Portrait
Foundation

Configuration Suite - TFS Source Control Setup and User Guide

Edition 4.0

01 July 2014

Pitney Bowes
Software

Portrait Foundation
Configuration Suite - TFS Source Control Setup and User Guide

Acknowledgement of trademarks

Other product names, company names, marks, logos and symbols referenced herein may be the trademarks or registered trademarks of their registered owners.

About Portrait Software

Portrait Software is now part of Pitney Bowes Software Inc.

Portrait Software enables organizations to engage with each of their customers as individuals, resulting in improved customer profitability, increased retention, reduced risk, and outstanding customer experiences. This is achieved through a suite of innovative, insight-driven applications which empower organizations to create enduring one-to-one relationships with their customers.

Portrait Software was acquired in July 2010 by Pitney Bowes to build on the broad range of capabilities at Pitney Bowes Software for helping organizations acquire, serve and grow their customer relationships more effectively. The Portrait Customer Interaction Suite combines world leading customer analytics, powerful inbound and outbound campaign management, and best-in-class business process integration to deliver real-time customer interactions that communicate precisely the right message through the right channel, at the right time.

Our 300 + customers include industry-leading organizations in customer-intensive sectors. They include 3, AAA, Bank of Tokyo Mitsubishi, Dell, Fiserv Bank Solutions, Lloyds Banking Group, Merrill Lynch, Nationwide Building Society, RACQ, RAC WA, Telenor, Tesco Bank, T-Mobile, Tryg and US Bank.

Pitney Bowes Software Inc. is a division of Pitney Bowes Inc. (NYSE: PBI).

For more information please visit: http://www.pitneybowes.co.uk/software/

UK

Portrait Software
The Smith Centre
The Fairmile
Henley-on-Thames
Oxfordshire, RG9 6AB, UK

Email: support@portraitsoftware.com
Tel: +44 (0)1491 416778
Fax: +44 (0)1491 416601

America

Portrait Software
125 Summer Street
16th Floor
Boston, MA 02110
USA

Email: support@portraitsoftware.com
Tel: +1 617 457 5200
Fax: +1 617 457 5299

Norway

Portrait Software
Portrait Million Handshakes AS
Maridalsveien. 87
0461 Oslo
Norway

Email: support@portraitsoftware.com
Tel: +47 22 38 91 00
Fax: +47 23 40 94 99

About this document

Purpose of document

This document describes the Microsoft Team Foundation Server (TFS) features that are supported within the Portrait Foundation Configuration Suite when TFS is used as the source control provider.

Intended audience

Anyone interested in storing the Portrait Foundation configuration repository in TFS and/or wanting to understand the TFS features that are available within the Configuration suite.

Software release

Portrait Foundation 5.0 or later.

Contents

1 TFS Features Supported

The Configuration Suite supports TFS features when Portrait Foundation configuration objects are checked-in or published from the Configuration Suite version control actions. These are:

- Check in / Check in all
- Publish / Publish all

These features are not supported for a number of other Configuration Suite operations that manipulate files in the Portrait Foundation repository. The operations that do not use these features are:

- Create new package
- Add / Remove package
- Create/Remove/Update Work folder
- Delete object
- Move to package
- All Repository Manager operations

The following sections describe the features supported during the check in and publish operations.

1.1 Changesets

All configuration objects that are checked in at the same time will be associated with the same TFS changeset. Similarly, when multiple objects are published at the same time the related files will be associated with the same changeset.

The check in and publish steps are always on to different changes sets. So, if a configuration object is published, but first requires to be added to source control or checked in, then two changesets will be created: one to do the check in and one to do the publish step.

1.2 Work items

Work items are tasks that are assigned to members of a team to carry out a piece of work. During the check-in and publish processes within the Configuration Suite the user can select work items to associate with the changeset.

TFS also allows work items to be 'resolved' when a check in is performed. Currently this feature is not supported in the Configuration Suite. Work items can only be "Associated" with the changeset.

1.3 Check-in notes

Check-in notes are configurable values that can be linked to a changeset. Check-in notes are used for capturing information regarding things like the name of the code reviewer, for example. Check-in notes are configurable and can be extended by the TFS administrator. If a check-in note is set to by mandatory the check-in will not be allowed to proceed until a value has been captured.

Portrait Foundation allows check-in notes to be captured and will force mandatory notes to be entered.

1.4 Check-in Policies

Check-in Policies are configured on the TFS server for each project and include rules such as 'Must select a work item'. Again, the set of available policies can be extended by the TFS administrator. If a policy is not valid then the user can override the policy and provide a reason. This reason is kept with the details of the changeset.

