Microsoft® Visual Studio 2005

Microsoft Visual Studio Team & MSBuild DeskSheet

Team Build Target Exec Order

***: Customisable Build Targets**

1. **BeforeEndToEndIteration***
2. **BuildNumberOverrideTarget***
3. InitializeEndToEndIteration
4. **BeforeClean***
5. CoreClean
6. **AfterClean***
7. Clean
8. InitializeBuild
9. **BeforeGet***
10. **BeforeLabel***
11. Label
12. **AfterLabel***
13. InitializeWorkspace
14. CoreGet
15. **AfterGet***
16. PreBuild
17. **BeforeCompile***
18. CoreCompile
19. **AfterCompile***
20. Compile
21. GetChangeSetsAndUpdateWorkItems
22. PostBuild
23. **BeforeTest***
24. CoreTest
25. Test
26. **AfterTest***
27. PackageBinaries
28. TeamBuild
29. **BeforeDropBuild***
30. CoreDropBuild
31. CopyLogFiles
32. **AfterDropBuild***
33. DropBuild
34. EndToEndIteration
35. **AfterEndToEndIteration***
36. ***BeforeOnBuildBreak ****

MSBuild Conditional Constructs

```
<Choose>
   <When
Condition="'StringA'=='StringB'">...
</When>
   <Otherwise>... </Otherwise>
</Choose>
```

Default Tasks in Microsoft.TeamFoundation.Build.targets

The following tasks can be customised by modifying their properties

Team Build Task	Associated target
Get – synchronizes the sources on the build server. Also raises log events to inform the logger of every file and version retrieved into the local workspace	CoreGet
Label – labels all source files of a specified workspace	CoreLabel
GenCheckinNotesUpdateWorkItems – generates the list of all changesets and work items created or affected during a build. Labels are used to tag a particular build	**GetChangeSetsAndUpdateWorkitems** target is invoked if the build is succesful. **GetChangeSetsOnBuildBreak** target is invoked if the build is not succesful.
CreateNewWorkItem – creates new work item for failures during build process. The Found In field in the work item is updated with a link to the specific build number	CreateWorkItem
CreateWorkspaceTask – creates new workspace using workspacemapping.xml	InitializeWorkspace
DeleteWorkspaceTask – deletes workspace created during the build. The actual files are not deleted from build server. If specified workspace does not exist, DeleteWorkspaceTask issues a warning	InitializeWorkspace
UpdateBuildNumberDropLocation – creates the drop folder at drop site and updates the build number	InitializeEndToEndIteration
TeamBuildMessage – extension of MSBuild Message Task	AfterCompile
TestToolsTask – runs a set of tests during the build and publishes the results	RunTestWithConfiguration

MSBuild Reserved Properties

1. **MSBuildProjectDirectory** – The absolute path of the directory where the project file is located, for example, C:\MyCompany\MyProduct.
2. **MSBuildProjectFile** – The complete file name of the project file, including the file name extension, for example, MyApp.proj.
3. **MSBuildProjectExtension** – The file name extension of the project file, including the period, for example, .proj.
4. **MSBuildProjectFullPath** – The absolute path and complete file name of the project file, for example, C:\MyCompany\MyProduct\MyApp.proj.
5. **MSBuildProjectName** – The file name of the project file without the file name extension, for example, MyApp.
6. **MSBuildBinPath** – The absolute path of the directory where the MSBuild binaries that are currently being used are located, for example, C:\Windows\Microsoft.Net\Framework\v2.0. This property is useful if you need to refer to files in the MSBuild directory.
7. **MSBuildProjectDefaultTargets** – The complete list of targets specified in the DefaultTargets attribute of the Project element. For example, the following Project element would have an MSBuildDefaultTargets property value of A;B;C. <Project DefaultTargets="A;B;C" >
8. **MSBuildExtensionsPath** – The MSBuild folder under the Program Files directory. This location is a useful place to put custom target files. For example, your targets files could be installed at \Program Files\MSBuild\MyFiles\Northwind.targets and then imported in project files with the following XML. <Import Project="$(MSBuildExtensionsPath)\MyFiles\Northwind.targets"/>

Customisable TFS Build Properties

1. SkipClean
2. SkipGet
3. SkipLabel
4. SkipInitializeWorkspace
5. SkipPostBuild
6. SkipDropBuild
7. SkipWorkItemCreation – skip work item creation on build failure
8. AdditionalVCOverrides – Property sheets passed to the VC++ solutions
9. SolutionRoot – Specifies the root of the sources folder

MSBuild Transforms

```
<ItemGroup>
   <Schema Include="sub1\**\*.xsd"/>
</ItemGroup>
<Target Name="Messages">
   <Message Text="rootdir: @(schema->'%(rootdir)')"/>
</Target>
```

MSBuild well-known Item Metadata

Using: <ItemGroup><a Include="**Src\Prog.cs**"/>...

%(FullPath)	C:\MyProject\Src\Prog.cs
%(RootDir)	C:\
%(Filename)	Prog
%(Extension)	.cs
%(RelativeDir)	Src\
%(Directory)	MyProject\Src\
%(RecursiveDir)	If Include attribute contains the wildcard **, this metadata specifies the path to the file, beginning at the location of the wildcard. In the following example the item would contain a RecursiveDir value of *MyProject\Src\*. ***<ItemGroup><a Include="C:\**\Prog.cs" /></ItemGroup>***
%(Identity)	Src\Prog.cs
%(ModifiedTime), %(CreatedTime), %(AccessedTime)	2004-07-01 00:21:31.5073316

Version 1.0 - Produced by http://**FreeToDev**.spaces.live.com