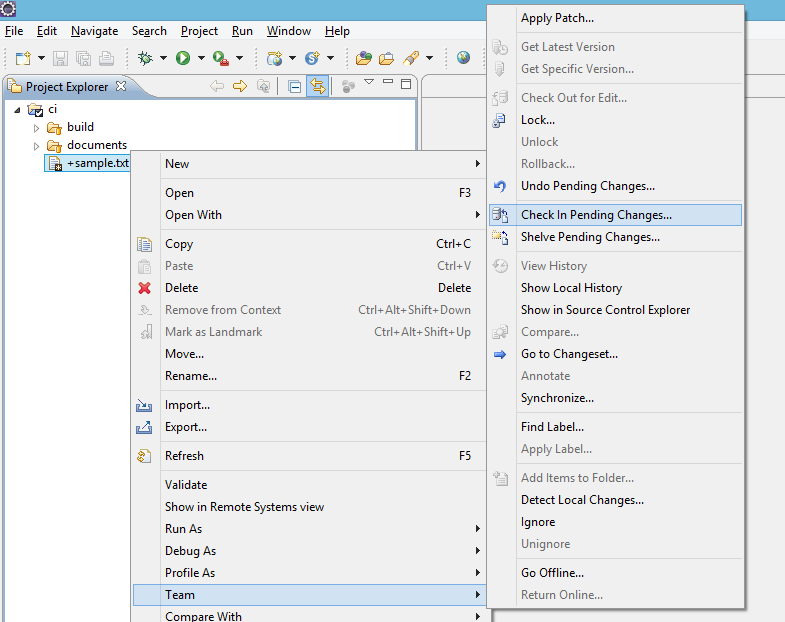
1. Go to Team Explorer 🡪 Settings.
2. Go to Team Project 🡪 Check-in Policies as shown below.
3. Check-in Policies Window appears. Click Add.

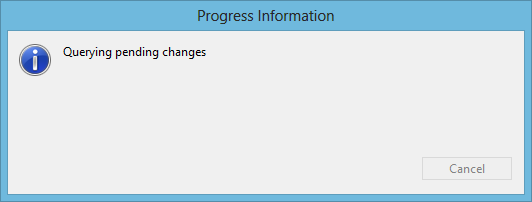


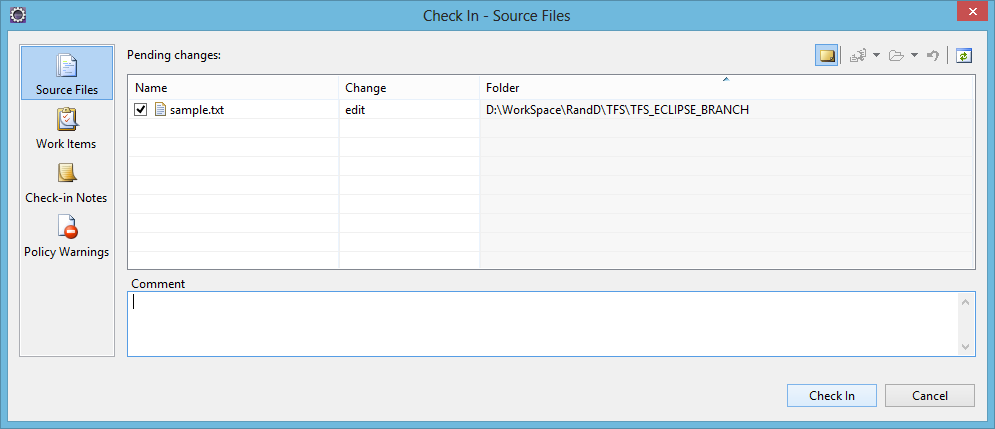
1. Choose the required Check-in policy from the drop down list and click OK.
2. Check-in Policy added as shown below. You can also add multiple Check-in policy to restrict User with several gates before each check-in.



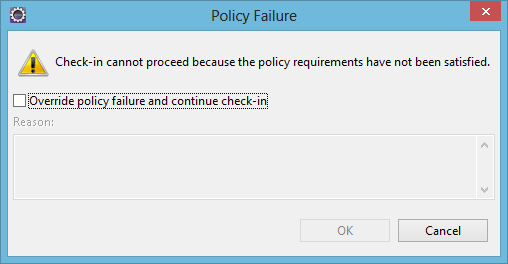
1. Progress Window
2. Right click Project Name 🡪 Team 🡪 Check In pending Changes…



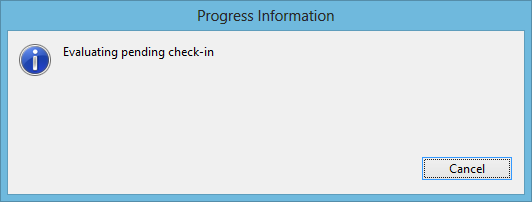
1. Progress Window.
2. Check the items for check-in. Try clicking Check In without providing the comment.

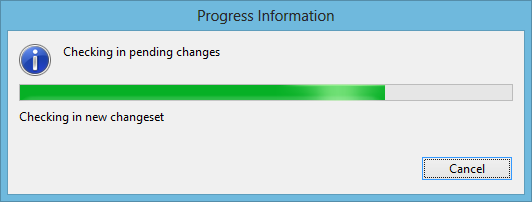


1. A new window appears as Policy Failure. It displays that Check-in cannot proceed because the policy requirement. Click Cancel.