Portrait Foundation supports Check-in policy validation and supports the policy override option.

2 Setting up Configuration Suite to use TFS

The following steps need to be followed to set up the Configuration Suite to use Team Foundation Server (TFS) as the source control provider.

2.1 Prerequisites

Microsoft Visual Studio or **Team Explorer** is installed and there is a relevant Workspace mapping for the Portrait Foundation Configuration Repository. The Configuration Suite TFS integration will only work if the Workspace location is set to **Server**. This setting can be accessed thought the "Advanced >>" button when editing the Workspace properties via Visual Studio or Team Explorer.

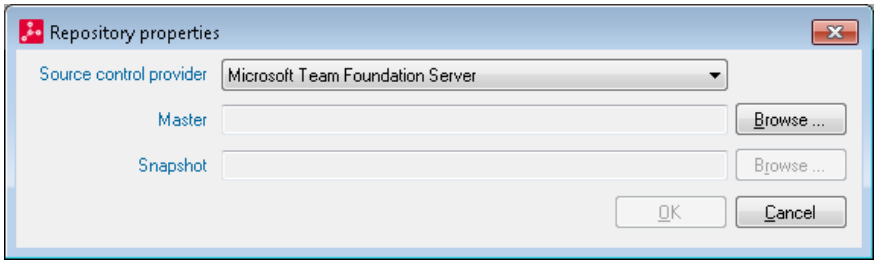
The appropriate **32-bit** version of **Microsoft Team Foundation Server MSSCCI Provider** must also be installed. Any older versions should be uninstalled before installing the latest version. Please refer to the release notes for details of supported versions.

NB: If Visual Studio is installed, then the Team Foundation Server MSSCCI Provider is not required.

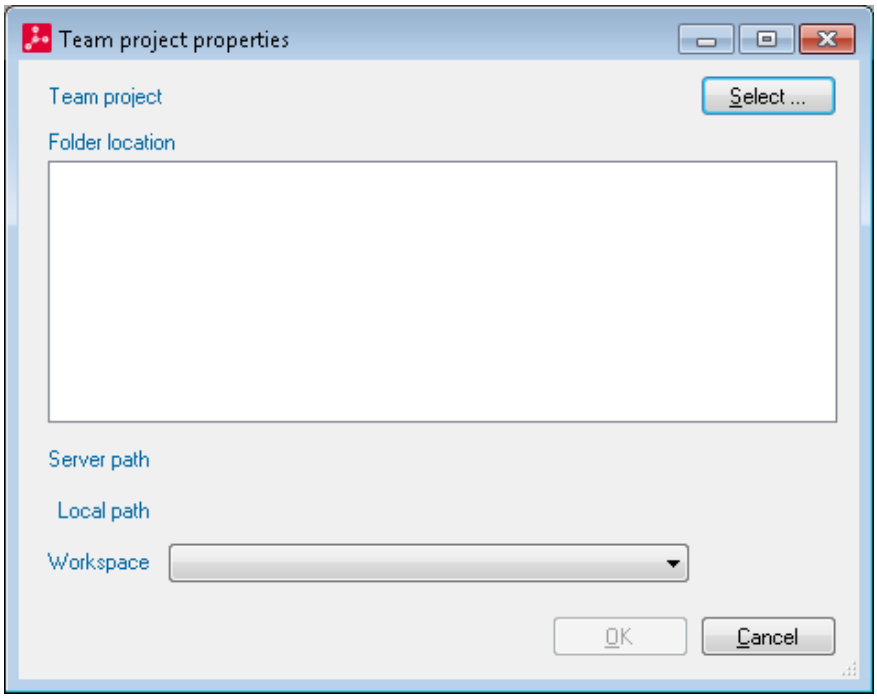
Configuration Suite users have been granted permission to access the Team Foundation Server.

