QLIKVIEW – TFS INTEGRATION

TABLE OF CONTENTS

INTRODUCTION:

Starting in version 11, QlikView has the capability to integrate with source control systems. In the initial release a Microsoft TFS connector was introduced and a further connector for Subversion was added in SR1. The feature is available when using QlikView Desktop.

A menu option in the **File** menu allows QlikView developers to connect a QlikView document to a source control system. This feature works with the XML project files that QlikView Desktop can generate and offers basic integration into third party source control systems.

QlikView 11 Source Control integration now ships with native support for Microsoft Team Foundation Server and Subversion for source control.

INSTALLATION OF MICROSOFT TFS MSSCCI PROVIDER:

To integrate TFS with Qlikview, we need the **Microsoft TFS MSSCCI Provider** software installed on the machine:

Below are the links where we can get the software version-wise:

➔ 2013 MSSCCI

https://marketplace.visualstudio.com/items?itemName=TFSPowerToolsTeam.MicrosoftVisualStudioTeamFoundationServer20132015M

➔ 2012 MSSCI

https://marketplace.visualstudio.com/items?itemName=TFSPowerToolsTeam.MicrosoftVisualStudioTeamFoundationServer2012MSSCC

➔ 2010 MSSCCI

https://marketplace.visualstudio.com/items?itemName=TFSPowerToolsTeam.MicrosoftVisualStudioTeamFoundationServer2010MSSCC

NOTE: Only 32-bit version of MSSCCI is supported by Qlikview.

Installation folder is "**C:\Program Files (x86)\Microsoft Team Foundation Server MSSCCI Provider**" as specified below. Once installed, we must ensure that Team Explorer is configured to a TFS Server before continuing.
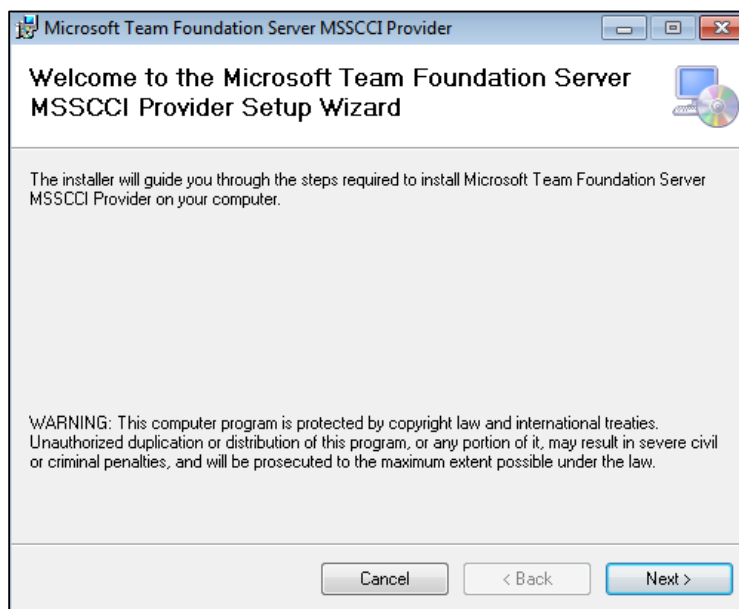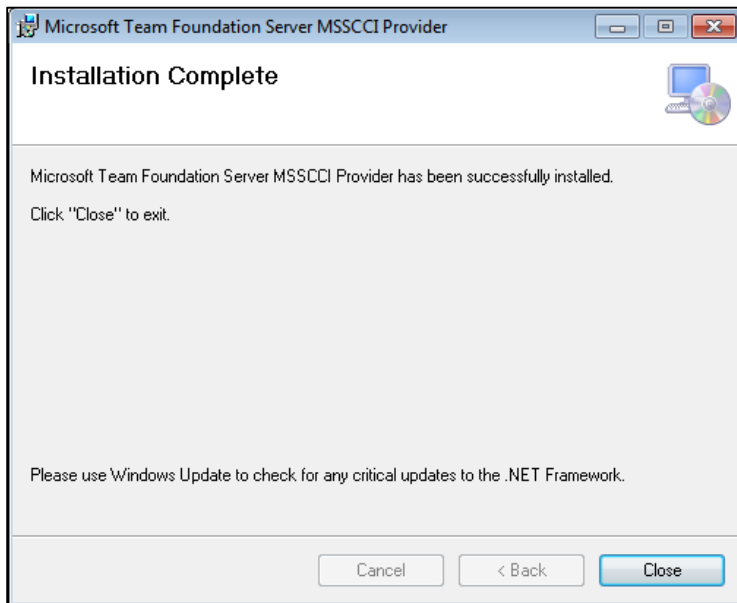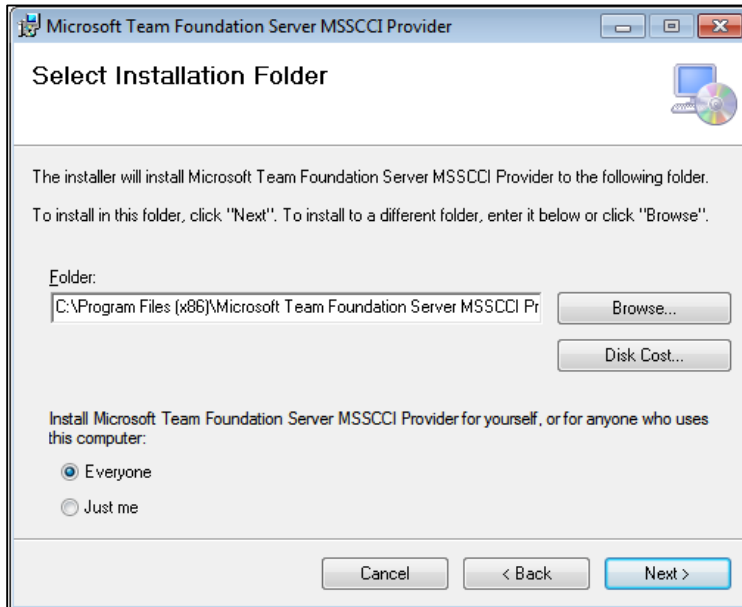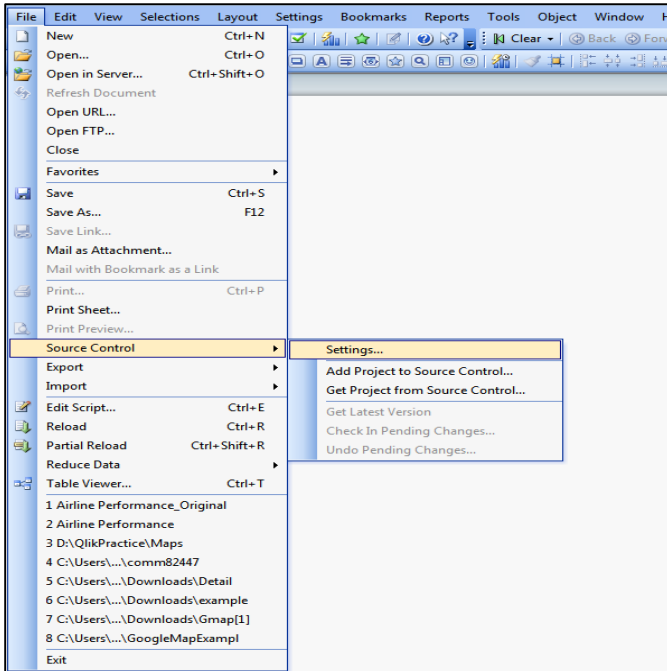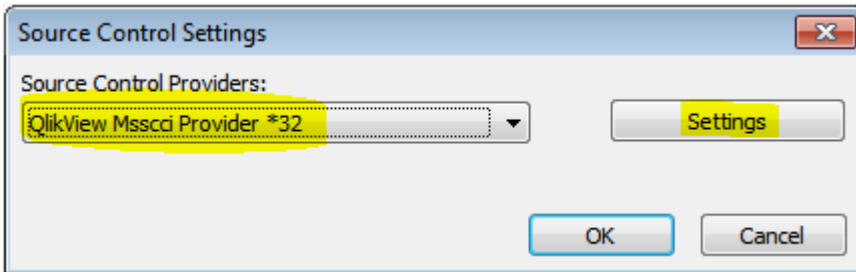
After the installation is complete you will find **TfsMsscciProvider.dll** file in the location "C:\Program Files (x86)\Microsoft Team Foundation Server MSSCCI Provider".
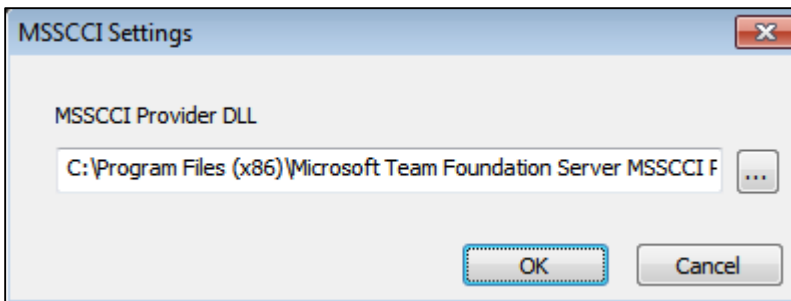
Open QlikView and go to this menu option **File→Source Control→Settings**..