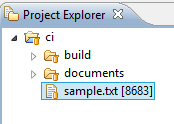


1. Now provide the comment and try Check In.
2. Progress Window.



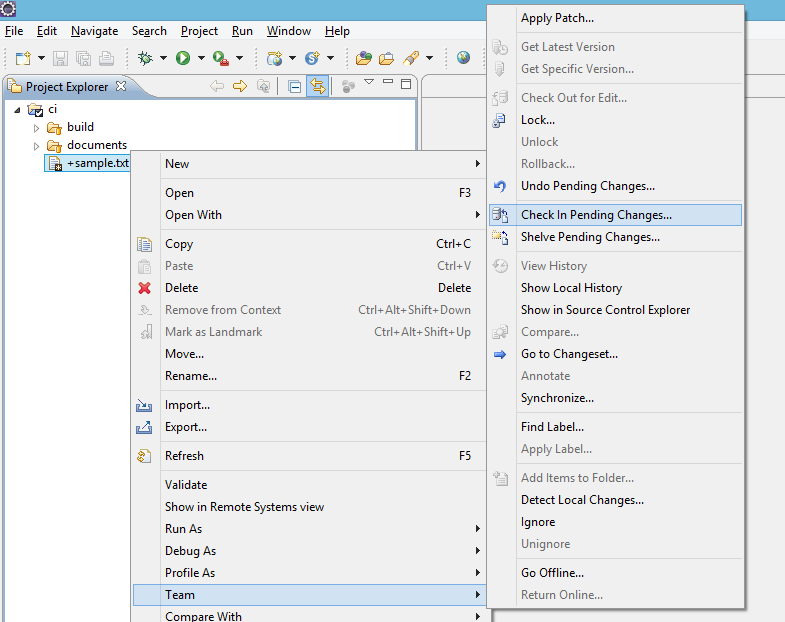


1. Successfully checked in without any policy failure.

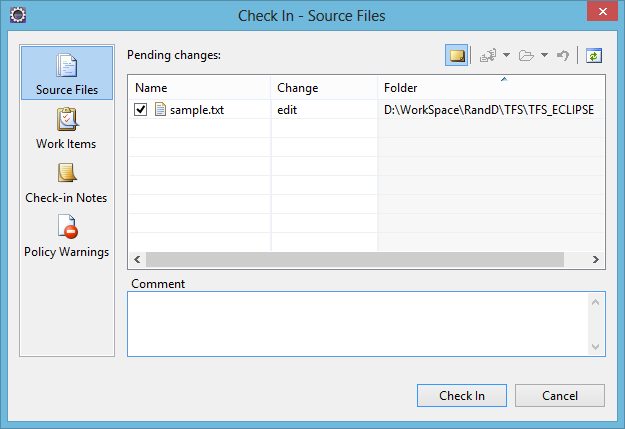


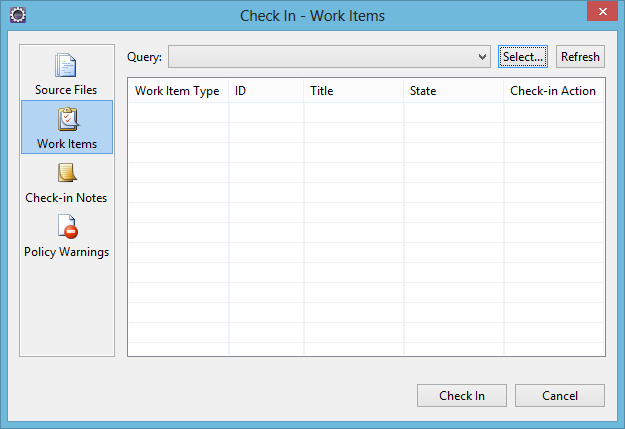
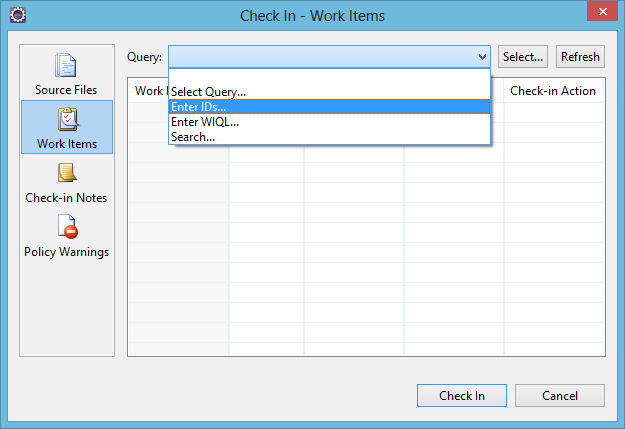
## User story traceability

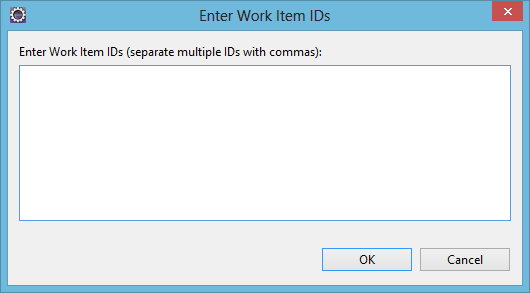
1. Go to Source Control Explorer 🡪 Project Name 🡪 Right click file/folder 🡪 Check In Pending Changes…

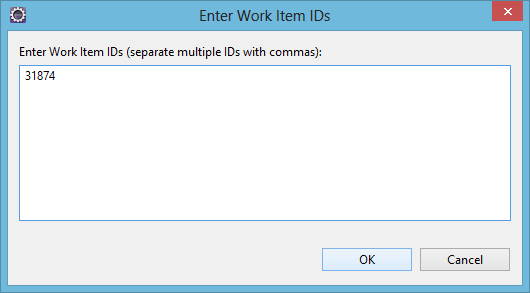


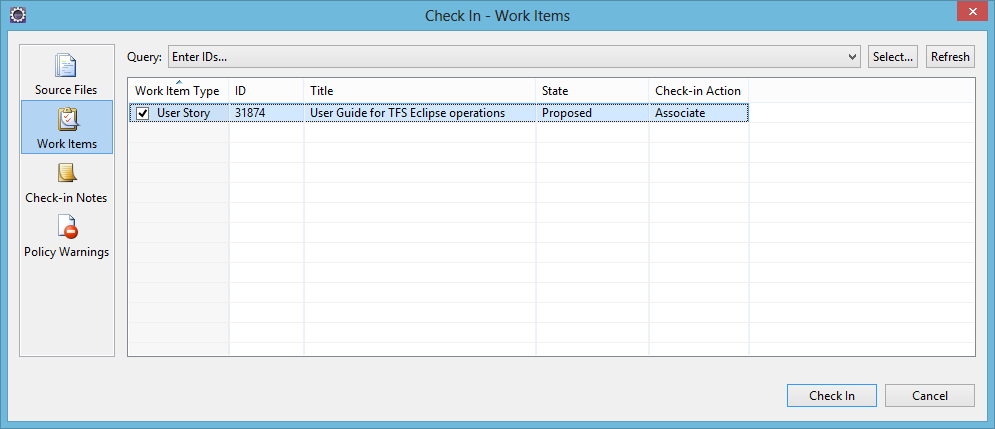
1. Check the required file/folder for check in.



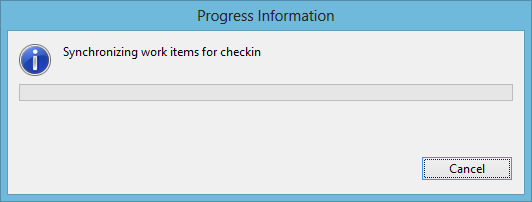
1. Go to Work Items tab.
2. Select the Enter IDs… from the Query drop down menu.
3. Enter Work Item IDs window appears.