2.2 Setup steps

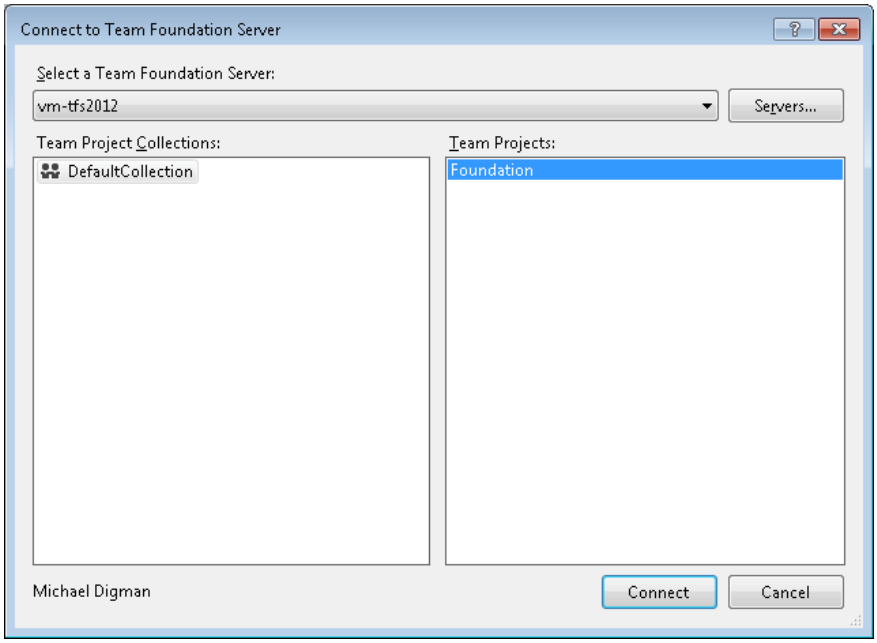
1. Launch the Configuration Suite.
2. Click on the "Settings..." button. The Repository properties dialog will be shown:



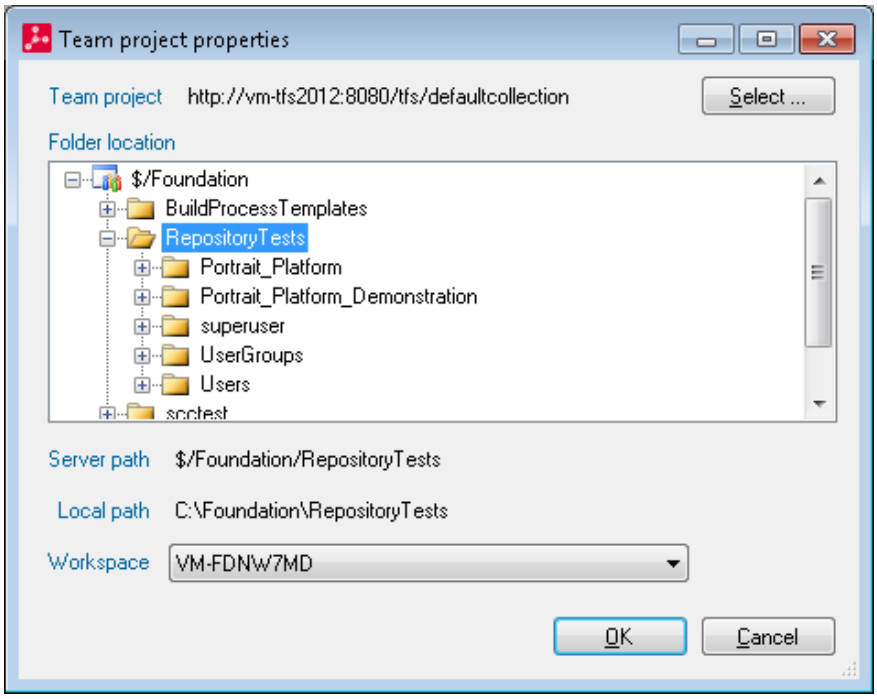
3. Select "Microsoft Team Foundation Server" from the source control provider drop-down list. If it doesn't appear in the list you need to make sure that it is installed – see above.
4. Click on the Browse button next to the "Master" text box. The following dialog will be shown:



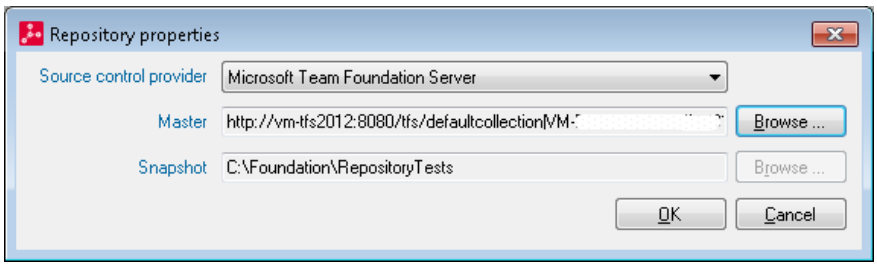
5. Click "Select..." to find the Team Foundation Server you wish to use. The following dialog will be shown:



6. Click "Servers..." to browse for other TFS servers available. Once the relevant Team project is selected, click "Connect". The Team project properties dialog will be shown with a populated list of Folders from TFS:



7. Select the folder that contains the Repository root (folder with .policy file in it). The "OK" button will only be enabled if there is a "Local path" setup. This is done by configuring a Workspace mapping for the Folder using Visual Studio or Team Explorer. Click "OK".



