

TFSVersioning Version 2.0 Update Guide

TFS 2012 Compatibility

Mark Nichols
January 22, 2013

© 2013 Mark Nichols

This document is part of the TfsVersioning project (CodePlex) and is subject to the Ms-PL Open Source License

Contents

Summary:	3
Files	3
Installation	3
Compatibility	3
Assembly Attribute Replacement	4
Replacement Patterns.....	5
Example of Combining Patterns and Text.....	5
Static Values and Dynamic Attribute Value Replacement.....	6
Examples	7
Example 1: Replacement Patterns in the Build Definition.....	7
Result:	7
Example 2: Replacement Patterns in the Version Seed File	9
Version Seed File.....	9
Result:	10

Summary:

The primary change with version 2.0 of TfsVersioning is that it has been built to operate within the Microsoft Team Foundation Server 2012 build process.

This document also describes the additional assembly attribute update capabilities of TfsVersioning version 2.0. This document only describes the additions made to the product which includes control over updating several assembly attributes (described in detail below). For information about TfsVersioning that is not described here, please review the “**TfsVersioning User and Development Guide**”.

This addition to TfsVersioning will allow you to generate descriptive dynamic (build-time) values in your assemblies. It will also allow you to insert common values across all assemblies in a build and manage them either in a common file (version seed file) or directly through a build definition.

Files

TfsBuild.Versioning.Activities.dll: This is the custom build activity and is a direct replacement if you are currently using a previous version.

VersioningBuildTemplate20.xaml: This build template provides an updated interface in the build definition to add the replacement patterns for an extended list of assembly attributes (detailed and defined below). This file is an addition to any existing build templates that may be in place.

Installation

Installation of version 2.0 is the same as the previous version. For more information see the “**TfsVersioning User and Development Guide**”.

Compatibility

The TfsVersioning 2.0 custom build activity and the associated build templates (workflows) have been created to work within the automated build process of Team Foundation Server 2012.

Assembly Attribute Replacement

The list below describes the various attributes that can be controlled through the TfsVersioning custom activity.

Attribute	Replaces
Assembly Title Pattern	Assembly Title Attribute: String value specifying a friendly name for the assembly. For example, an assembly named comdlg might have the title Microsoft Common Dialog Control.
Assembly Description Pattern	Assembly Description Attribute: String value specifying a short description that summarizes the nature and purpose of the assembly.
Assembly Configuration Pattern	Assembly Configuration Attribute: String value indicating the configuration of the assembly, such as Retail or Debug. The runtime does not use this value.
Assembly Company Pattern	Assembly Company Attribute: String value specifying a company name.
Assembly Product Pattern	Assembly Product Attribute: String value specifying product information.
Assembly Copyright Pattern	Assembly Copyright Attribute: String value specifying copyright information.
Assembly Trademark Pattern	Assembly Trademark Attribute: String value specifying trademark information.
Assembly Culture Pattern	Assembly Culture Attribute: Enumerated field indicating the culture that the assembly supports. An assembly can also specify culture independence, indicating that it contains the resources for the default culture.
Assembly Informational Version Pattern	Assembly Informational Version Attribute: String value specifying version information that is not used by the common language runtime, such as a full product version number.

Replacement Patterns

The patterns described below allow you to designate values that are taken from the system at build time and define build-specific, detailed attributes and attributes that are managed in a much more controlled fashion. The patterns below can be combined with text to create more complex and informative descriptions within an application's assembly attributes.

They can be used anywhere within the attribute patterns but, in all cases; the values within the attributes must abide by the rules set forth within the .NET environment. For example, "Assembly Culture Pattern" values must abide by the possible enumerated values within .NET or compile-time errors may result.