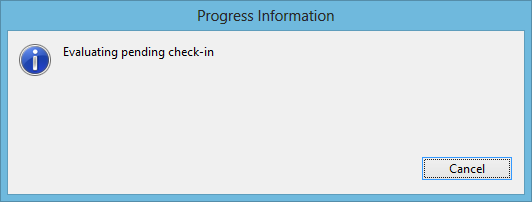


1. Enter the Work Item ID number as shown below.
2. Thus the Work Item is associated with the Check In.



1. Progress Window.

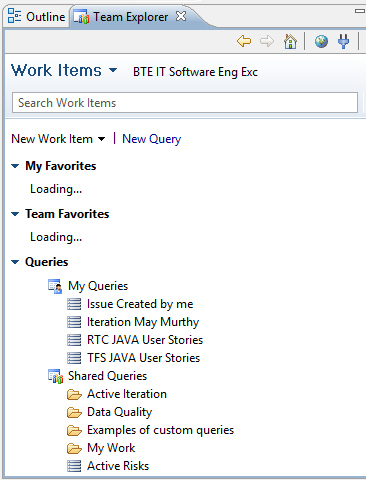






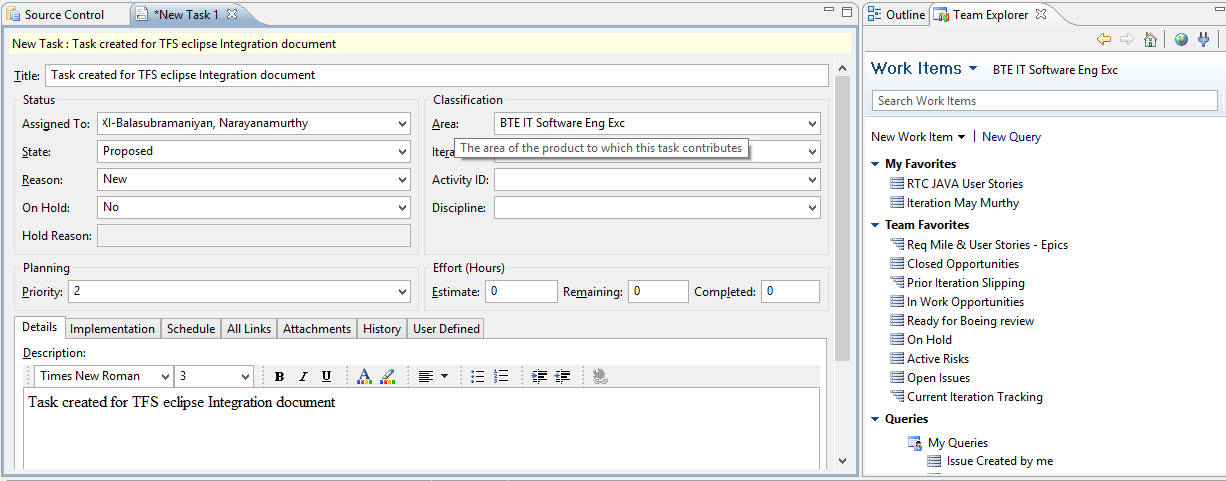
1. You can also associate the work item [Using Team Explorer](#_Using_Team_Explorer) during check-in.

## Work Items

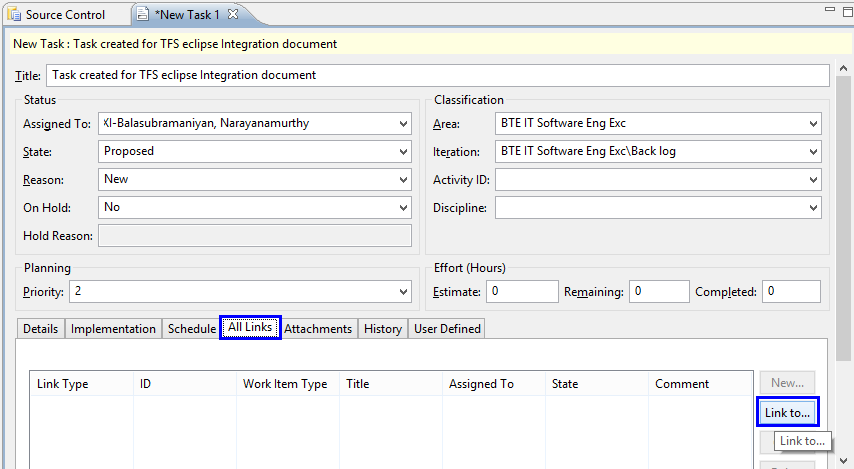
1. Go to Team Explorer 🡪 Work Items.
2. Work Items Window appears as shown below.

### Create a new Work Item

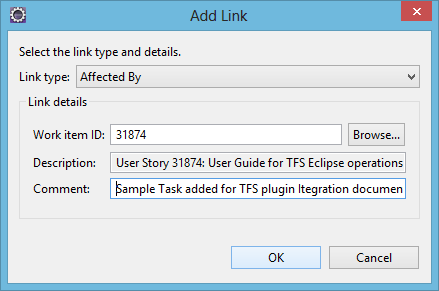
1. Go to Team Explorer 🡪 Work Items 🡪 New Work Item 🡪 <Any Work Item from the drop down as required>. Here we have chosen “Task” as the work item
2. A new window appears as shown below. Provide the Task name in the Title text box.

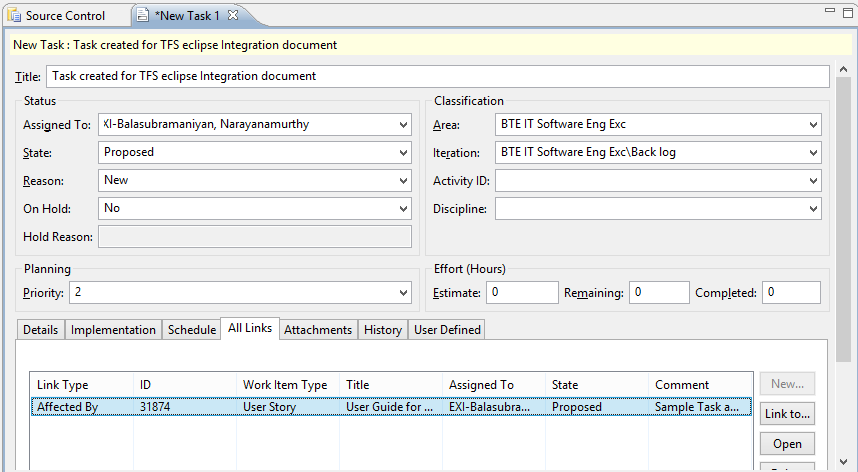


1. Click All Links 🡪 Link to…



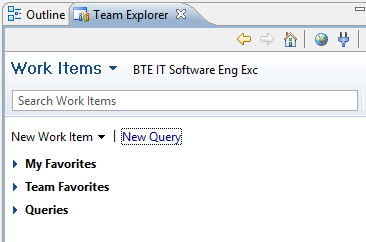
1. Add Link Window appears. Provide the Work Item ID to map the details of the same. Click OK.

