BUILD DEFINITION EXTENSION

Contents

Introduction

Thank you for choosing xeam Build Definition Extension for Microsoft Visual Studio 2012^{TM}. Xeam Build Definition Extension improves your handling with Build Definitions in the following areas:

- Copy Build Definitions
- Merge Build Definitions
- Branch Build Definitions
- Save and Restore Build Definitions in Source Control
- Tracking changes and modifications for Build Definitions
- History for Build Definitions
- Rollback of Build Definition changes
- Import/Export Build Definitions (e.g. between On Premise and Visual Studio Online)

Once you have installed xeam Build Definition Extension, changes made to build definitions are automatically tracked in TFS Source Control.

Setup

Installation

Install xeam Build Definitions Extension using Xeam.VisualStudioExtensions.BuildDefinition.vsix. You can find the extensions in Visual Studio Gallery or for download under http://www.xeam-solutions.com/products/build-definition-extension.html.

During startup of Visual Studio^{TM} a trial version screen will be shown. If you want to try out Build Definitions Extension click “Continue” and you will get 30 days trial version.

If you already bought a license, you can activate it by clicking “Enter License Key” in the trial dialog. Copy and Paste the content of the license file into the text field and click “Save license”. A license is required for every machine where you install xeam Build Definition Extension. You are not allowed to share a license over multiple machines. You can access the license dialog from Tools->Build Definition Extension.

Configuration

The Build Definitions files used for save, merge, history will be stored under “$/<project>/BuildProcessTemplates/BuildDefinitions”. Make sure that the users who has rights to edit build definitions on a project, has also sufficient rights on this source folder, especially the rights add / check in / check out / delete / merge. These rights are available for Build Administrators by default.

To come around workspace mapping issues and conflicts, xeam Build Definition Extension creates its own workspace (Laika42_BuildDef<guid>) under which source control operations are performed. Please do not change the mappings within this workspace, nor use it for general work.

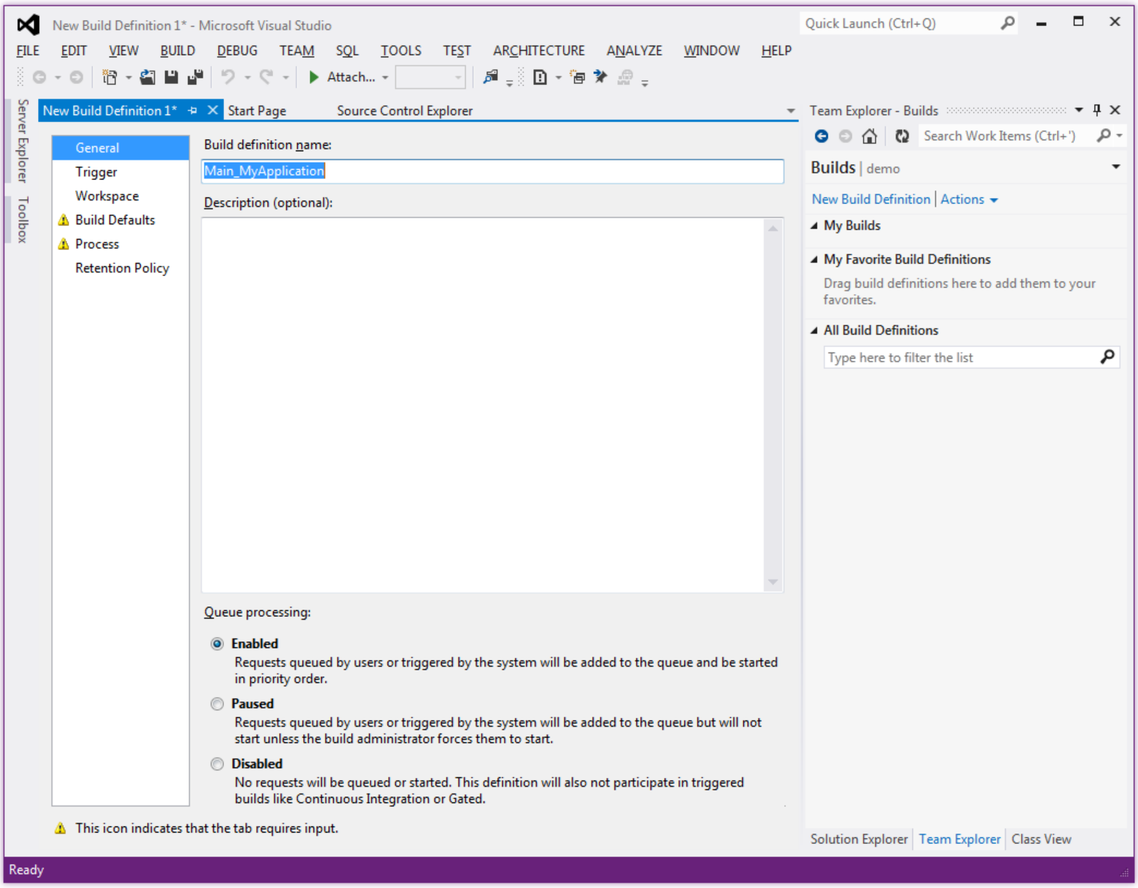
Attention: For tracking history without loss of information, xeam Build Definition has to be installed on every Visual Studio instance, where build definitions are changed.

Quick Start Guide

Create a new Build Definition

After installing xeam Build Definition Extension, create a new build definition in your TFS project.

In team explorer select Builds -> New Build Definition
Setup your build definition parameters as normal.


New Build Definition 1* - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
FILE EDIT VIEW BUILD DEBUG TEAM SQL TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Attach...
Server Explorer
Toolbox
New Build Definition 1*
Start Page
Source Control Explorer
General
Trigger
Workspace
⚠ Build Defaults
⚠ Process
Retention Policy
Team Explorer - Builds
Search Work Items (Ctrl+')
Builds | demo
New Build Definition | Actions
My Builds
My Favorite Build Definitions
Drag build definitions here to add them to your favorites.
All Build Definitions
Type here to filter the list
Solution Explorer
Team Explorer
Class View
Ready

Build definition name: Main_MyApplication

Description (optional):

Queue processing:

● **Enabled**
Requests queued by users or triggered by the system will be added to the queue and be started in priority order.

○ **Paused**
Requests queued by users or triggered by the system will be added to the queue but will not start unless the build administrator forces them to start.

○ **Disabled**
No requests will be queued or started. This definition will also not participate in triggered builds like Continuous Integration or Gated.

⚠ This icon indicates that the tab requires input.