8. You will be returned to the Repository properties screen which will now show the selected server and local folders. Click on "OK" and logon to the Configuration Suite in the usual way.

3 Using TFS in the Configuration Suite

3.1 Check-in and Publish

The TFS details are captured as part of the Configuration Suite Check-in and Publish operations.

The TFS dialog (shown below) is presented after the standard Check-in or Publish dialogs prior to the progress dialog starting.

3.1.1 TFS Details Dialog

The dialog has four pages for:

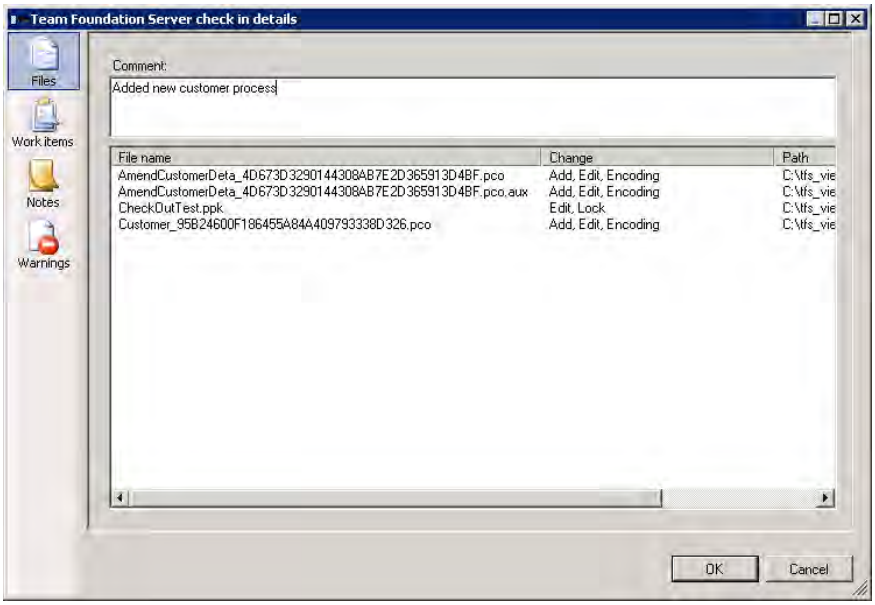
- Displaying the files to be checked-in
- Work item selection
- Capturing check-in notes
- Displaying policy warnings

To cancel the check-in process click on the Cancel button.

3.1.1.1 Files

The 'Files' page shows the files related to the objects being checked-in (or published) and the check-in comment that was captured on the check-in screen.

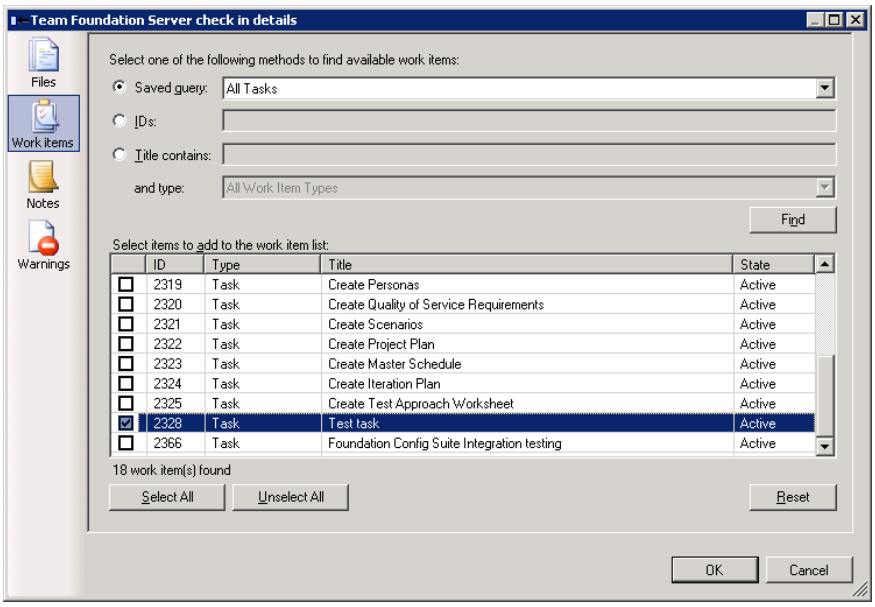
The check-in comment can be modified from this page if necessary.