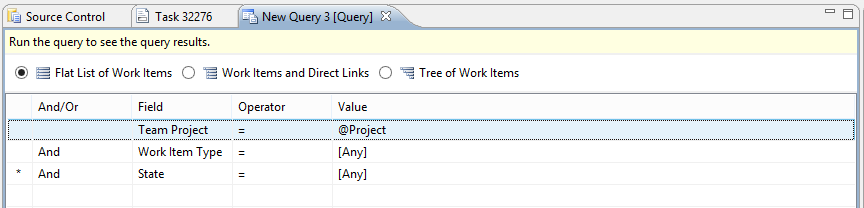


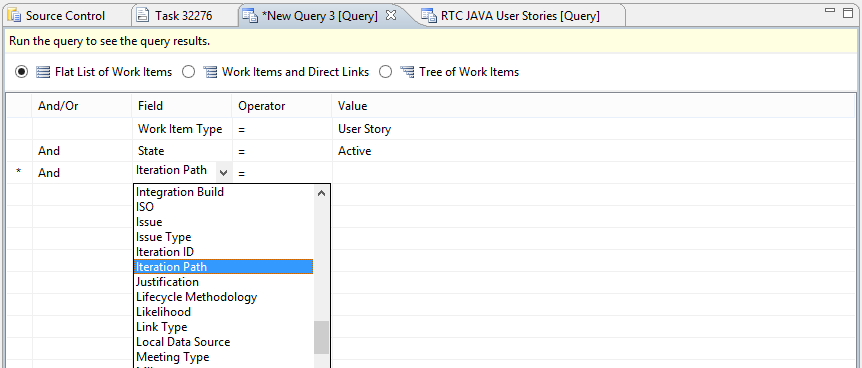
1. Work Item has been linked to the task as shown below.
2. Save the created work item by pressing the keys Ctrl+S.

## Queries

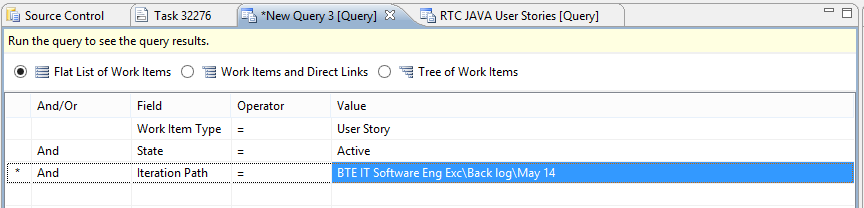
1. Go to Team Explorer 🡪Work Items 🡪 New Query



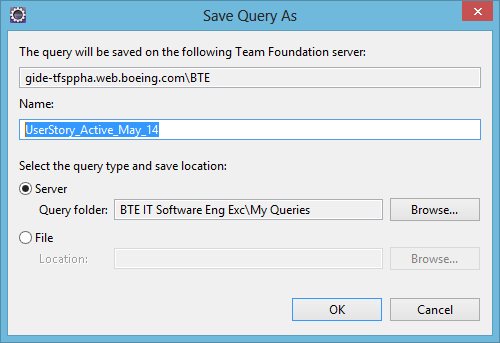
1. A new Query Window appears as shown below.
2. Choose the required Field, Operator and Value. Press Ctrl+S to save the query.

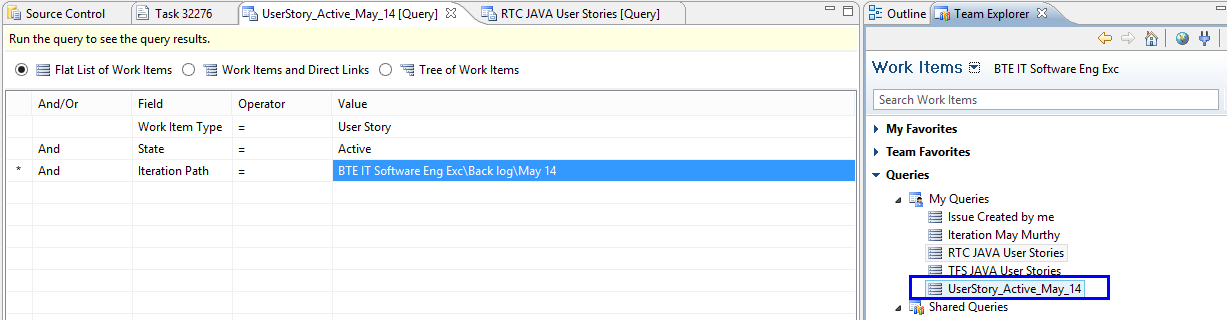


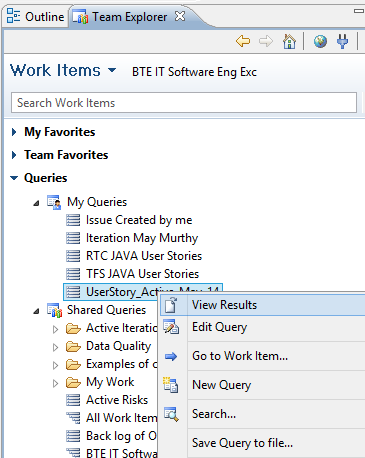
1. A Sample query as shown below.

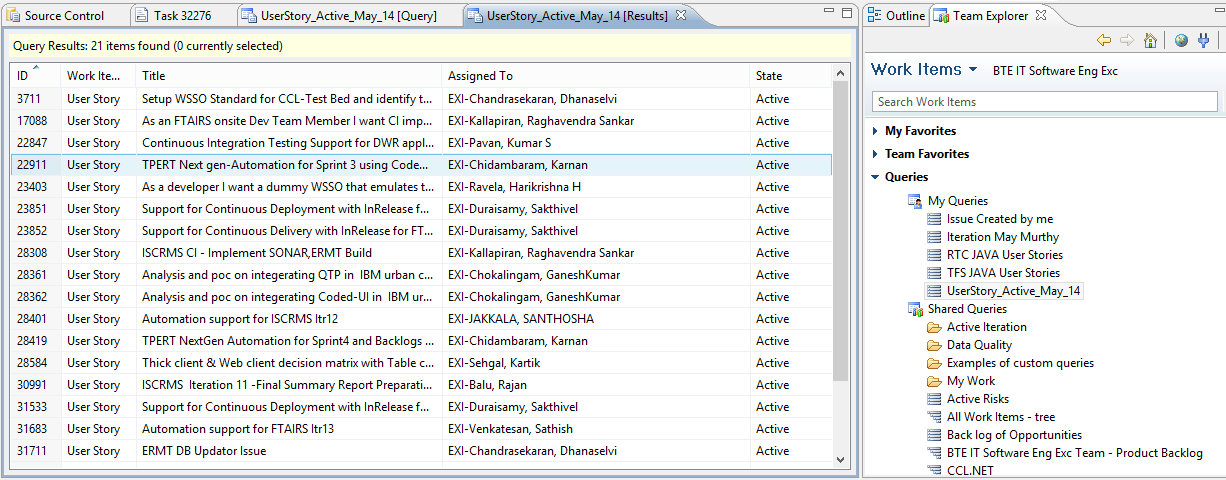


1. New Window “Save Query As” appears as shown below.
2. Provide the name for the created query and Click OK.



1. Go to Team Explorer 🡪 Work Items 🡪 Queries 🡪 My Queries. You could see the newly created query.
2. Go to Team Explorer 🡪 Work Items 🡪 Queries 🡪 My Queries 🡪 <Query Name> 🡪 View Results.