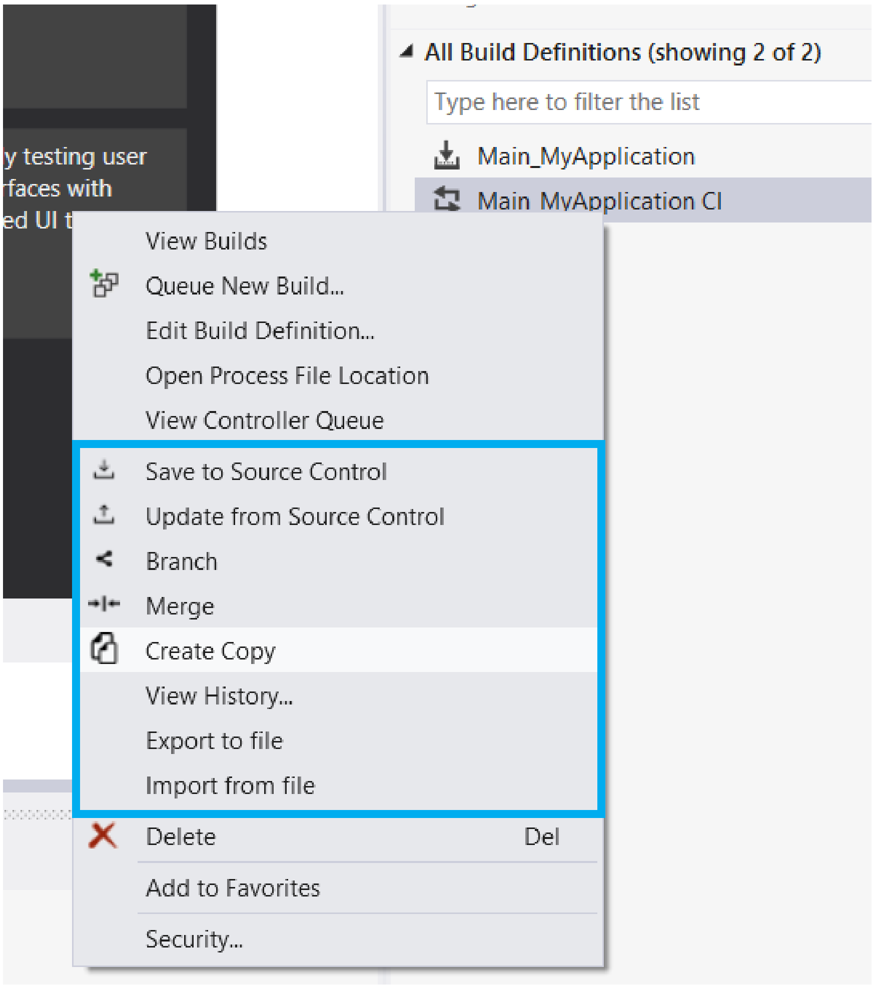
Save your build definition by clicking the save button. The new build definition is stored as normal. In addition, a file in source control is created ($/<project>/buildprocesstemplates/builddefinitions/<build definition name>.builddef), where changes to the build definitions are tracked automatically. You do not have to care about this location for normal usage.

View History of Build Definition

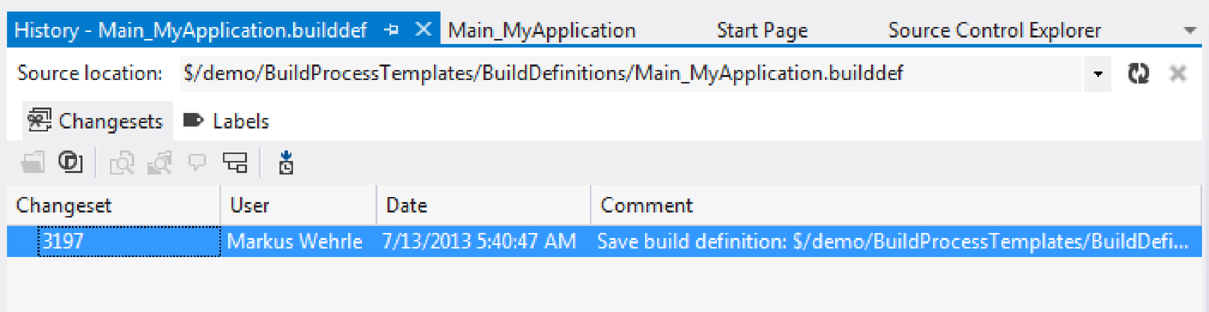
After saving the build definition, it will appear in Build Explorer as normal.

Right click the new created build definition.

You will recognize, that the popup menu has been extended with the options “Save to Source Control”, “Update from Source Control”, “Branch”, “Merge”, “Create Copy”, “View History…”, “Export to file” and “Import from file”.



By clicking “View History…”, a changeset history is shown for the selected build definition, where you can see, which user has added the build definition. You can also use the default “View History” feature provided by Visual Studio on the file in Source Control.



Source location: $/demo/BuildProcessTemplates/BuildDefinitions/Main_MyApplication.builddef

Changeset	User	Date	Comment
3197	Markus Wehrle	7/13/2013 5:40:47 AM	Save build definition: $/demo/BuildProcessTemplates/BuildDefi...

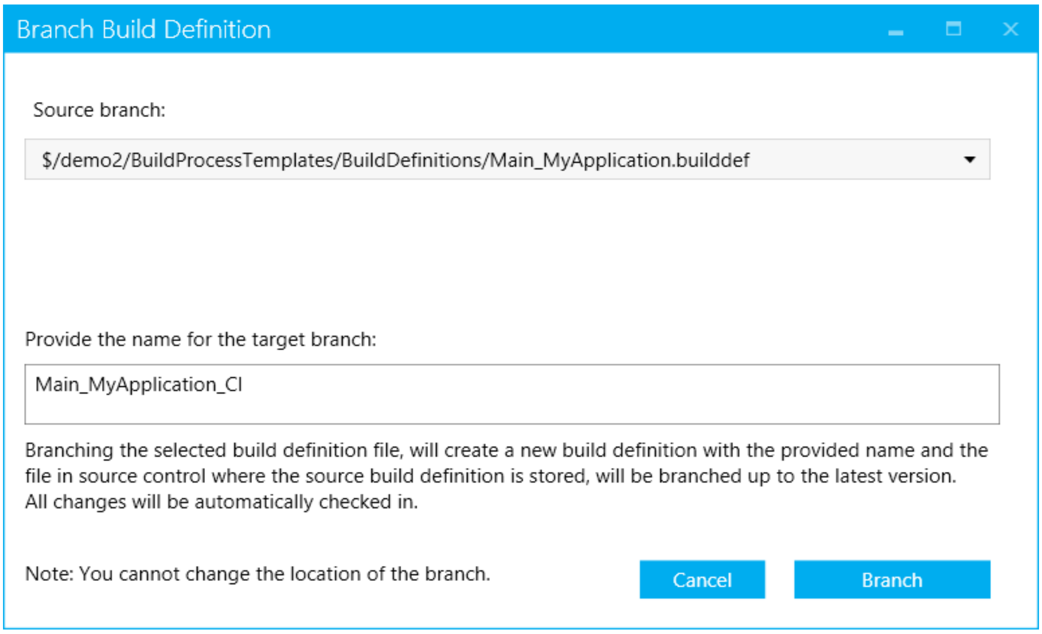
Branch Build Definition

If you want to have two build definitions connected for being able to merge them, you can branch your build definition.