3.1.1.2 Work Items

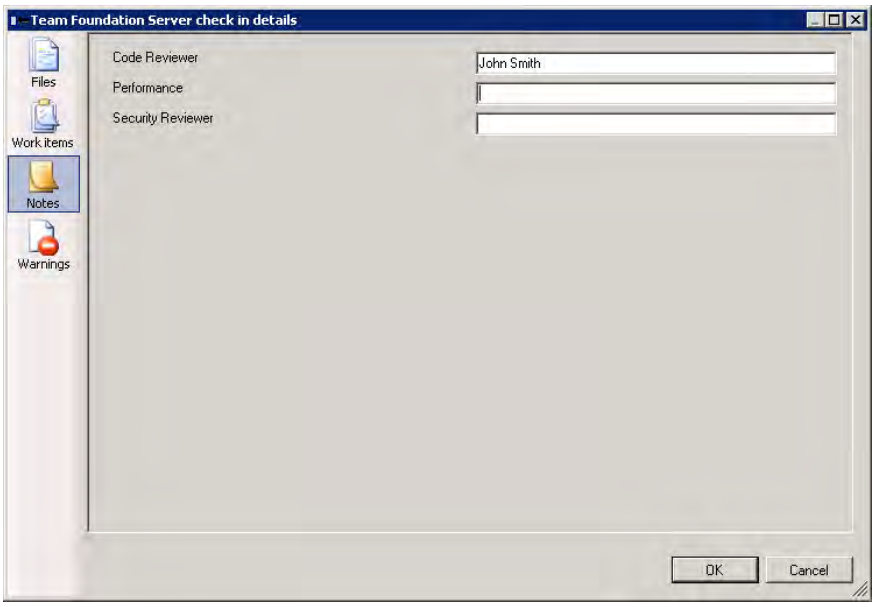
The Work Items page allows work items to be associated with the changeset.

Work items can be searched by selecting a query, by entering the work item ID or by searching for values in the work item title.



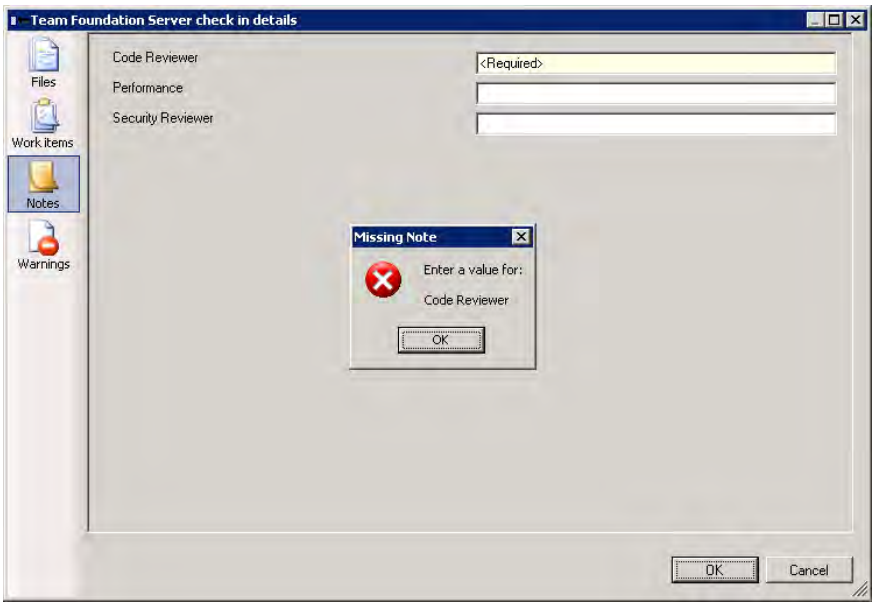
3.1.1.3 Notes

The 'Notes' Page allows configured check-in notes to be captured.