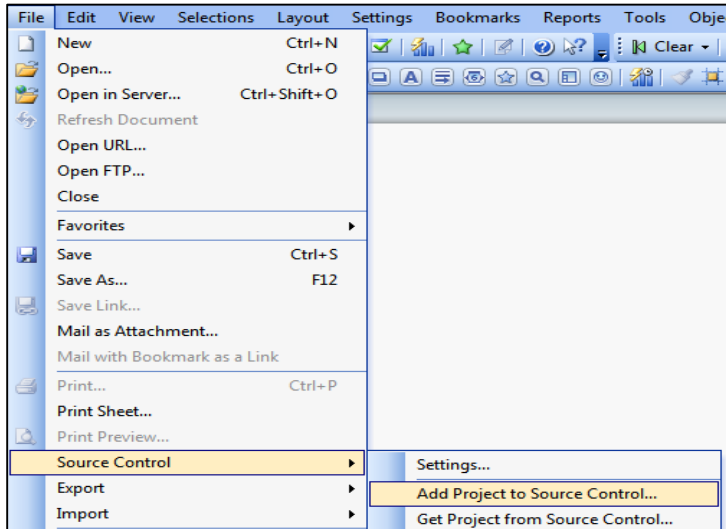Select "**Qlikview Msscci Provider *32**" option in the drop down and then click **Settings** button:

Provide the **.dll** path in the text box and click ok:
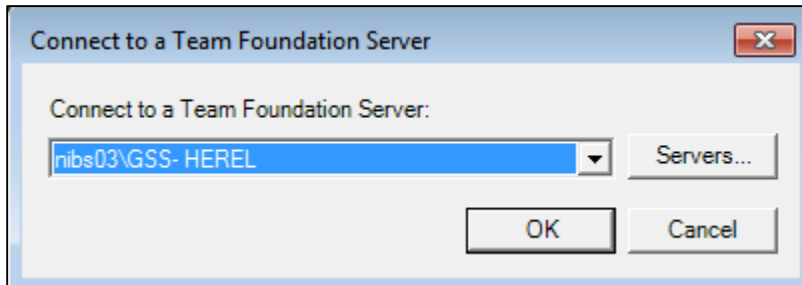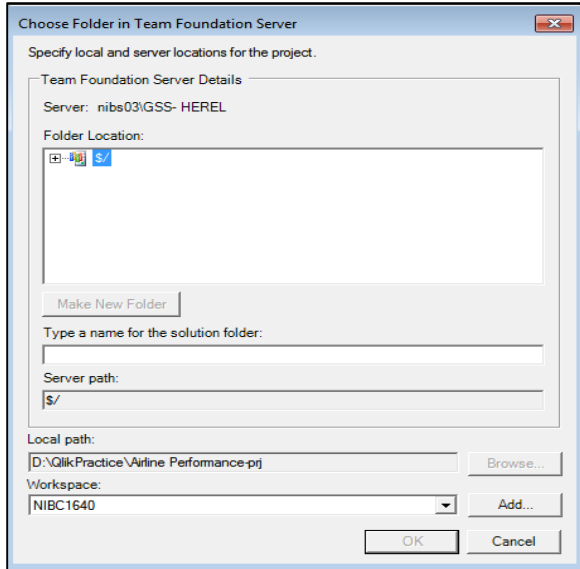
Start by opening the first document that you wish to add to source control, and select **File→Source Control→Add Project to Source Control**:
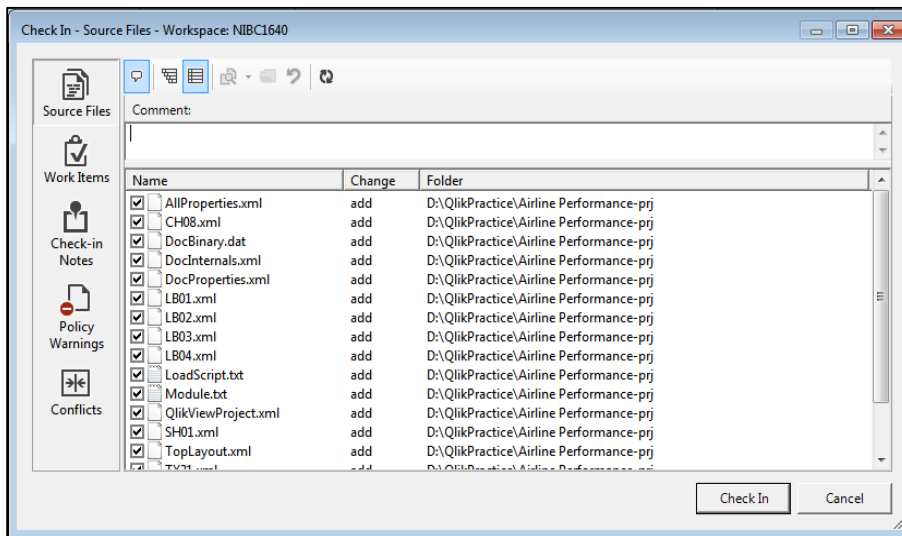

Select the **TFS Server** name:

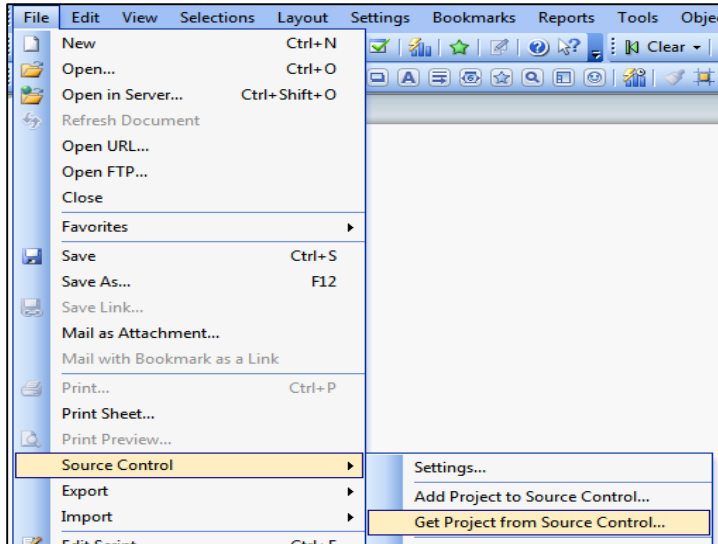Select the path where you want to place the app in TFS:

Once it is done, you will be redirected to below screen, specify the comments if you want and click on "Check In" button, you can see the files at TFS in "**Appname-Prj**" folder:

To retrieve app data from Source Control we need to select "**Get Project from Source Control**":


Once the fetch is done from TFS, the qlikview application does not exist. So, we need to create an empty qlikview app to proceed further, the procedure to create qvw file is as follows:

- Create a blank qvw file with the **same application name (Map_Test)** Map_Test.qvw (Start QlikView, File -> New, File -> Save As) in folder where Map_Test-prj is placed as shown below:
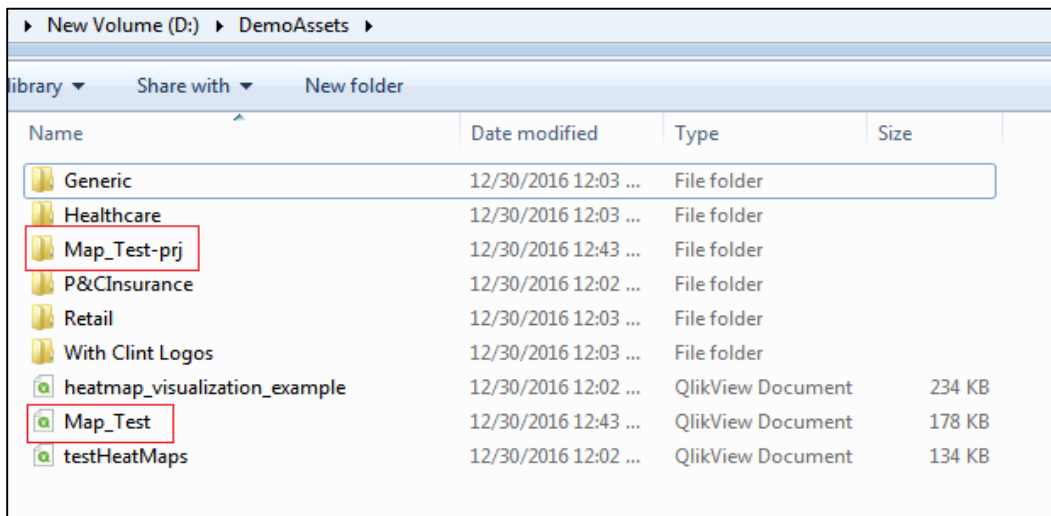
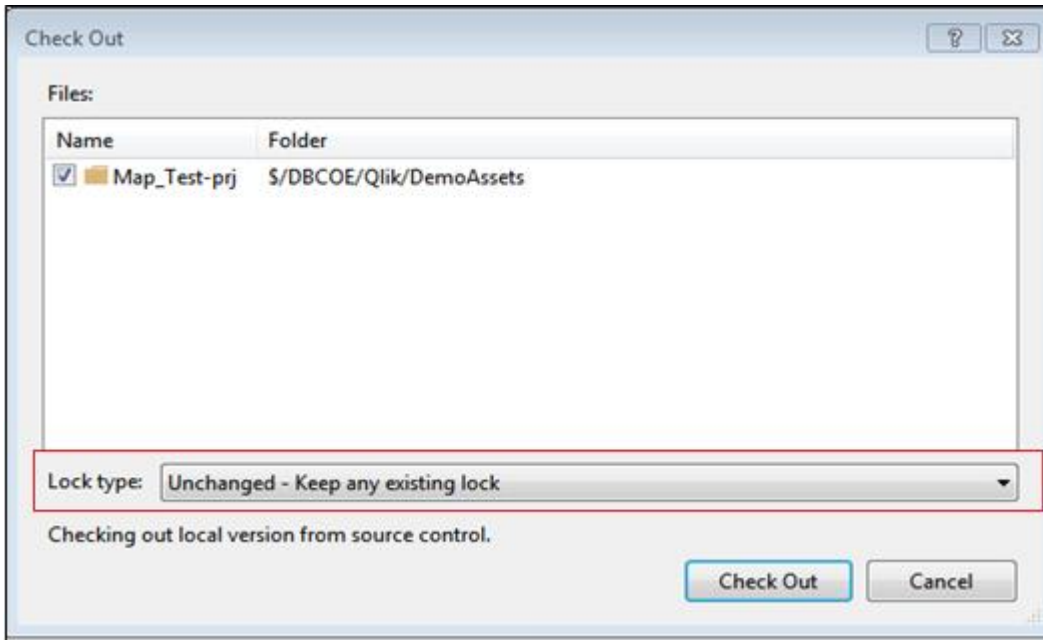- Click **No** button when asked to replace the project file in the below window:
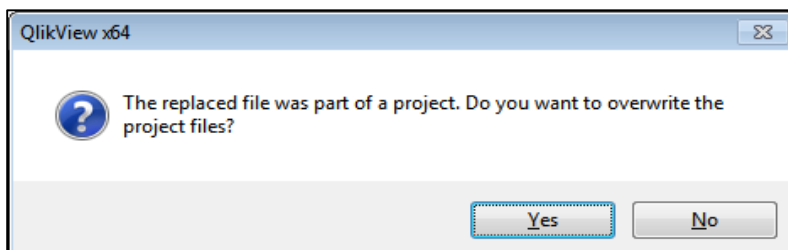
- Now, open Map_Test.qvw file in QlikView which will read the contents of the Map_Test-prj folder and update the Map_Test.qvw file. But to view the data, reload is to be done for the first time.

There are two types of check out which are differentiated by lock type:

Lock Type1:

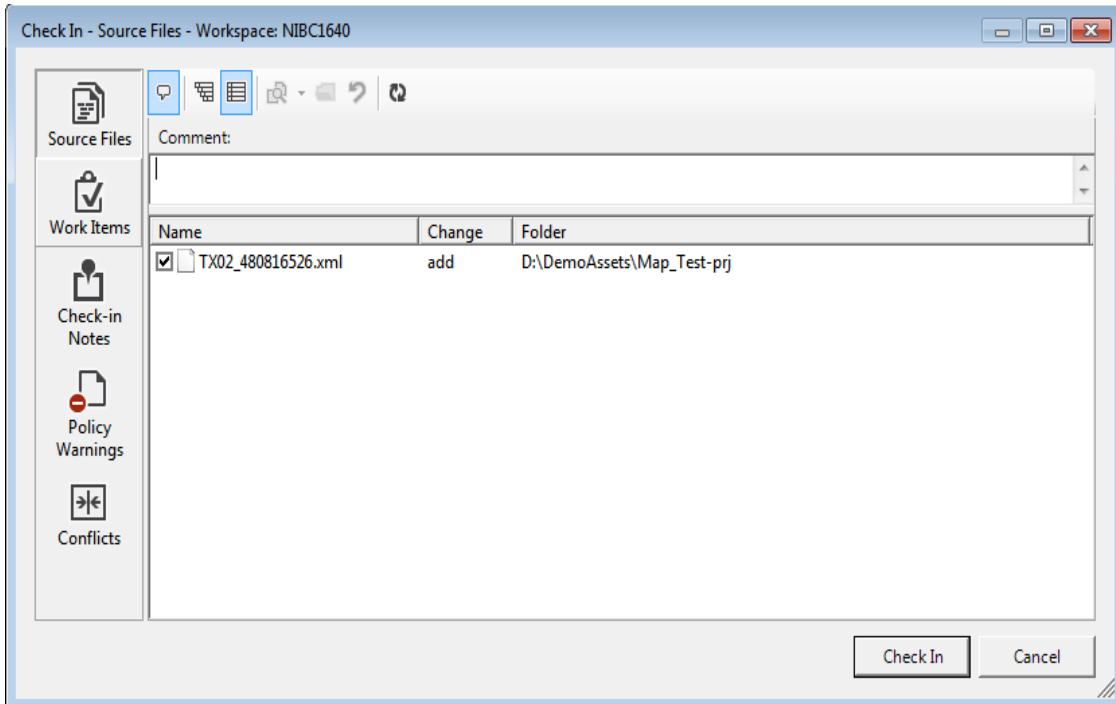➔ Check out for edit with the lock type as "**Unchanged – Keep any existing lock**" by 2 users as shown below:

➔ Both the users can try to edit the application and check-in the code at a time:
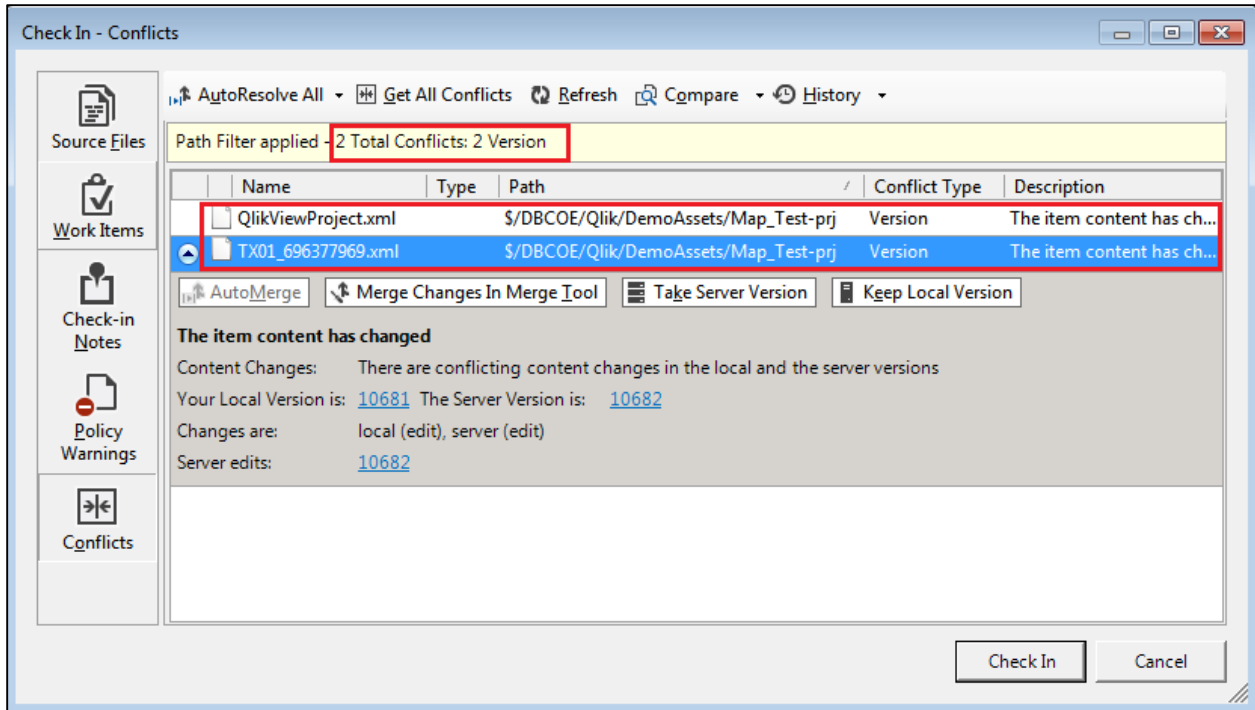
- New additions to the application will be added in first window:

- Changes done to existing objects are shown in next window:
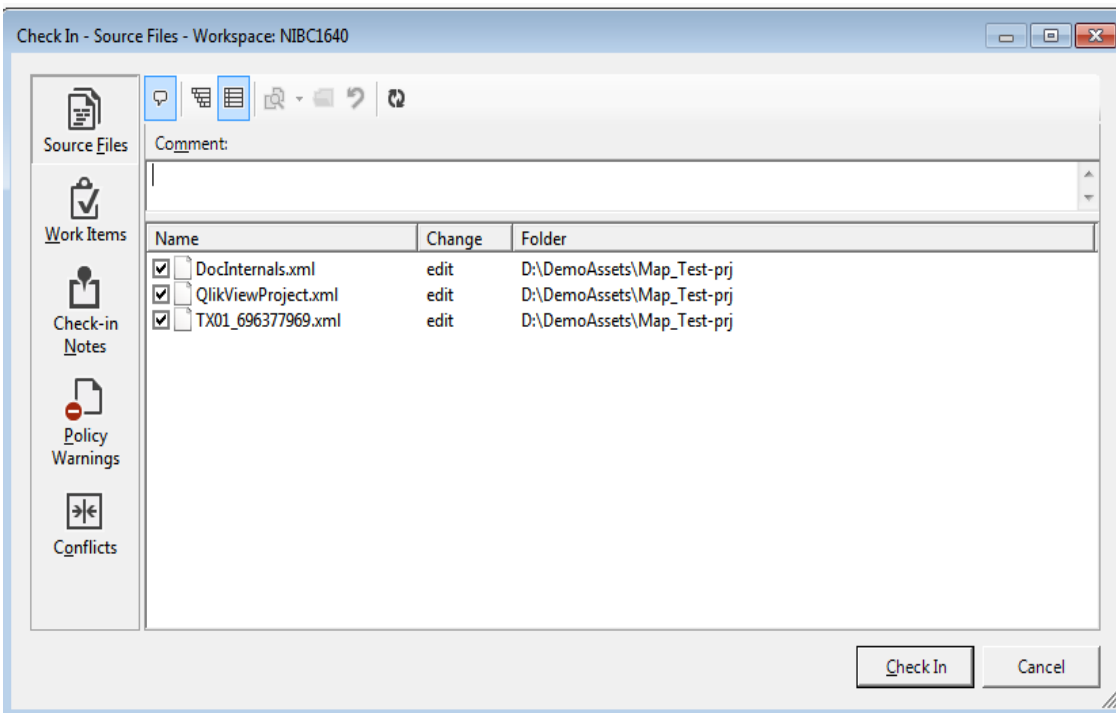
- The below window pops up when we try to check in same object **"TX01_696377969"/ "QlikViewProject.xml"** related changes by more than one user, which is a conflict. We need to **resolve it manually** by clicking on of the conflicts one after the other:
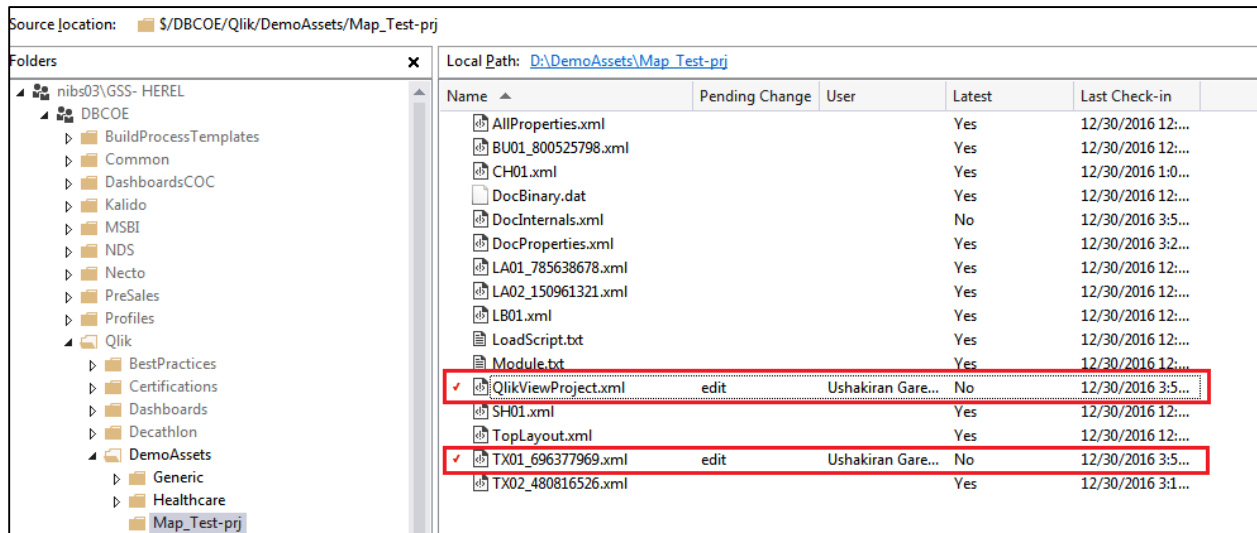
o When we click on the conflict below window is shown:
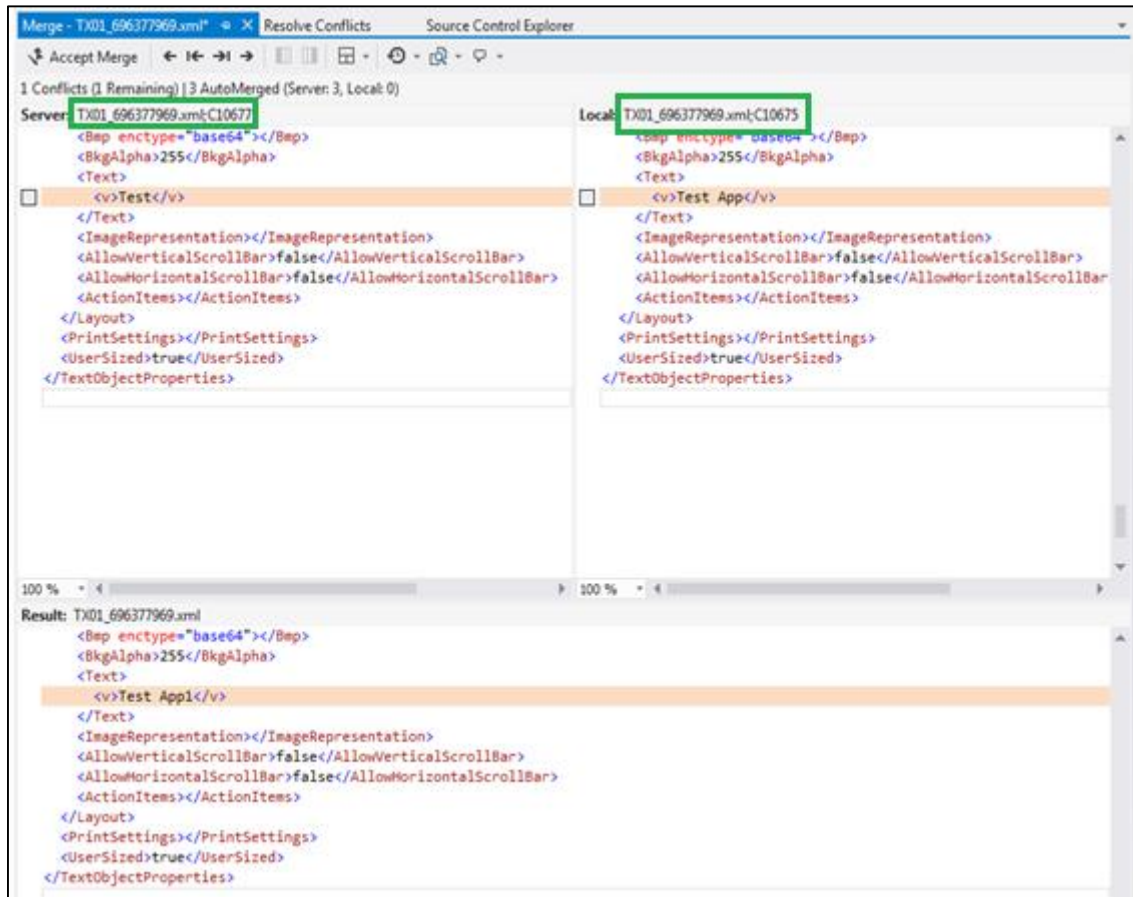