1. The result of the saved query as shown below.

# TFS in Linux

## Glossary

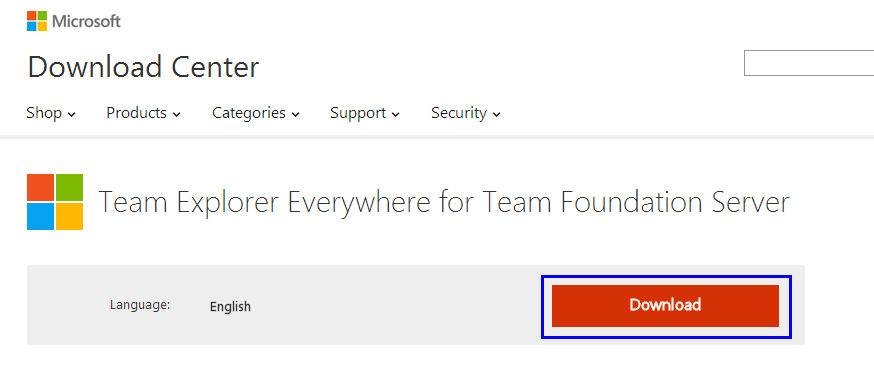
We use these terms below to explain with examples.

|  |  |  |
| --- | --- | --- |
| **Key\_term** | **Description** | **Example** |
| ***$TEE\_INSTALL\_DIR*** | Directory path where extracted TEE-CLC-11.0.0 folder is present | /home/murthy/sw/TEE-CLC-11.0.0 |
| ***$workspace\_name*** | Name of a workspace in TFS | CCT\_Workspace |
| ***$tfsserver*** | TFS Server URL | <http://10.105.5.109:8080/tfs/BoeingBU> |
| ***$loginuser*** | A Login username to access TFS | HCLTECH\\backup |
| ***$loginpasswd*** | Password for the above username to access TFS | \*\*\*\*\*\*\*\*\*\* |
| ***$mappath*** | A path in TFS where your work area path is present. | $/Boeing\_CCT |
| ***$workspacedir*** | A path in the local machine where your workspace is being mapped | /home/murthy/projects/src |
| ***$ReleaseTag*** | Name of the Valid Label name available in TFS | CCT\_Release\_Build\_01\_11Aug2014 |
| ***$shelveset\_name*** | Name of the shelveset | test\_shelveset |

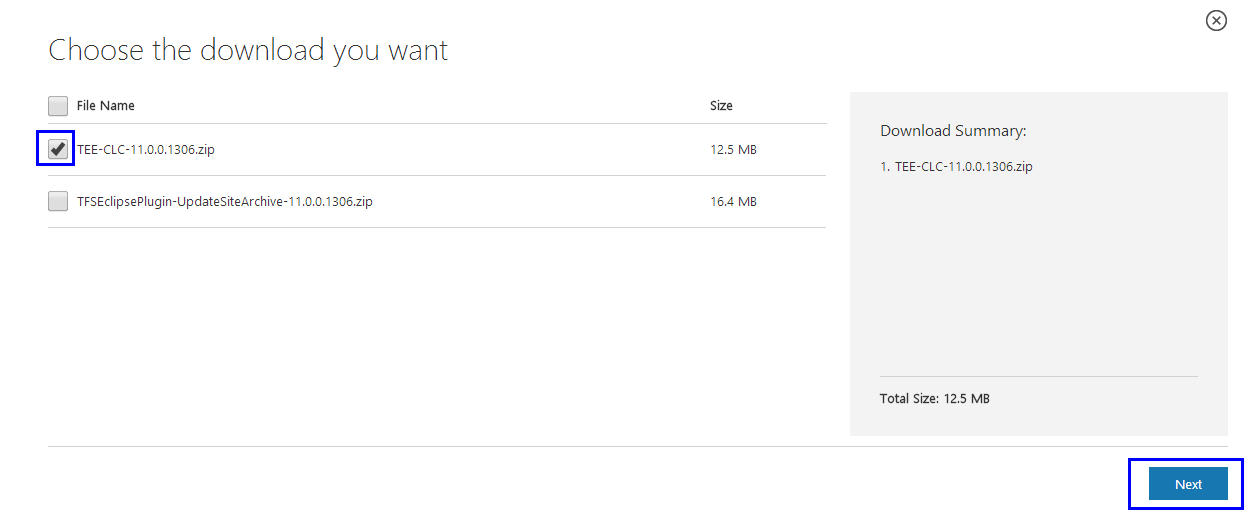
## Install TFS in Linux

1. Download TEE (Team Explorer Everywhere) for Team Foundation Server software from the below link.

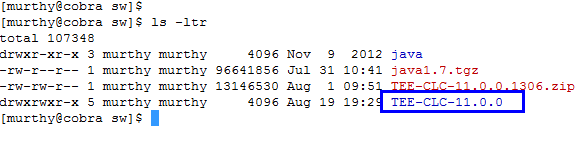
<http://www.microsoft.com/en-ca/download/details.aspx?id=30661>



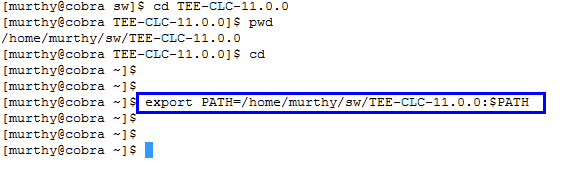
1. Choose the option **TEE-CLC-11.0.0.1306.zip** and click next.



1. Copy the downloaded software (**TEE-CLC-11.0.0.1306.zip**) to Linux box and extract the same.

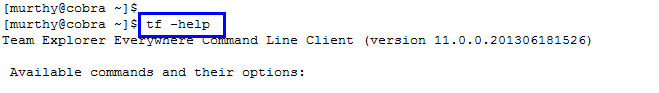


1. Add its executable to the PATH environmental variable.

***export PATH=/path/to/$TEE\_INSTALL\_DIR/:$PATH*** 

1. Check if Path has been properly set. You could see the output as shown below.

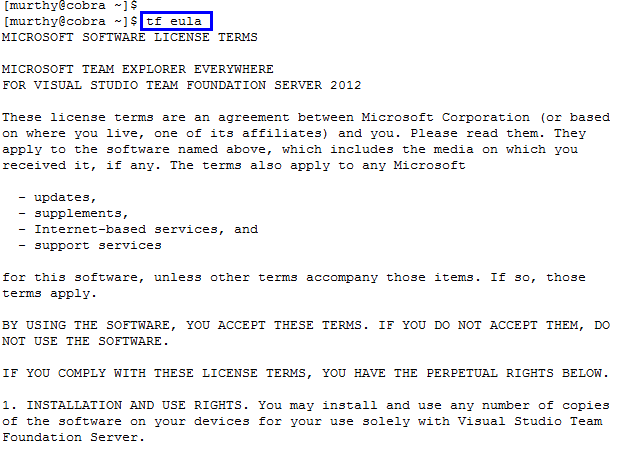
***tf -help***



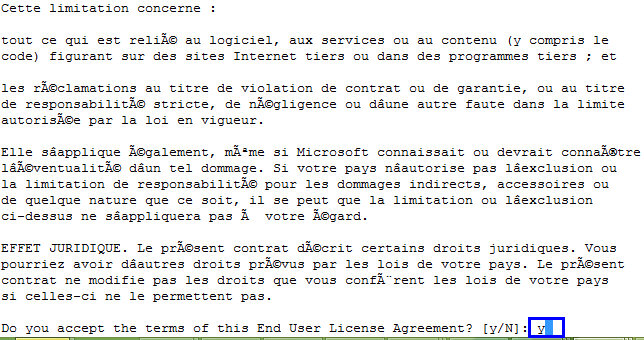
## Accept License Agreement

View and accept the Microsoft Software License Terms for the Cross-platform Command-Line Client. You must run the EULA command below before you can perform any version control operations.