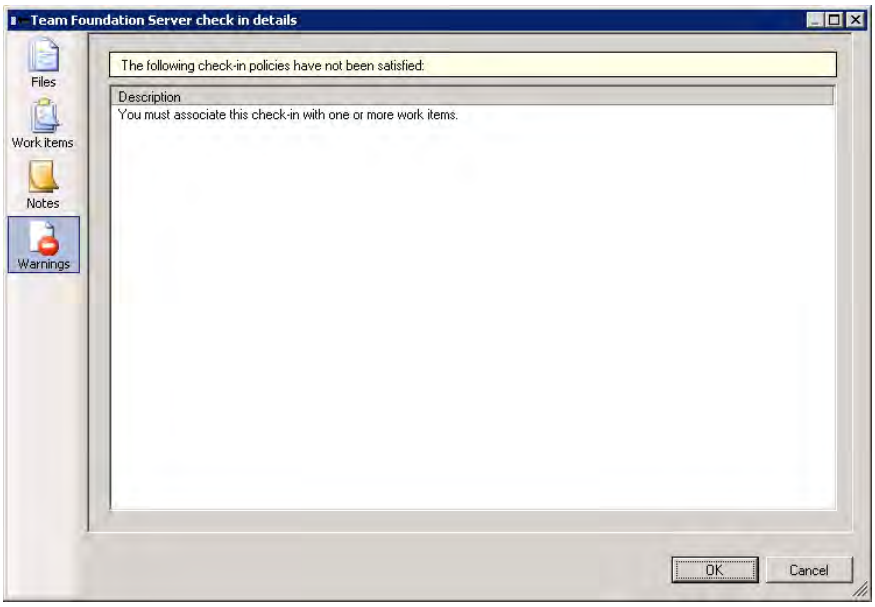
If a note is mandatory it will show highlighted with the text **<Required>** in it.

All mandatory notes must be entered. If a mandatory note is not entered a message box will be displayed stating which note must be entered.



3.1.1.4 Policy warnings

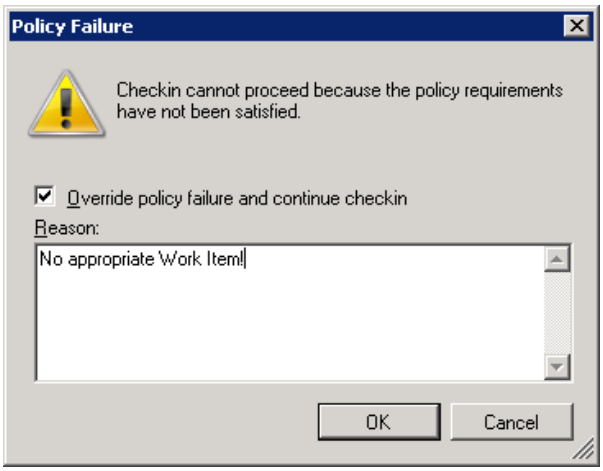
The 'Warnings' page shows information about any policies that have not been met.



If the 'OK' button is selected and there are TFS check-in policies that have not been satisfied, a dialog is shown warning that policies have not been met.

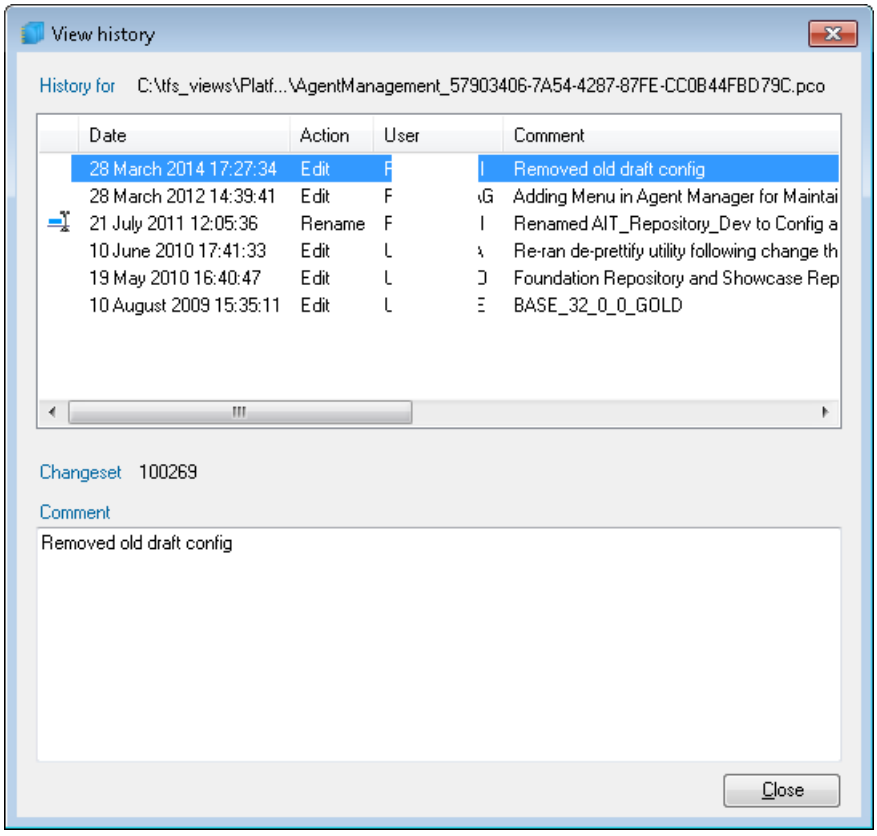
If the policy cannot be satisfied there is the option to override the policy and provide a reason.