Assume, that you want to have a continuous integration and a nightly scheduled build linked together. Your nightly build was created in the previous steps and scheduled e.g. at 3am every night.

Right Click your Build definition -> Branch

A branch dialog appears. Give your new Build definition a name, e.g. Main_MyApplication_CI.



Branch Build Definition

Source branch:

$/demo2/BuildProcessTemplates/BuildDefinitions/Main_MyApplication.builddef

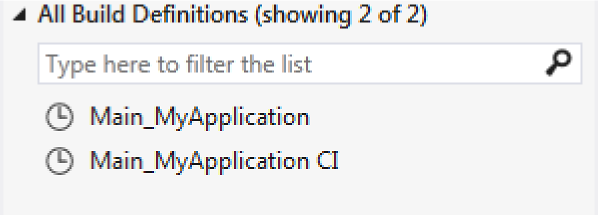
Provide the name for the target branch:

Main_MyApplication_CI

Branching the selected build definition file, will create a new build definition with the provided name and the file in source control where the source build definition is stored, will be branched up to the latest version.
All changes will be automatically checked in.

Note: You cannot change the location of the branch.

Click Branch and the new build definition will appear in build explorer.



All Build Definitions (showing 2 of 2)

Main_MyApplication

Main_MyApplication CI

You now can edit your CI build definition to define a different build trigger. During the branch process, a linked clone has been created. So as branch result you will get a copy of your build definition.

Right click the CI build definition -> Edit build definition…

Goto Trigger -> select Continues Integration

Click Save

Merge Build Definition

After you have created the branched build definition, we assume that you have to modify you nightly build definition, which we created a few steps before. We also assume, that you have to add a new solution to the build definition. In the sample we will add a solution named “My Class Library”

Right click your build definition -> Edit Build Definition…

Process -> items to build

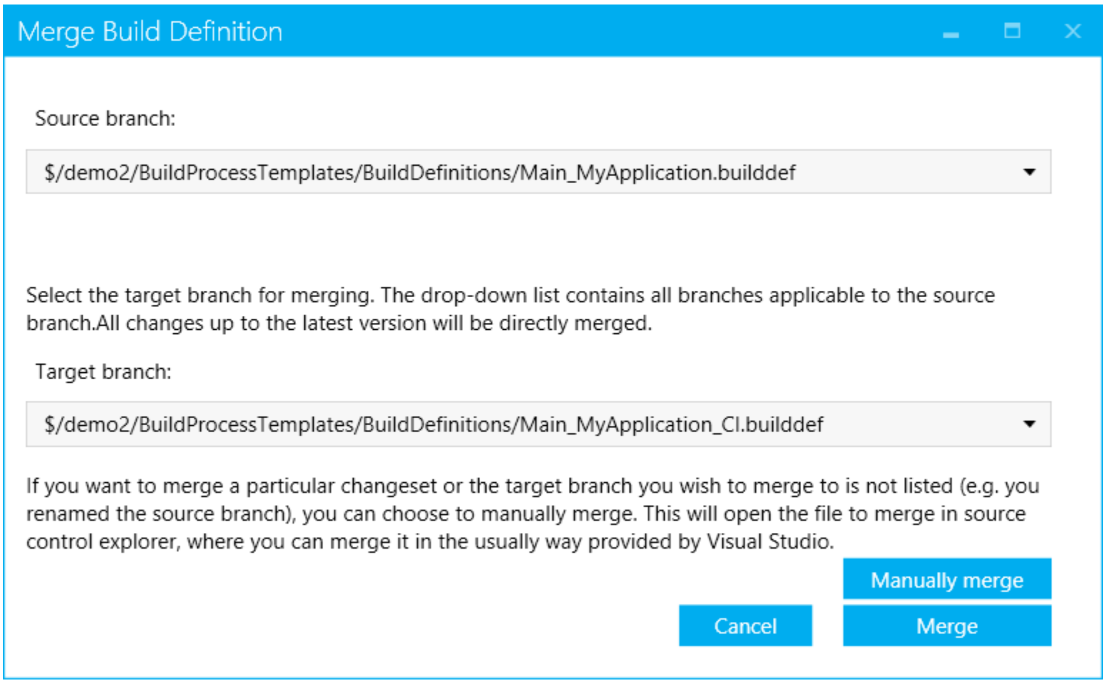
Add your solution

Click Save

After you made changes to the build definition

Right click your build definition -> Merge

The following Merge Build Definition dialog comes up



Source branch: $/demo2/BuildProcessTemplates/BuildDefinitions/Main_MyApplication.builddef

Select the target branch for merging. The drop-down list contains all branches applicable to the source branch.All changes up to the latest version will be directly merged.

Target branch: $/demo2/BuildProcessTemplates/BuildDefinitions/Main_MyApplication_CI.builddef

If you want to merge a particular changeset or the target branch you wish to merge to is not listed (e.g. you renamed the source branch), you can choose to manually merge. This will open the file to merge in source control explorer, where you can merge it in the usually way provided by Visual Studio.

Select the target build definition and click merge. All changes made to the source build definition are merged to the target build definition. If you only want to merge single changes you can use “Manually merge”.

Click Merge

The changes will be merged. Inspect your target build definition using “Edit build definition…” and you will recognize, that the added solution appears in process, but no change happened to your CI configuration.

Save and Update Build definition

To get a history or if you want to save your build definitions for backup reasons, you can save your existing build definitions to source control by right clicking -> Save to Source Control.

You can restore your build definition configuration from the file stored in source control by right clicking your build definition -> Update from Source Control.

Import/Export Build Definition

Beside the simple “Create Copy” functionality, “Import/Export” offers you the possibility to copy a build definition to **another team project** in your TFS or **even to a team project of another TFS**.

Right click your build definition -> Export to file

In the “Select or enter a file name” dialog provide a name.

Click Save

Connect to the team project where you want to import the build definition and select one build definition.

Right click the selected build definition -> Import from file

In the “Select a build definition to import” dialog navigate to the exported build definition in the previous step.

Click Open -> Done.