1. ***tf eula***



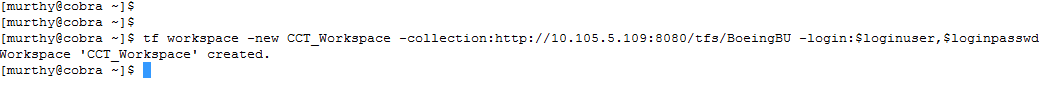
1. Press **y** to accept the terms and conditions from Microsoft.



## Create TFS Workspace

A workspace is a local copy of the files and folders on the server, plus any changes that you have made locally. When you add, edit, delete, move, rename, or otherwise change any version-controlled item, your changes are isolated in your workspace where you can make and test your changes. Your pending changes are committed to the server and become available to other users outside your workspace when you perform a check-in.

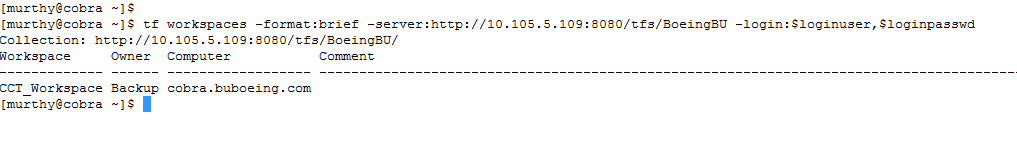
***tf workspace -new $workspace\_name -collection:$tfsserver -login:$loginuser,$loginpasswd***



## View list of available workspaces

The workspaces command of the tf command-line utility displays information about workspaces on the current computer, owned by a specified user, or for all workspaces associated with a specific Team Foundation Server. For each workspace, Team Foundation displays the name, owner, comment, and computer name. With detailed output, it also shows the workspace mappings. Unless you provide a filter such as owner, computer or workspace name, Team Foundation only displays information about the workspaces that you have created.

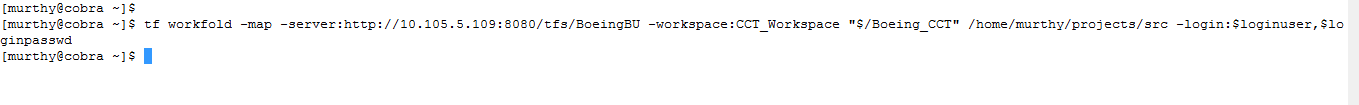
***tf workspaces -format:brief -server:$tfsserver***



## Map TFS Working folder

Run the workfold command to create workspace mappings between local folders and folders on the server. By default, workspace mappings are applied recursively. When you map a local folder to a server folder, the system implicitly creates a mapping between all its current and future subfolders.

***tf workfold -map -server:$tfsserver -workspace:$workspace\_name $mappath $workspacedir -login:$loginuser,$loginpasswd***



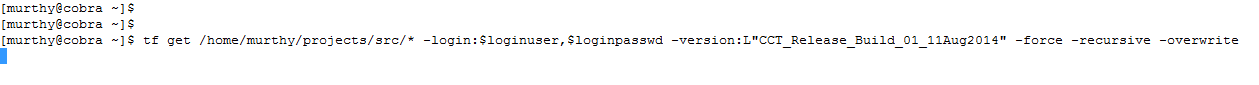
## Get

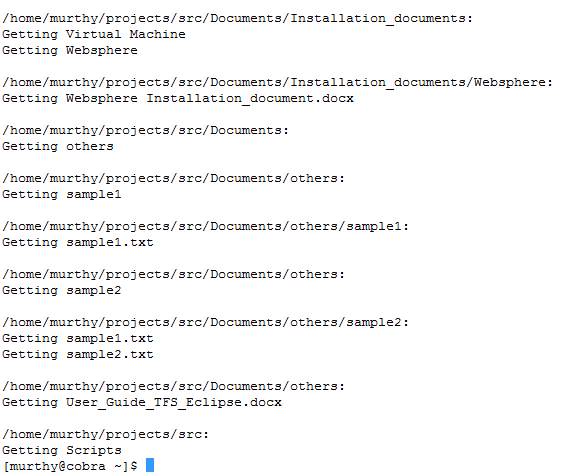
Run the get command to retrieve a copy of a file from Team Foundation Server or to synchronize your workspace with the most recently checked-in changes on the server.

The below image shows getting the source code from a valid label name

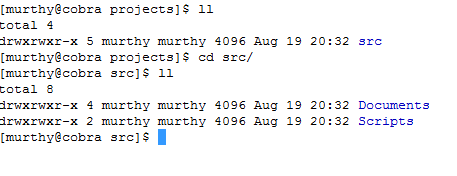
(***-version:L$ReleaseTag***).

***tf get "$workspacedir"/\* -login:$loginuser,$loginpasswd -version:L$ReleaseTag -force -recursive –overwrite***





Source code location in Local machine:

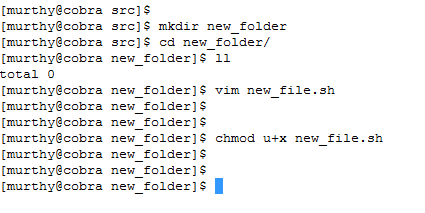


You can use “-***version:T***” to **get latest** source code from TFS.

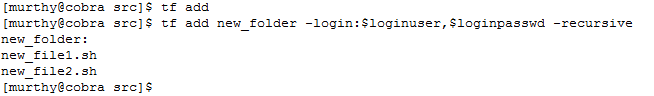
## Add Items to folder

Run the add command to add the new file to Team Foundation Server.

Add new folder from the checked out location and create a file and provide permissions.



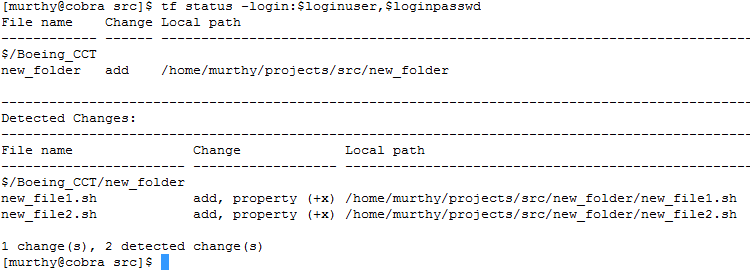
***tf add new\_folder -login:$loginuser,$loginpasswd -recursive***



## Status

Run the status command to review information about pending changes to items in the workspace. You can check the status of TFS operation in the middle of any stage.