(In all cases below, the examples assume the date is October 5, 2011 - 12:57:00 PM - GMT -05:00)

Pattern	Replaced Value
\$TPROJ	TFS Team Project Name
\$REQBY	Build Requested By ID – Example: "domain:userid"
\$BNAME	Build Definition Name – Example: "BuildVersioning - Dev - v1.5"
\$UTIME	Universal Time – Example: "10/5/2011 5:57:00 PM"
\$LDATE	Long Date – Example: "Wednesday, October 05, 2011"
\$LTIME	Long Time – Example: "12:57:00 PM"
\$SSDATE	Short Date – Example: "10/5/2011"
\$STIME	Short Time – Example: "12:57 PM"
\$BNUM	Build Number (This is the full build number as defined by the "Build Number Format" property in the build definition.) Example: "BuildVersioning - Dev - v1.5_20111005.27"
\$YYYY	Full year value. – Example: "2011"
\$YY	2 digit year value – Example: "11"
\$MM or \$M	1 or 2 digit month of the year – Example: "10"
\$DD or \$D	1 or 2 digit day of the month – Example: "5"
\$J	Julian Date - YYDDD – Example: "11278"
\$B	Build Number of the Day (This is typically the last number value in the "Build Number Format". Example: Looking at the value above in \$BNUM, the build number is "27"

Example of Combining Patterns and Text

You may want to include specific build information within your assembly (in addition to the numeric version info) to help you during debugging and deployment. Information about how and when the code was built will definitely help as you make decisions around what has been tested and what can be deployed. For example: TFS labels your code as part of every build. You may decide that it would be beneficial to include the label information in an assembly attribute along with the universal time that it was built AND who initiated the build.

To satisfy these needs, the Assembly Informational Version Pattern could be set to:

“\$BNUM on \$UTIME by \$REQBY”. This combination of replacement patterns and text will create a descriptive notation contained in the assembly that can be read by anyone just by looking at the file properties in Windows Explorer. More description on this capability is detailed below.

Static Values and Dynamic Attribute Value Replacement

Static values are those values set directly within the AssemblyInfo.* files. Dynamic values are those set within the build definition or version seed file – both can contain replacement values that are generated during the build based.

If you would like to utilize static values within the AssemblyInfo file and not have them modified during the build then leave those attribute values blank in the build definition or don't include them in the version seed file. If the TfsVersioning activity does not see a replacement value for an attribute, it will not attempt to replace that value in the AssemblyInfo file. So, you only need to include the attribute values that you want to change.

Examples

Example 1: Replacement Patterns in the Build Definition

Below is an example of a build definition that modifies the AssemblyVersion, AssemblyFileVersion and AssemblyInformationalVersion attributes. Note: the following build definitions utilize the updated build assembly and the updated build template “VersioningBuildTemplate20.xaml”.

Build process parameters:	
1. Required	
Items to Build	Build \$/BuildActivities/SolutionBuildVersioning/Dev/Version 1.5.0.0/Build
2. Basic	
Automated Tests	Run tests in assemblies matching ***test*.dll using settings from \$/BuildA
Build Number Format	\$(BuildDefinitionName)_\$(Date:yyyyMMdd)\$(Rev:.r)
Clean Workspace	All
Logging Verbosity	Normal
Perform Code Analysis	AsConfigured
Source And Symbol Server Settings	Index Sources
3. Advanced	
4. Build Versioning	
Assembly File Version Pattern	1.5.J.B
Assembly Version Pattern	1.5.0.0
AssemblyInfo File Pattern	AssemblyInfo.*
Build Number Prefix	0
Force Create Version	True
Perform Check-in of the AssemblyInfo Files	False
Use Version Seed File	False
Version Seed File Path	TfsVersion\VersionSeed.xml
5. Build Versioning (Additional)	
Assembly Company Pattern	
Assembly Configuration Pattern	
Assembly Copyright Pattern	
Assembly Culture Pattern	
Assembly Description Pattern	
Assembly Informational Version Pattern	Team Project: \$TPROJ - Build Name: \$BNAME - Requested by: \$REQBY
Assembly Product Pattern	
Assembly Title Pattern	
Assembly Trademark Pattern	

Result:

- “Use Version Seed File” is set to False so the patterns for modifying the assembly attribute properties are taken from the build definition.
- The “Assembly File Version Pattern” is 1.5.J.B which appears (in Windows file properties for the current date) as 1.5.11194.2.
- “Assembly Informational Version Pattern” is “Team Project: \$TPROJ - Build Name: \$BNAME - Requested by: \$REQBY” which will display the:
 - Team Project name

- Build Name
- Requested by ID
- The rest of the assembly attribute properties are either taken from the “AssemblyInfo” file or general defaults.

A screen shot of the file “Properties/Details” dialog box from Windows Explorer.