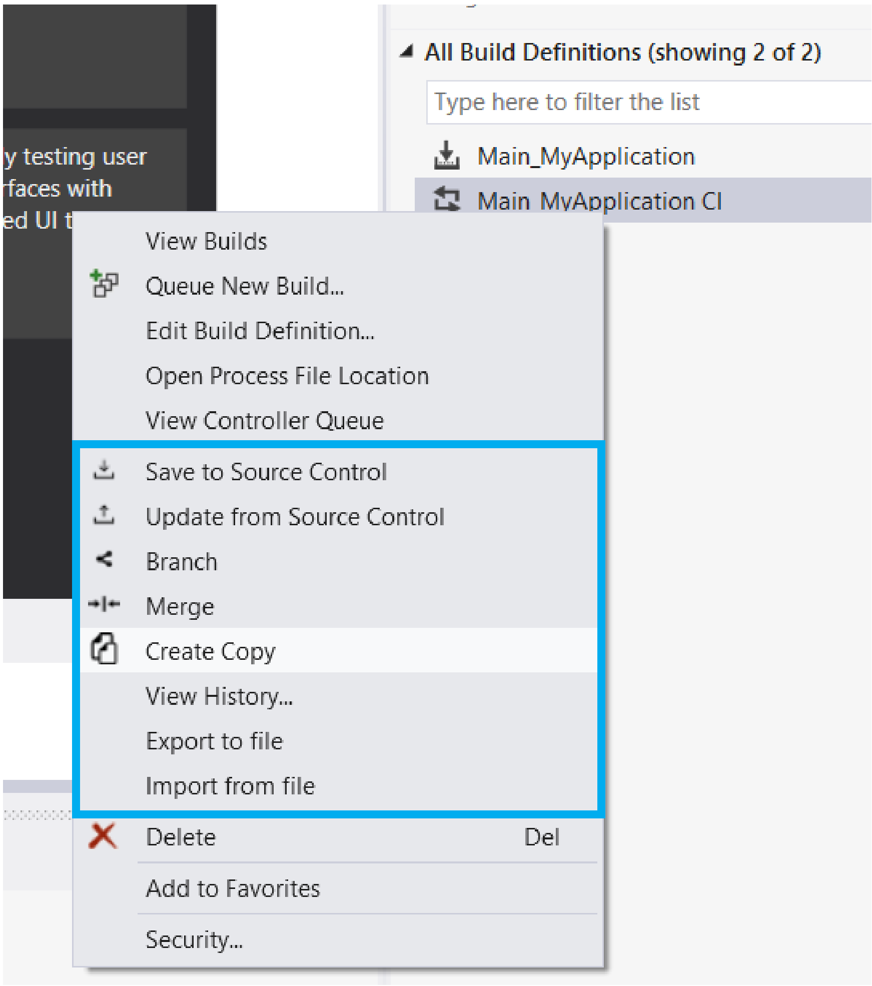
The list of your build definition will be refreshed and you will see the imported build definition.

Note: if the process template of the exported build definition does not exist in the team project where you want to import it, you will have to add it first.

Basic Functionality

Build Definition Context Menu

The main functionality of xeam Build Definition Extension can be accessed in using the build definition context menu.



In the following sections the different commands are described in detail

Save to Source Control

The "Save to Source Control" command stores the selected build definition and its configuration options in source control of the current selected team foundation server project under $/<team project>/BuildProcessTemplate/BuildDefinitions/<builddefinition name>.builddef.

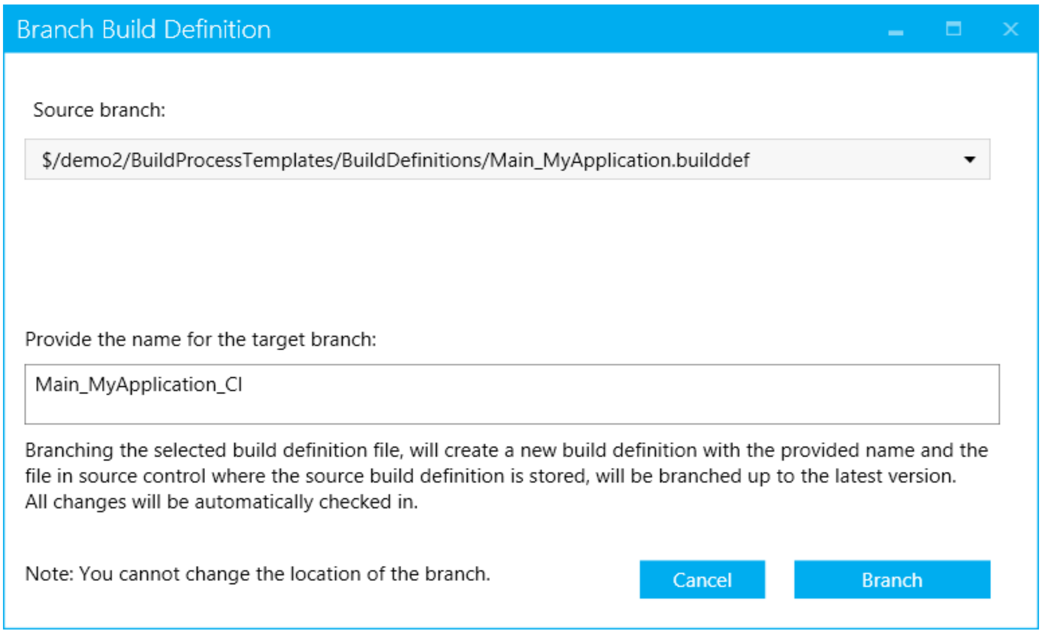
Update from Source Control

The "Update from Source Control" command loads the build definition configuration from source control and updates the selected build definition options. **Attention:** All properties of the build definition will be changed to the values set in the source file.

Branch

The "Branch" command provides the ability to branch a build definition. Use this command, whenever you want to have two build definitions linked together and you want them to be mergeable. The branch strategy of build definitions can be different from your branch strategies of your project. See "Enhances Scenarios -> branch strategies" for more information.

To branch a build definition, click on the "Branch" in context menu. A branch dialog will appear.



Branch Build Definition

Source branch:

$/demo2/BuildProcessTemplates/BuildDefinitions/Main_MyApplication.builddef

Provide the name for the target branch:

Main_MyApplication_CI

Branching the selected build definition file, will create a new build definition with the provided name and the file in source control where the source build definition is stored, will be branched up to the latest version. All changes will be automatically checked in.

Note: You cannot change the location of the branch.