o We need to select which changes are to be kept further **Right or Left** (specified in Green color) and then click **"Accept merge"**:
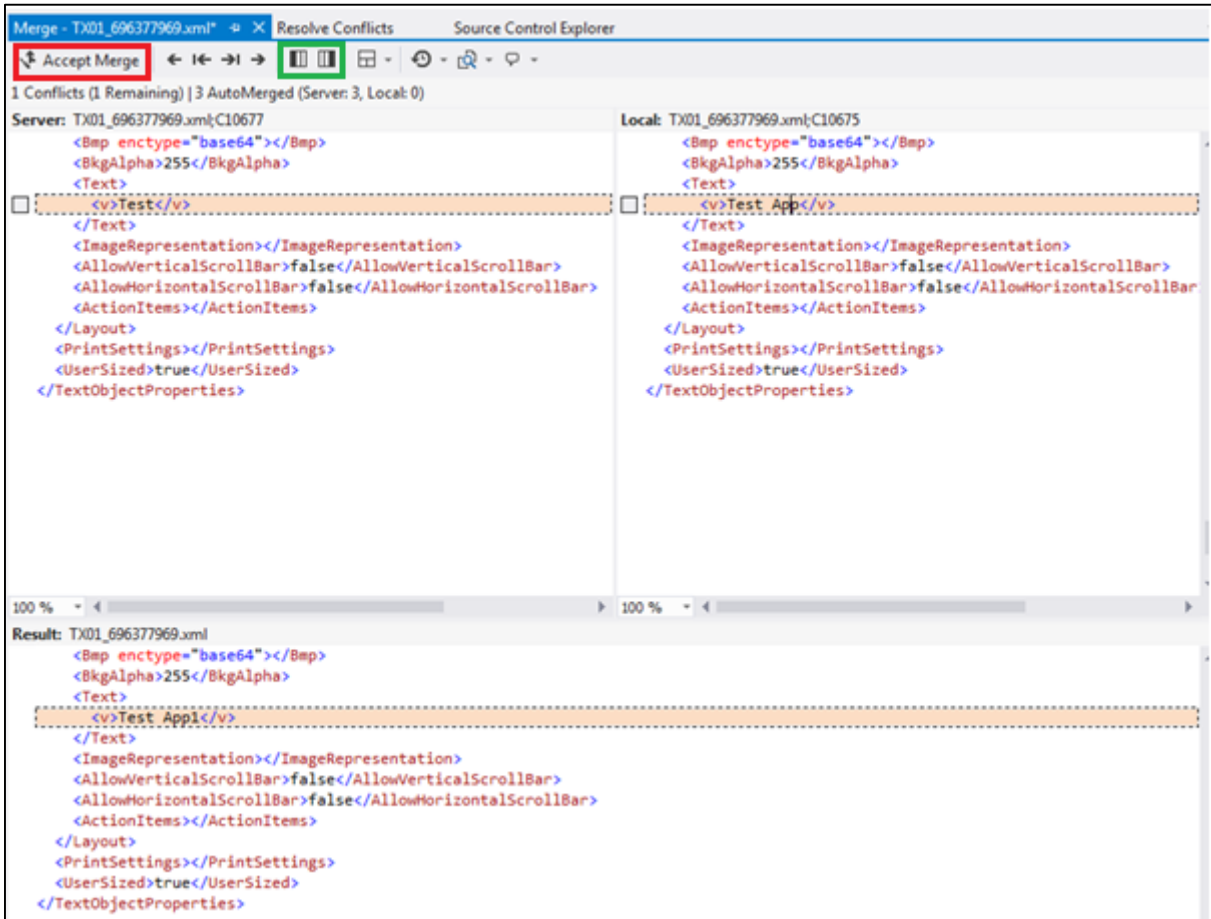
o We need to check-in conflict related files **from TFS** after merge as qlikview will be showing no files to be checked-in i.e., **project is up-to date**:
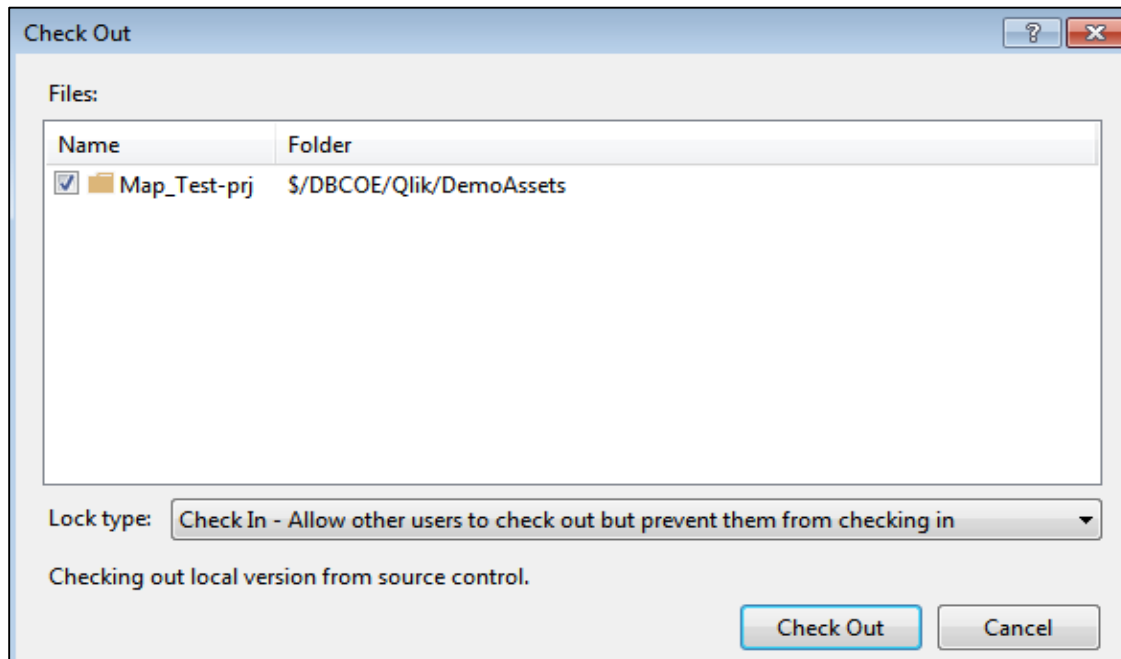
➔ Once the check-in is completed the application gets refreshed. We may see the below pop-up if it is local machine and not a Server. We need to click **"NO"** and proceed:
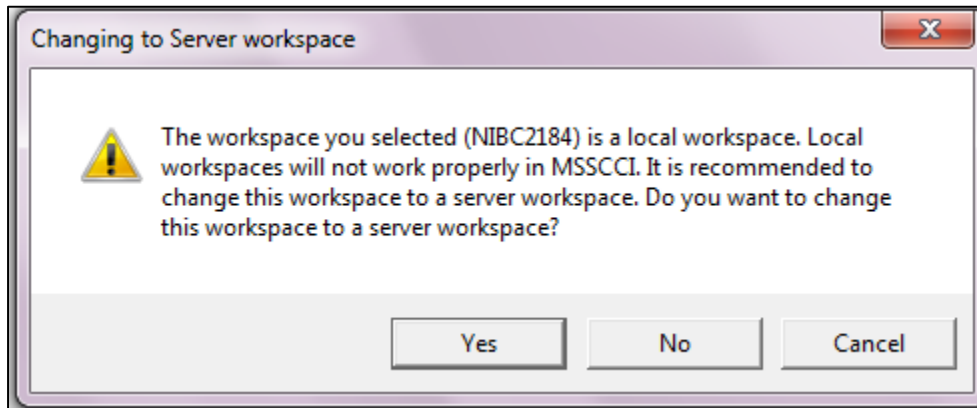
Lock Type2:

➔ Check out for edit with the lock type as "**Check In – Allow other users to check out but prevent them from checking in**" by 2 users as shown below:

In this type of scenario, the code gets locked out making other user not to do check-in until and unless the user who checked out check in the code as shown below:
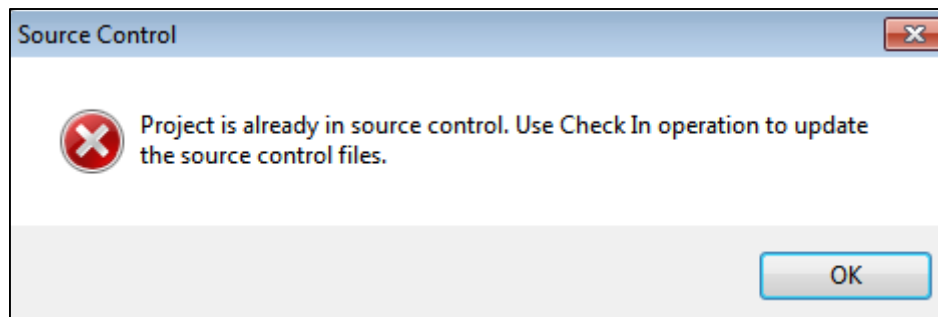
Issue 1: Project already in source control

If a App-prj folder is deleted in source control and checked-in, but it exists in any other user machine, following error will be displayed:

Best Practices:
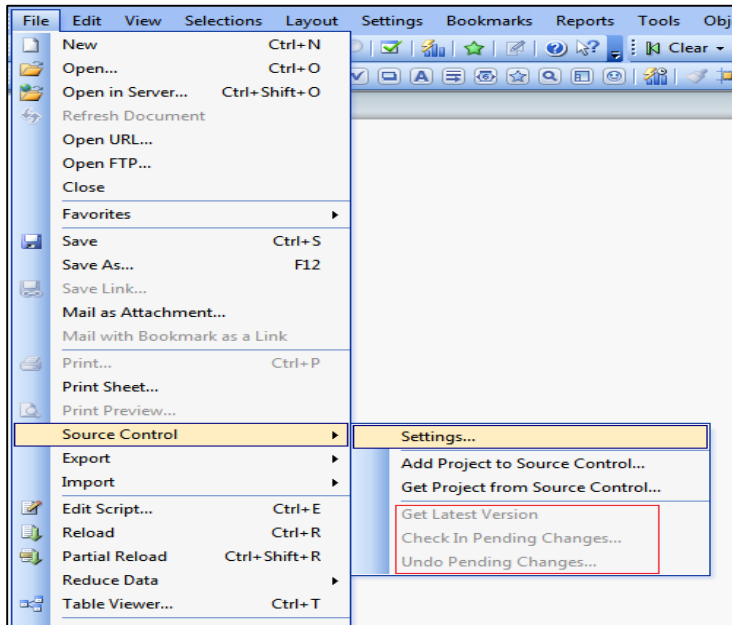
➔ Clean up the TFS temp files related to the App-prj (project) folder from Server.
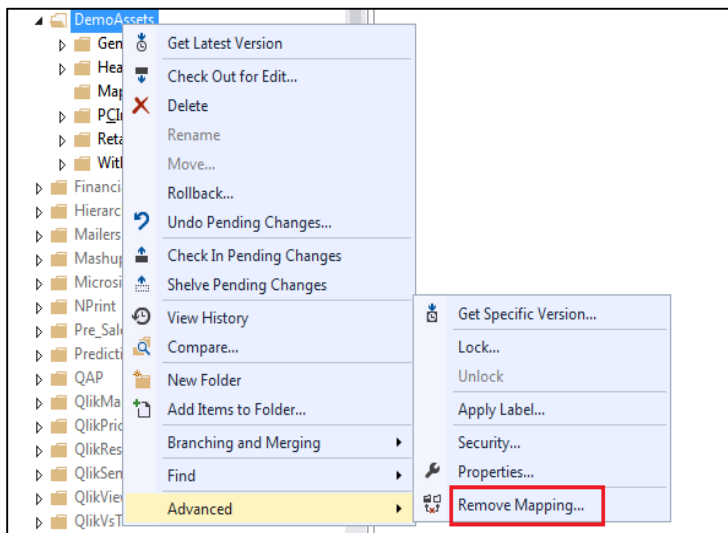➔ Rename the app and do a check-in.

Issue 2: Incorrect TFS mapping

If the local mapping for TFS is wrong, we will not be able to see the options highlighted as below:

Best Practice:

➔ Remap the projects to the correct location in TFS and get the latest version of the app from QlikView desktop.

Issue 3: Empty Qlikview App Creation at incorrect location

This scenario may occur in few scenarios, when an empty qlikview application is created elsewhere rather than the Source control mapping folder. The app still remains empty, not showing the objects.

Best Practice:

➔ Create a new app in the Source control mapping folder instead of creating it some other location and copying it.

Issue 4: Document addition to Source Control

In menu option there is the add project to source control enabled, but when clicked on it nothing happens, and the usual pop up window doesn't appear. A -prj folder is created in the folder where the document is stored, but it is empty. Any attempts afterwards to add it to source control result in an error saying that it cannot find the Qlikview Project file in the -prj folder.

Best Practices:

➔ Delete the -prj folder and save the application.
➔ Re-create the application.

Reference: https://community.qlik.com/thread/215848