If a policy is overridden the reason text is associated with the changeset.



3.2 View history

When using TFS as the source control provider the View history operation (from the editor window "File" menu) shows the TFS history for the related PCO file.



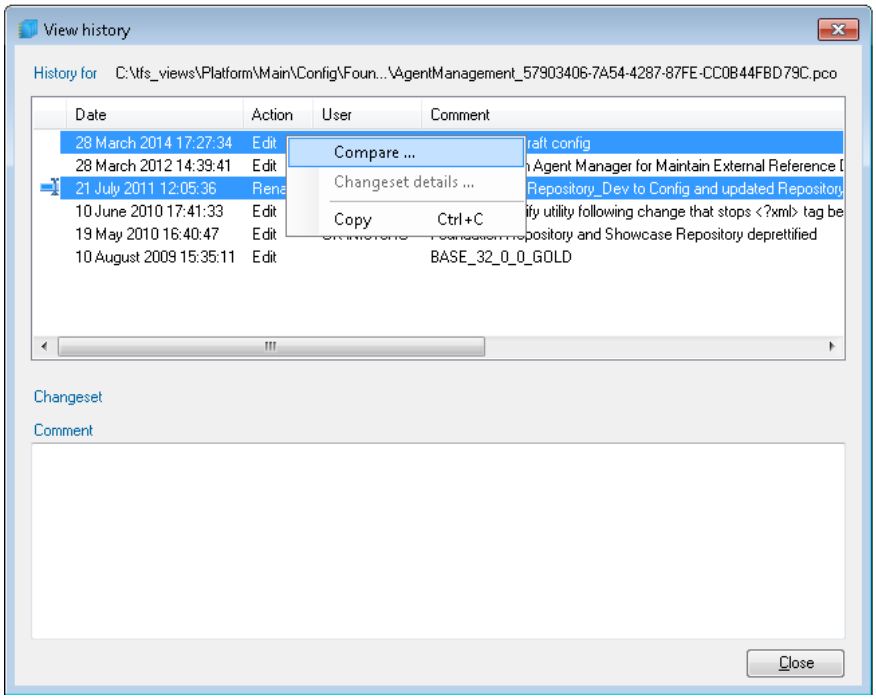
There is a context menu available for each row in the list.

The '**Changeset Details**' option launches the TFS dialog which shows the related work items, files in the changeset, check-in comments, check-in notes, etc.

The '**Compare**' option shows the differences between the 2 selected versions of the PCO file in TFS. If the Configuration Comparison tool has been configured correctly, then clicking this button will open a pre-configured instance of the Configuration Comparison tool. Otherwise, this button will just compare the raw PCO xml files. Comparing from TFS using the Configuration Comparison tool is explained more detail in section 3.3.

3.3 Compare versions

If the Configuration Comparison tool has been installed with the option to integrate with TFS, comparing versions PCO files will launch a pre-configured instance of the Configuration Comparison tool. If you choose to compare just one entry in the history list you will compare that server version against the current local version, if you choose to compare two entries in the history list you will compare those two server versions.



If the Configuration Comparison tool has been installed but not integrated with Visual Studio during install it is possible to do this manually.