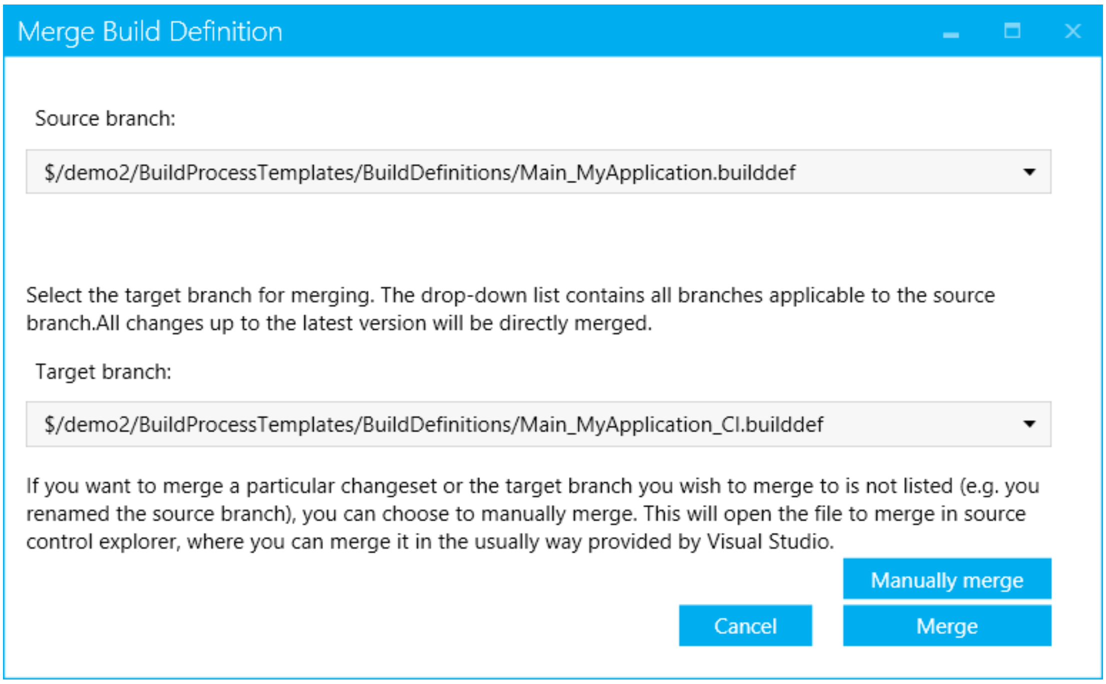
Provide a name for the branched build definition and click Branch.

During branching, a clone of the source branch build definition is created and you will see your branched build definition in build explorer.

The connection between the 2 branches is done using build definition files that are created in source control. These 2 files are linked together as they are linked, when a file based branch is created in source control.

Merge

With the "Merge" command you can merge 2 build definitions that previously have been branched by the Branch command. If you select "Merge" in context menu of a build definition, the Build Definition Merge Dialog appears.



Source branch:

$/demo2/BuildProcessTemplates/BuildDefinitions/Main_MyApplication.builddef

Select the target branch for merging. The drop-down list contains all branches applicable to the source branch.All changes up to the latest version will be directly merged.

Target branch:

$/demo2/BuildProcessTemplates/BuildDefinitions/Main_MyApplication_CI.builddef

If you want to merge a particular changeset or the target branch you wish to merge to is not listed (e.g. you renamed the source branch), you can choose to manually merge. This will open the file to merge in source control explorer, where you can merge it in the usually way provided by Visual Studio.

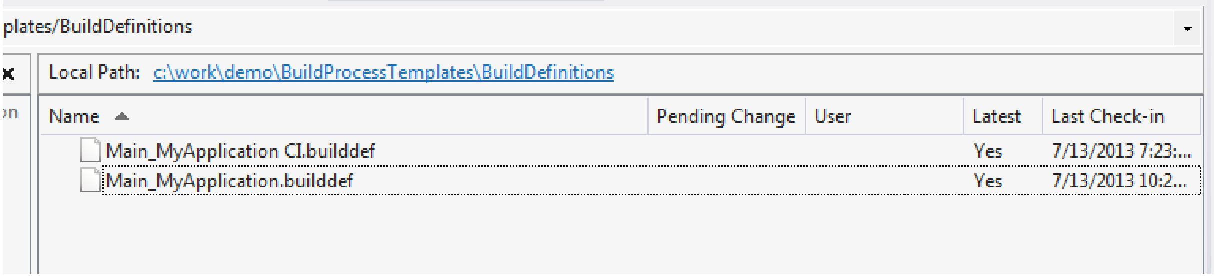
In the source branch list, select the build definition you want to merge. In the target branch list select the build definition on which you want to apply the changes.

Auto Merge

By clicking "Merge", all changes up to the latest version will be merged from source to target and your target build definition will be updated. The merge operation correspond to "All changes up to specific version" -> "latest version" when you merge files in source control. If a merge conflict occurs a message will be displayed and you can then choose if you still want the files to be merged automatically. In case of conflicts, we recommend you to merge the files manually, since in special cases, the target build definition file could contain invalid data after the merge, caused by the auto merge feature of TFS. If you still choose to auto merge and if then some conflicts cannot be merged, then you have to go through the manual merge process (the default merge feature provided by Visual Studio).

Manual Merge

If you want to have more control over merging, you have the ability to perform a manual merge on the build definition files by clicking “Manually merge”. If you click manual merge Source Control will open in the folder, where the build definitions are stored.



plates/BuildDefinitions

Local Path: c:\work\demo\BuildProcessTemplates\BuildDefinitions

Name	Pending Change	User	Latest	Last Check-in
Main_MyApplication CI.builddef			Yes	7/13/2013 7:23:...
Main_MyApplication.builddef			Yes	7/13/2013 10:2...

The build definition files in Source Control have the same names as your build definition. Get latest version of the build definition files you want to merge and use the normal merge mechanism of source control. E.g. you can use merge single change sets there. Check in your changes after you have merged your changes.

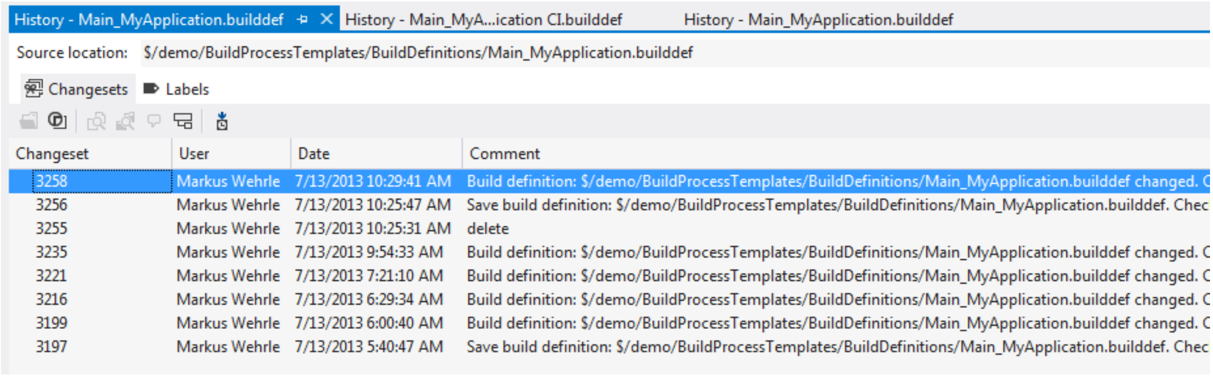
Attention: After you have merged your changes in source control, you have to update your target build definition using the “Update from Source Control” command. Also, if the merged build definition is opened in the Build Definition Editor in Visual Studio, you have to close and reopen it, to see the changes (The refresh on the opened build definitions in the editor is not made automatically).