Property	Value
Description	
File description	TfsBuild.Versioning.Activities
Type	Application extension
File version	1.5.11194.2
Product name	TfsBuild.Versioning.Activities
Product version	Team Project: BuildActivities - Build Name: Build
Copyright	Copyright © Microsoft 2011
Size	189 KB
Date modified	7/13/2011 9:45 PM
Language	Language Neutral
Original filename	TfsBuild.Versioning.Activities.dll

Example 2: Replacement Patterns in the Version Seed File

Below is an example of a build definition that modifies the AssemblyVersion, AssemblyFileVersion and AssemblyInformationalVersion attributes.

Build process parameters:	
1. Required	
Items to Build	Build \$/BuildActivities/SolutionBuildVersioning/Dev/Version 1.5.0.0/Build
2. Basic	
Automated Tests	Run tests in assemblies matching ***test*.dll using settings from \$/BuildA
Build Number Format	\$(BuildDefinitionName)_(Date:yyyyMMdd)\$(Rev:.r)
Clean Workspace	All
Logging Verbosity	Normal
Perform Code Analysis	AsConfigured
Source And Symbol Server Settings	Index Sources
3. Advanced	
4. Build Versioning	
Assembly File Version Pattern	1.5.J.B
Assembly Version Pattern	1.5.0.0
AssemblyInfo File Pattern	AssemblyInfo.*
Build Number Prefix	0
Force Create Version	True
Perform Check-in of the AssemblyInfo Files	False
Use Version Seed File	True
Version Seed File Path	TfsVersion\VersionSeed.xml
5. Build Versioning (Additional)	
Assembly Company Pattern	
Assembly Configuration Pattern	
Assembly Copyright Pattern	
Assembly Culture Pattern	
Assembly Description Pattern	
Assembly Informational Version Pattern	Team Project: \$TPROJ Build Name: \$BNAME Requested by: \$REQBY
Assembly Product Pattern	
Assembly Title Pattern	
Assembly Trademark Pattern	

Version Seed File

```
<VersionSeed>
  <Solution name="BuildVersioning">
    <AssemblyVersionPattern>1.5.0.0</AssemblyVersionPattern>
    <AssemblyFileVersionPattern>1.5.j.b</AssemblyFileVersionPattern>
    <AssemblyTitlePattern>TfsVersioning Custom Build Activity built on $SDATE</AssemblyTitlePattern>
    <AssemblyInformationalVersionPattern>BuildName: $BNAME - Requested by: $REQBY</AssemblyInformationalVersionPattern>
  </Solution>
  <Solution name="Default">
    <AssemblyVersionPattern>1.0.2.0</AssemblyVersionPattern>
    <AssemblyFileVersionPattern>1.0.j.b</AssemblyFileVersionPattern>
  </Solution>
</VersionSeed>
```

As you can see above, the only values (outside of the version numbers) that will be modified in the AssemblyInfo file(s) are the AssemblyTitle and AssemblyInformationalVersion attributes.

Result:

- “Use Version Seed File” is set to True so the patterns for modifying the assembly attribute properties are taken from the “version seed” (XML) file.
- The “Assembly File Version Pattern” is 1.5.J.B which appears (in Windows 7 file properties for the current date) as 1.5.11194.4.
- “Assembly Informational Version Pattern” is “BuildName: \$BNAME - Requested by: \$REQBY” which will display the:
 - Build Name
 - Requested by ID
- Even though the “Assembly Informational Version Pattern” is set to a value in the build definition, it is not used because the “Use Version Seed File” property is set to True.
- The Assembly Title is set to: “TfsVersioning Custom Build Activity built on \$SDATE”
- The rest of the assembly attribute properties are either taken from the “AssemblyInfo” file or general defaults.

A screen shot of the file “Properties/Details” dialog box from Windows Explorer.

Property	Value
Description	
File description	TfsVersioning Custom Build Activity built on 7/13/2011
Type	Application extension
File version	1.5.11194.4
Product name	TfsBuild.Versioning.Activities
Product version	BuildName: BuildVersioning - Dev - v1.5 - Requested by: NORTHAMERICA:marknic
Copyright	Copyright © Microsoft 2011
Size	189 KB
Date modified	7/13/2011 11:52 PM
Language	Language Neutral
Original filename	TfsBuild.Versioning.Activities.dll