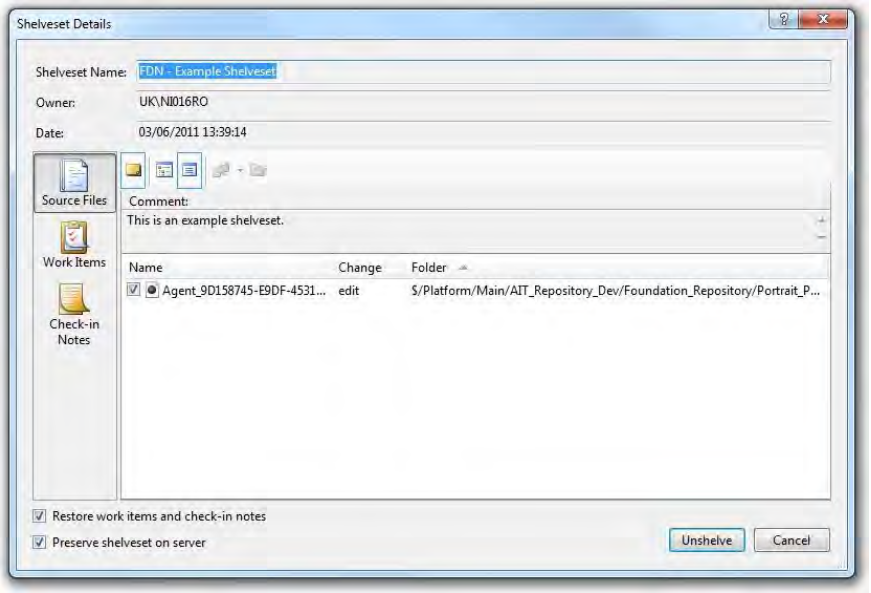
1. Open Visual Studio.
2. Go to '**Tools'** -> **'Options'**.
3. Select **'Source Control'** -> **'Visual Studio Team Foundation Server'**.
4. Click on the button labelled **'Configure User Tools...'**.
5. Click the '**Add'** button.
6. In Extension type **'.pco'**.
7. Leave operation as **'Compare'**.
8. Enter the path of the Configuration Comparison executable in Command.
   (e.g. **'C:\Program Files\PST\Portrait Foundation\common\bin\PortraitSoftware.Foundation.Tools.ConfigComparison.UserInterface.exe'**)
9. In Arguments enter **'%1 %2 %6 %7'**.
10. Click OK.

If the integration between TFS and the Configuration Comparison tool was not installed or manually configured then comparing PCO files will use the built-in TFS text comparer to compare raw XML content.

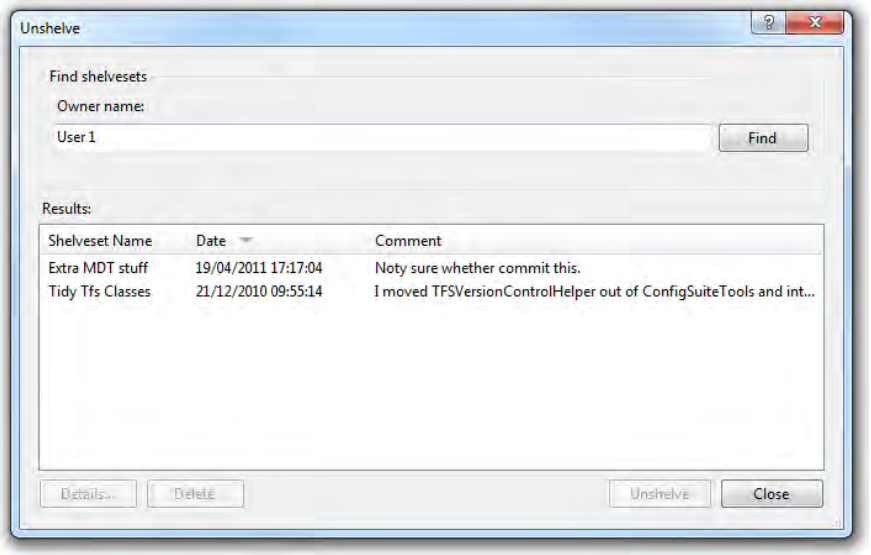
3.4 Shelve changes

When you have checked out items, you can choose to shelve any changes to those items. Selecting "Shelve" will copy the changes to a shelveset, but keep the existing changes checked out. This can be useful during the development process to facilitate code reviews.

All shelvesets must be given a name and a comment. Shelvesets created from within the Configuration Suite have their names appended with **'FDN – '** as seen in the screenshot below, this is to aid in identifying configuration shelvesets.



Shelvesets are maintained separately to the main source tree. They can be found in Visual Studio by opening the '**Pending Changes**' window, selecting the '**Source Files**' tab and finally the '**Unshelve**' button. Shelvesets are stored against the name of the user who shelved the changes, you can search for other users' shelvesets by filling in their user name in the top text box and clicking '**Find'**.



Items in shelvesets can be compared against the relative items in the source tree by right clicking an item and selecting **'Compare'** and then a compare option.