Create Copy

If you just want to make an unconnected copy of your build definitions you can use the “Create Copy” command. In comparison to the Clone command you get with the Team Foundation Power Tools, xeam Build Definition Extension Create Copy command will automatically save the clone in source control.

View History…

Since all changes done to build definitions are tracked in source control when xeam Build Definition Extension is installed, you automatically get a history of the changes. If you want to see the history of changes and who did the changes you can use “View History…” command. A Source Control Window will open, showing the change set history of the selected build definition.



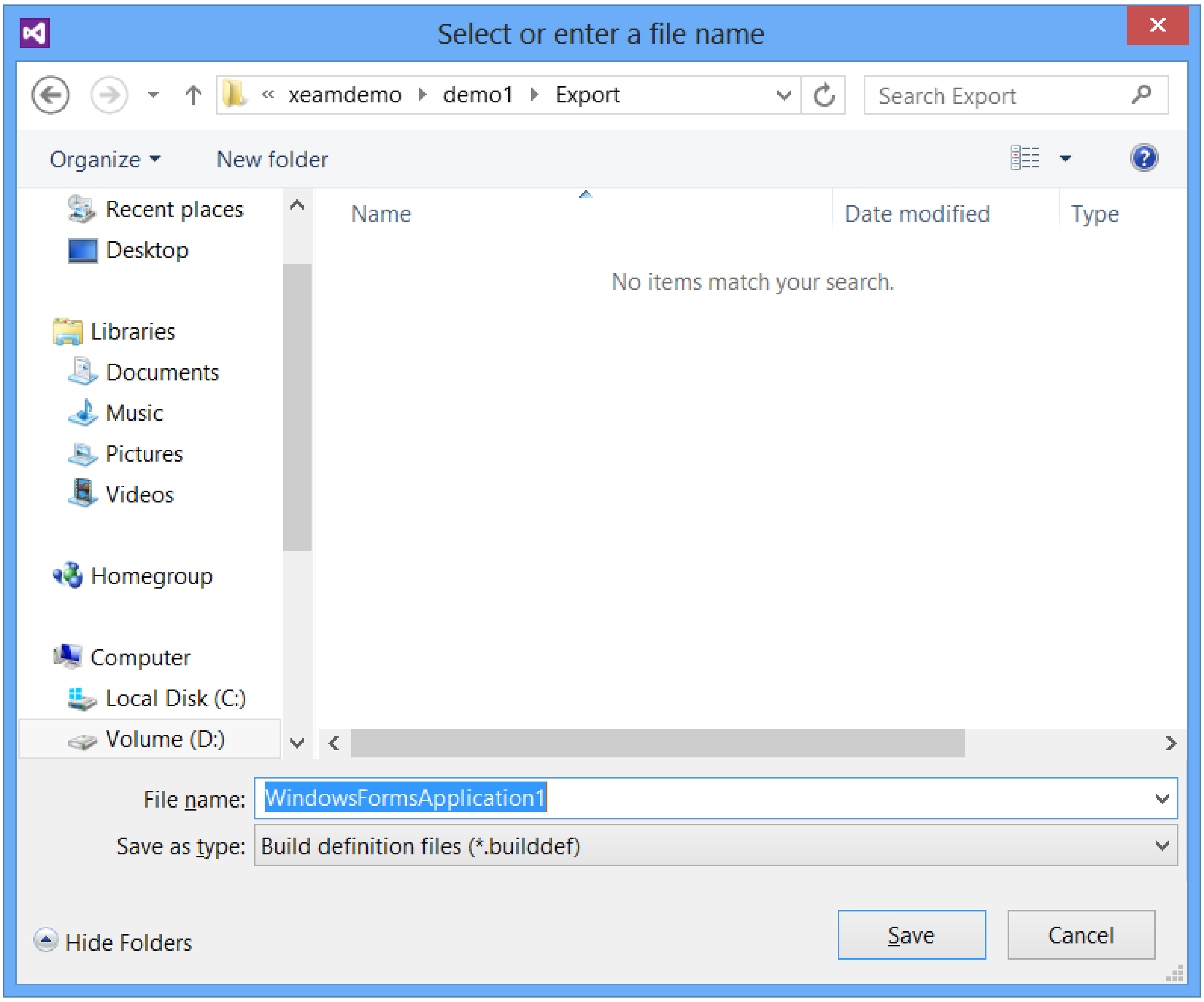
History - Main_MyApplication.builddef
History - Main_MyA...ication CI.builddef
History - Main_MyApplication.builddef
Changesets
Labels

Source location: $/demo/BuildProcessTemplates/BuildDefinitions/Main_MyApplication.builddef

Changeset	User	Date	Comment
3258	Markus Wehrle	7/13/2013 10:29:41 AM	Build definition: $/demo/BuildProcessTemplates/BuildDefinitions/Main_MyApplication.builddef changed. C
3256	Markus Wehrle	7/13/2013 10:25:47 AM	Save build definition: $/demo/BuildProcessTemplates/BuildDefinitions/Main_MyApplication.builddef. Chec
3255	Markus Wehrle	7/13/2013 10:25:31 AM	delete
3235	Markus Wehrle	7/13/2013 9:54:33 AM	Build definition: $/demo/BuildProcessTemplates/BuildDefinitions/Main_MyApplication.builddef changed. C
3221	Markus Wehrle	7/13/2013 7:21:10 AM	Build definition: $/demo/BuildProcessTemplates/BuildDefinitions/Main_MyApplication.builddef changed. C
3216	Markus Wehrle	7/13/2013 6:29:34 AM	Build definition: $/demo/BuildProcessTemplates/BuildDefinitions/Main_MyApplication.builddef changed. C
3199	Markus Wehrle	7/13/2013 6:00:40 AM	Build definition: $/demo/BuildProcessTemplates/BuildDefinitions/Main_MyApplication.builddef changed. C
3197	Markus Wehrle	7/13/2013 5:40:47 AM	Save build definition: $/demo/BuildProcessTemplates/BuildDefinitions/Main_MyApplication.builddef. Chec

Export to file

To export a build definition to a file click “Export to file”. The following dialog is shown:



No items match your search.