***tf status -login:$loginuser,$loginpasswd***

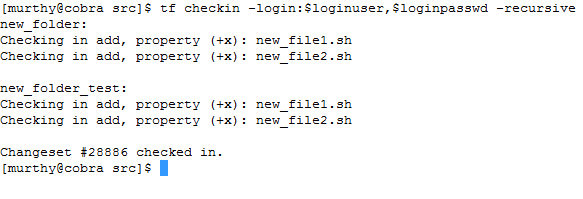


## Checkin

Run the check-in command to check in pending changes from current workspace to Team Foundation Server.

The following example submits the pending changes associated with work item 11 in the current workspace to the Team Foundation version control server together with the specified comment to create a new changeset.

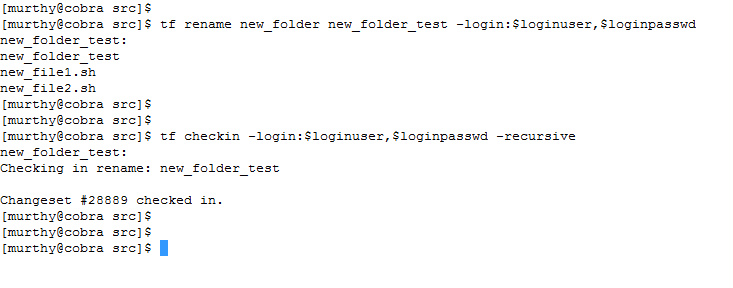
***tf checkin -comment:"TFS in Linux checkin comment test" -login:$loginuser,$loginpasswd -recursive***



## Rename

You can use the **rename** command of the **tf** command-line utility to move or rename a version-controlled item in your workspace. Use the **rename** command to move multiple files by specifying wildcard characters. The ability to rename multiple version-controlled files or folders, is only available from the command-line. Check-in after performing rename operation.

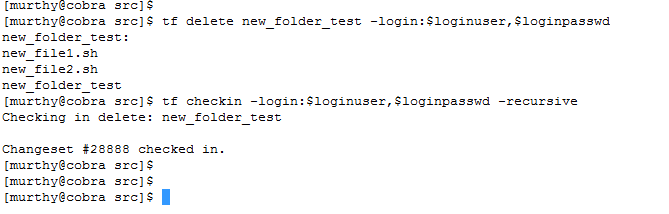
***tf rename $old\_folder $new\_folder -login:$loginuser,$loginpasswd***



## Delete

Run the delete command to remove a file or folder from the Team Foundation Server and delete them from the disk. The results of this command are not visible in other workspaces until you perform a check-in operation. Check-in after deletion.

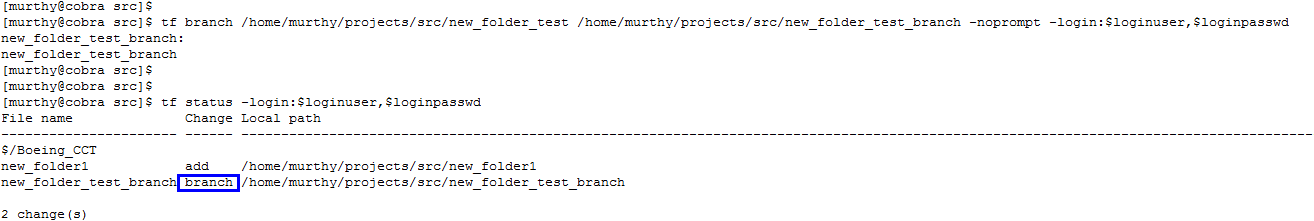
***tf delete ${folder/file} -login:$loginuser,$loginpasswd***



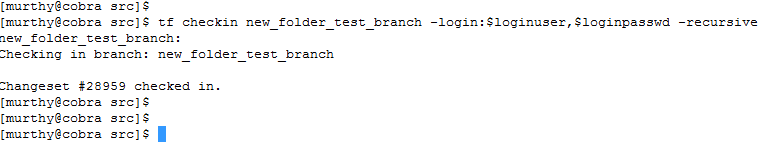
## Branch

The **branch** command copies an item or set of items, including metadata and version control history, from one location to another in the Team Foundation version control server and in the local workspace.

***tf branch $old\_branch\_name $new\_branch\_name -noprompt -login:$loginuser,$loginpasswd***



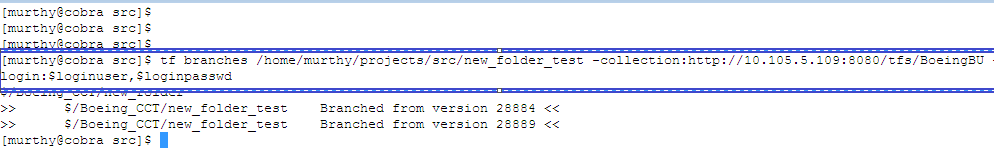
Check-in after applying a branch.



## View Branches

The branches command displays when an item has been the source or destination of a branch operation. The output displays the parent branch for each version.

***tf branches $workspacedir -collection:$tfsserver -login:username,password***



## Label

A label is a marker that you can attach to a set of unrelated files and folders in the Team Foundation version control server. Use the label to simplify their retrieval to a workspace for either development or build purposes. Label command is used to attach a label from a version of a file or folder in the server for Team Foundation version control

***tf label $ReleaseTag itemspec -comment:"$comment" -recursive -login:$loginuser,$loginpasswd -collection:$tfsserver***





## View Labels

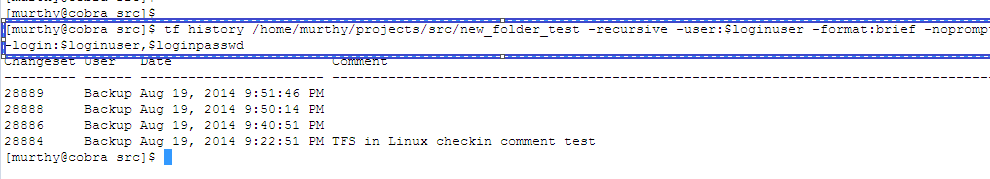
The **labels** command displays information about labels in the Team Foundation version control server. By default, the results display the label name, owner and creation date for each label.

***tf labels -format:brief -collection:$tfsserver -login:$loginuser,$loginpasswd***

## History

**History** command is used to display revision history for one or more files and folders. Team Foundation displays a revision entry in the changeset output. Changeset details include information about the user who committed the change to the Team Foundation version control server, the files and folders for which pending changes were committed, the type of pending change that was committed, the changeset number, and the date and time when the changeset was created.