File name: WindowsFormsApplication1

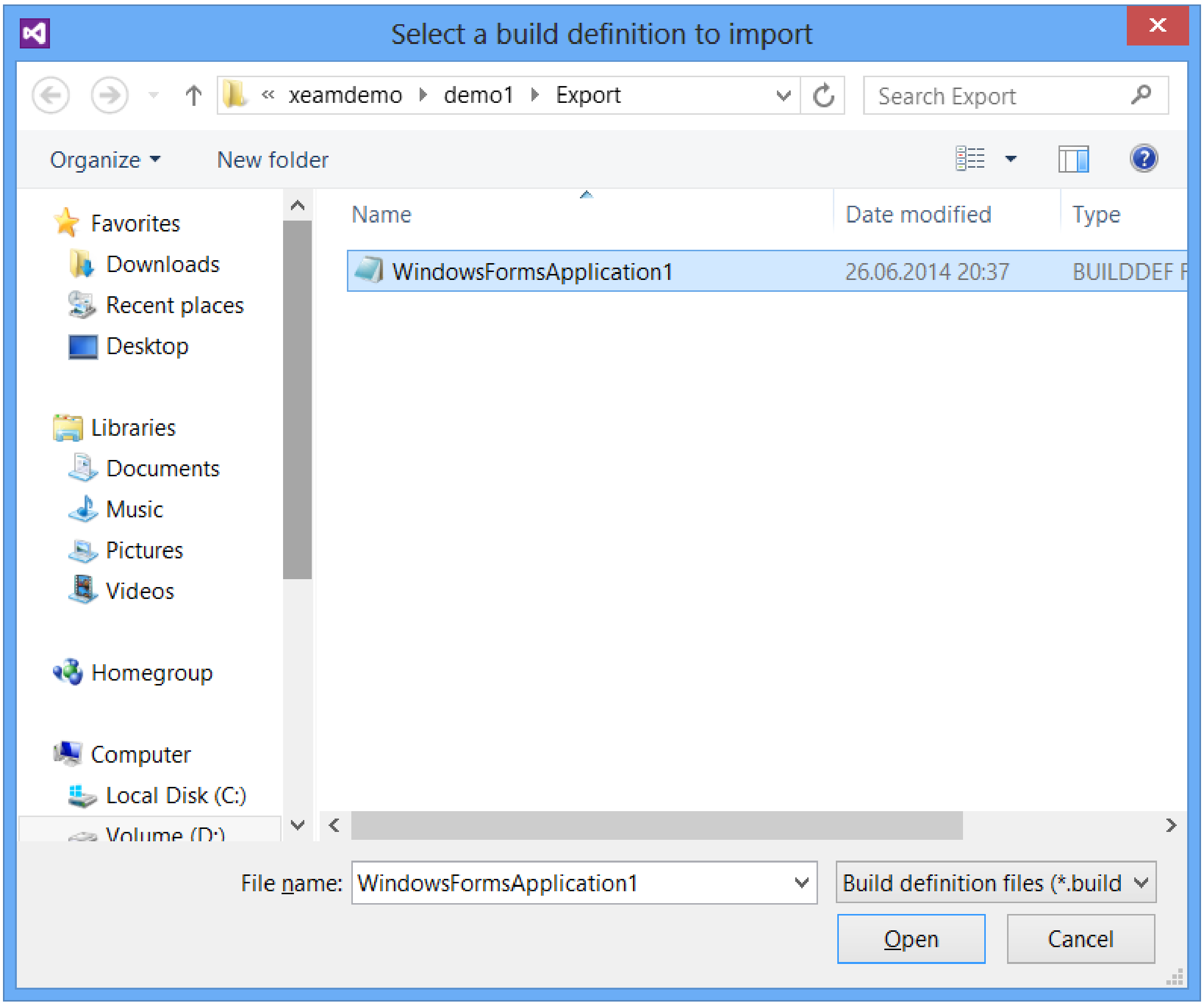
Save as type: Build definition files (*.builddef)

Provide a name for the build definition file and click “Save”.

Import from file

Connect to the TFS and navigate to the team project where you want to import a build definition.

Select one build definition (it doesn't matter which one). Right click on it and click "Import". The following dialog is shown:



Name	Date modified	Type
WindowsFormsApplication1	26.06.2014 20:37	BUILDDEF F

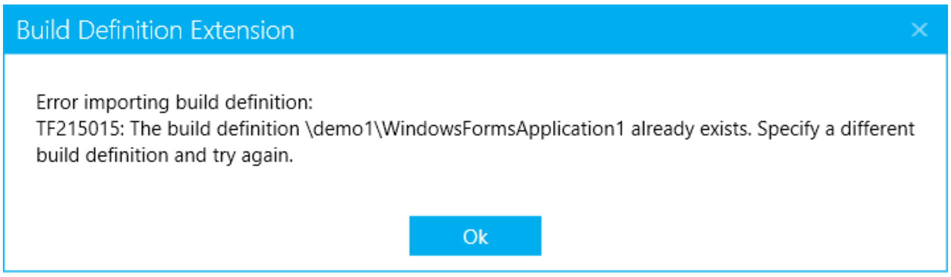
File name: WindowsFormsApplication1

Build definition files (*.build

Navigate to the build definition file you want to import. Select it and click "Open".

Naming conflict

If a build definition with the same name already exists in the team project, a error message will be shown and the build definition will not be imported.

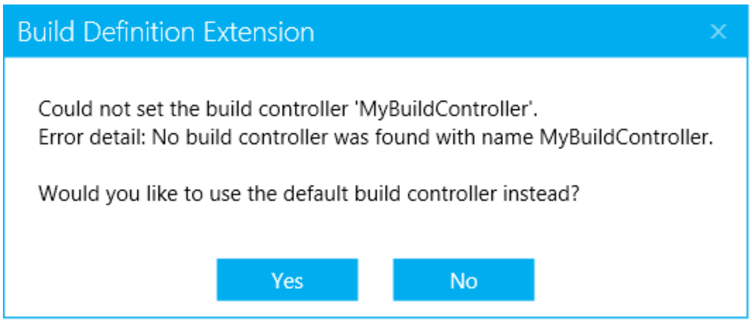


Error importing build definition:
TF215015: The build definition \demo1\WindowsFormsApplication1 already exists. Specify a different build definition and try again.

You can open the .builddef file in any editor and change the value of the <Name> node. Save the file and import it again.

Build Controller

If you try to import a build definition which uses a build controller that is not available in the project where you import the build definition, the following message is shown:



Could not set the build controller 'MyBuildController'.
Error detail: No build controller was found with name MyBuildController.

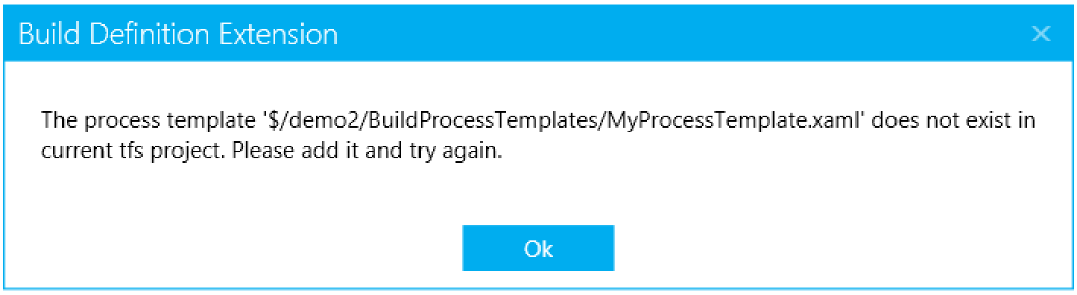
Would you like to use the default build controller instead?

Clicking “Yes” will set the default build controller of the team project for the imported build definition.

Clicking “No” will stop the import process and the build definition will not be imported.

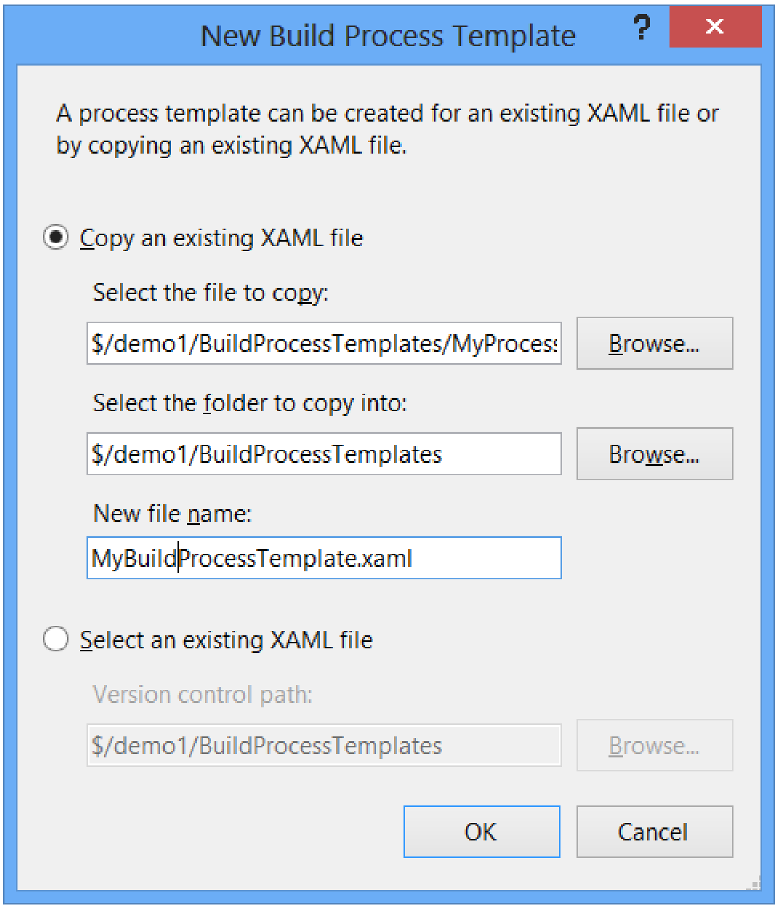
Build Process Template

If you use a custom build process template in a build definition and you try to import it to a team project where your custom build process is not available, you will get following message:



The process template '$/demo2/BuildProcessTemplates/MyProcessTemplate.xaml' does not exist in current tfs project. Please add it and try again.

To add your custom build process template to the team project click on "Edit Build Definition…". Select the "Process" tab. In "Build Process Template" click "Show Details" and the click on "New" button. The following dialog will be shown:



New Build Process Template

A process template can be created for an existing XAML file or by copying an existing XAML file.

● Copy an existing XAML file

Select the file to copy: $/demo1/BuildProcessTemplates/MyProcess

Select the folder to copy into: $/demo1/BuildProcessTemplates

New file name: MyBuildProcessTemplate.xaml

○ Select an existing XAML file

Version control path: $/demo1/BuildProcessTemplates

Provide a name for the new build process template or select an existing XAML and click "OK".

Retry to import the build definition.

Enhanced Scenarios

Rollback of Build Definition Changes

If you want to rollback changes you can proceed as followed:

- Open source control at location where build definitions are stored ($/<project>/BuildProcessTemplates/BuildDefinitions)
- Rollback the changes on the build definitions file which has the same name as the build definition you want to rollback

- Check in your changes
- Right click your build definition and choose “Update from Source Control”

Your build definition is now rolled back.

Branch Strategies

Branches for build definitions can be used in more scenarios like you normally use for branching your source code. In this section common scenarios are explained.

Branch together with source branches

If you branch your source code, you might want also to branch your build definitions. Doing so, you can merge your build definition together with your source. If you, for example, add a new solution to your build, you merge your source and after that you can also merge the corresponding build definition. After merging your build definition to your target branch, the only thing to do is to adjust the source folder where the merged solution comes from.

Branch special builds

Often seen are special build definitions for Continuous Integration or Build definitions only build debug or release. These build definitions often only differ in configuration stuff, not in the solutions or files to build. You can reduce the administration of these special build if you created branches connecting them. If a new solution or some other changes are done for a build, which you want to apply to branched build, you can merge between them easily.

Migration of existing Build Definitions

After installing xeam Build Definition Extension changes made to all build definitions are tracked in source control. History of build definitions are tracked after installing the extension. Merging between build definitions which has not been created using xeam Build Definition Extension Branch feature is not able to merge changes.

If you want to be able to merge existing Build Definitions which have been created before installing xeam Build Definition Extension, please contact us and we provide a service for you to setup this. Note: This service is separately charged and not included in the license of xeam Build Definition Extension.

Delete Build Definition

When a build definition is deleted in Build Explorer, the Build Definition file in Source Control has to be deleted manually. Please make sure, that you delete the file in Source Control after deleting the build definition, otherwise it cannot be recreated during build definition branch creation, if you create a build definition that has existed previously.

Migration of Build Definitions to another team project

To migrate your build definitions to another team project use the import/export functionality. Export to files the build definitions you want to migrate. Connect to the team project where you want to migrate them. Import the exported files.

Migration of Build Definitions to another TFS

To migrate your build definitions to another TFS use the import/export functionality. Export to files the build definitions you want to migrate. Connect to the TFS where you want to migrate them. Connect to the desired team project. Import the exported files.

Feedback

You are welcome to send us your feedback, suggestion or report issues on the following email address: **support@xeam-solutions.com**