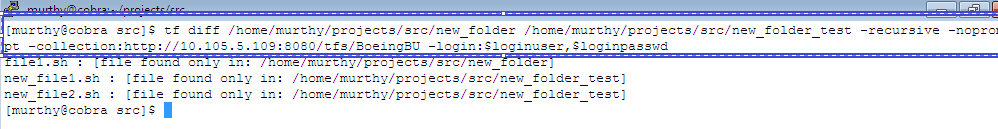
***tf history $workspace\_dir -recursive -user:$loginuser -format:brief -noprompt -login:$loginuser,$loginpasswd***



## Diff

Diff command is used to display and compare a visual representation of the differences between files in two server folders, in a server folder and a local folder, or in two local folders.

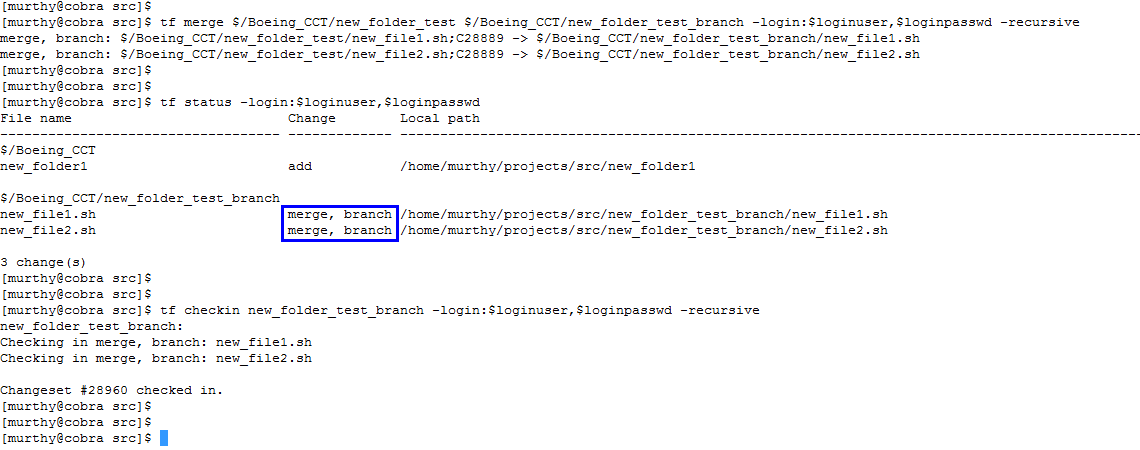
***tf diff $sourcePath $targetPath -recursive -noprompt -collection: $tfsserver -login:$loginuser,$loginpasswd***



## Merge

You can use the **merge** command of the **tf** command-line utility to apply changes in an existing source branch to an existing target branch. You can merge an individual revision or a complete changeset to the target branch. You can merge changes from the source to the target branch or from the destination to the source branch.

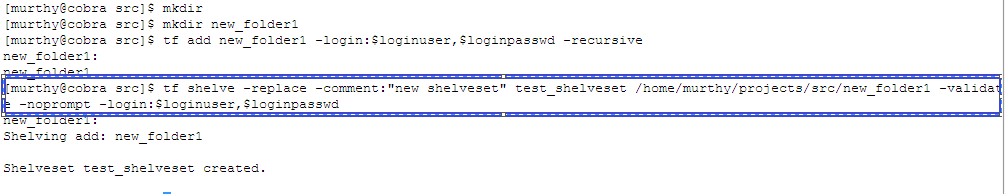
The **merge** command also lets you query for changes in a source branch which have not been migrated to the target branch. Additionally, it lets you indicate that certain changes will never be merged from the source to the destination and should no longer be displayed as candidates for a merge operation.



## Shelve

The **shelve** command of the **tf** command-line utility backs up pending changes, a list of associated work items, in-progress check-in notes, and comments in a shelveset on the Team Foundation Server. A shelveset is much like a changeset that is not committed to the server. Like a changeset, a shelveset can be retrieved from the server into a local workspace by any user who has sufficient permissions.

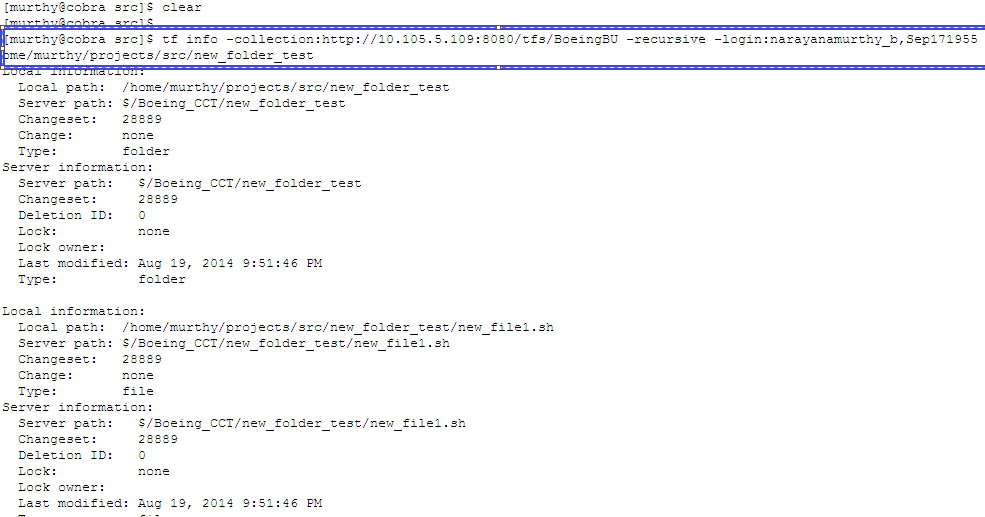
***tf shelve -replace -comment:"$comment" $shelveset\_name -validate -noprompt -login:$loginuser,$loginpasswd***



## Properties

The properties command displays several pieces of information about a version-controlled item. Local and server information is listed separately.

***tf properties -collection:$tfsserver -recursive -login:$loginuser,$loginpasswd $workspacedir***



## Undo

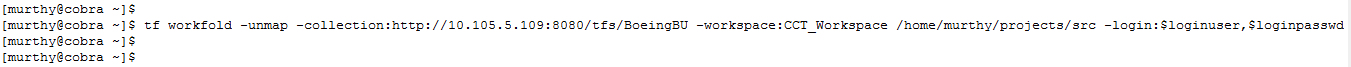
The **undo** command checks to see whether files that have pending edits being undone have been modified on disk. If modifications are detected, the undo command without /noprompt displays a prompt asking whether to continue with undoing the changes. You may respond with Yes, No, or All.

***tf undo -workspace:$workspace\_name;$workspaceowner\_name -recursive $workspacedir -noprompt -login:$loginuser,$loginpasswd -collection:$tfsserver***



## Unmap TFS Working folder

***tf workfold -unmap -collection:$tfsserver -workspace:$workspace\_name $workspacedir -login:$loginuser,$loginpasswd***



# Reference

1. <http://marketplace.eclipse.org/content/tfs-plug-eclipse>
2. <http://msdn.microsoft.com/en-us/library/hh301122.aspx>
3. <http://msdn.microsoft.com/en-us/library/cc31bk2e%28v=vs.100%29.aspx>