Release Notes NuPattern 1.3.20.0

Note: These release notes are shipped within this update to NuPattern, and are also made available online at the NuPattern project site: http://nupattern.codeplex.com/releases/view/94069

Summary

This is a major release of NuPattern (formerly known as VSPAT). It is a compatibility release for Visual Studio 2012. Furthermore, this release introduces two separate versions of NuPattern that are released to the Visual Studio Gallery: **NuPattern Toolkit Builder VS2010**, and **NuPattern Toolkit Builder VS2012**.

We strongly recommend migrating existing pattern toolkits to this version using the instructions below, as several critical changes have been made to transition of the NuPattern project.

IMPORTANT NOTICE: Support for previous versions of VSPAT will be delivered as updates to new versions of NuPattern.
Any existing pattern toolkits built with a previous version of VSPAT (1.2.19.0) are not forward compatible with this version of NuPattern (1.3.20.0). Existing pattern toolkits that are not migrated may not work correctly when this version of NuPattern is installed in Visual Studio, and may not work with other pattern toolkits that are built with or migrated to this version of NuPattern.

To upgrade to this release of NuPattern, existing Pattern Toolkit projects are required to be manually migrated for running in either Visual Studio 2010, or Visual Studio 2012. See the Migration Notes in this document.

Please refer to the Installation Notes section for correctly installing this version of NuPattern.

This document includes the sections 'Who Needs to Migrate?' and 'Migration FAQ' that should address any questions you may have about compatibility and migration of existing toolkits.

As always, any questions that are not answered in this document should be posted to the 'Discussions List' at the project site: http://nupattern.codeplex.com.

Document Contents

What's New In this Release?

Note: To see more details for What's New in this release, see the notes at http://nupattern.codeplex.com/releases/view/94069

- This version of NuPattern fully supports both Visual Studio 2012, and Visual Studio 2010. There will be two separate install packages for each version of Visual Studio. They may both be installed side-by-side if both versions of Visual Studio are installed side-by-side.
- The NuPattern project ownership, identity and roadmap have been changed, and this has led to necessary changes in the identity and copyright changes in the installed packages. The license terms have not changed.
- The look and feel of this release has been updated to support the new themed support in Visual Studio 2012.
- Numerous changes have been made to support Visual Studio 2012, and as such pattern toolkits that were built with a previous version of VSPAT (version 1.2.19.0 or earlier) are required to be migrated and rebuilt with this version of NuPattern (1.3.20.0) in order to work with any version of NuPattern (1.3.20.0 or later).
- Several minor bug fixes have been addressed in this release.
- The dependency of the 'Feature Builder Power Tool' extension has now been removed, and NuPattern now embeds elements of the 'Feature Extension Runtime' extension which was previously installed as a separate extension into Visual Studio 2010. This permits future enhancements to the Guidance Explorer and Guidance Browser windows in Visual Studio.

Note: Independent of NuPattern, the 'Feature Extension Runtime' extension is shipped and installed by installing the 'Feature Builder Power Tool' extension, or by installing any 'Feature Extension' that was built by the 'Feature Builder Power Tool' extension. The 'Feature Extension Runtime' was also installed by previous versions of NuPattern (a.k.a. VSPAT) in Visual Studio 2010.
According to Microsoft, the 'Feature Extension Runtime' and the 'Feature Builder Power Tool' are no longer supported, and no version will be shipped for Visual Studio 2012.

IMPORTANT: We strongly recommended that you install the new version of NuPattern, and uninstall or disable the 'Feature Builder Power Tool' and 'Feature Extension Runtime' extensions when upgrading to this version of NuPattern, or when installing any toolkit built with this version of NuPattern.

Installation Notes

This version of NuPattern (1.3.20.0) is not backwardly compatible with any version of VSPAT (1.2.19.0 or earlier).

NuPattern is also not compatible with any version of the 'Feature Builder Power Tool' extension, or the 'Feature Extension Runtime' extension.

For Visual Studio 2010: before installing this version of NuPattern in Visual Studio 2010, you must uninstall any version of the following extensions if they are already installed. Use the 'Extension Manager' window in Visual Studio to uninstall (or disable) these extensions, and restart Visual Studio:

- 'Feature Builder Power Tool' extension (1.0.1 or earlier)
- 'Feature Extension Runtime' extension (1.0.1 or earlier)
- 'Pattern Toolkit Builder' extension (1.2.19.0 or earlier)
- 'Pattern Toolkit Library Support' extension (1.2.19.0 or earlier)
- 'Pattern Toolkit Manager' extension (1.2.19.0 or earlier)
- Any other extension that is dependent on the 'Feature Extension Runtime' extension, or 'Pattern Toolkit Builder' extension.

Note: Visual Studio 2012 users can ignore this step

Who Needs to Migrate?

Please use the following notes as guidance for determining whether or not to update existing pattern toolkits built with previous versions of VSPAT (1.2.19.0 or earlier).

Note: New pattern toolkits created with this version of NuPattern will NOT require any migration, but will only target the version of Visual Studio with which they were built.

You will be required to migrate an existing pattern toolkit in either of the following scenarios:

1. As an author/creator of a pattern toolkit built against VSPAT (1.2.19.0 or earlier), you wish your pattern toolkit to work with NuPattern (1.3.20.0 or later) in either Visual Studio 2010, or in Visual Studio 2012.
    - To upgrade your pattern toolkit you will be required to:
        i. Uninstall the older versions of VSPAT and related 'Feature Builder Power Tools' extensions. See Installation Notes above.
        ii. Install the latest version of 'NuPattern Toolkit Builder' extension.
        iii. Migrate the pattern toolkit to the latest version of NuPattern (using the Migration Notes at the end of this document)
2. As a user of a pattern toolkit built against VSPAT (1.2.19.0 or earlier), you wish your pattern toolkit to work in Visual Studio 2012, or with NuPattern (1.3.20.0 or later) in Visual Studio 2010.
    - To upgrade your pattern toolkit you will be required to:
        i. Obtain a newer version of your pattern toolkit from the original author/creator of the pattern toolkit.
        ii. Uninstall your existing pattern toolkit.
        iii. Uninstall the older versions of VSPAT and related 'Feature Builder Power Tools' extensions. See Installation Notes above.
        iv. Manually install the latest version of your pattern toolkit, provided by the author/creator of the pattern toolkit.

Note: Neither of these upgrade scenarios will occur automatically for an existing pattern toolkit.

You will NOT be required to migrate an existing pattern toolkit in these scenarios:

1. As an author/creator of a pattern toolkit built against VSPAT (1.2.19.0 or earlier), you wish your pattern toolkit to continue work in Visual Studio 2010, without updating to NuPattern (1.3.20.0).
2. As a user of a pattern toolkit built against VSPAT (1.2.19.0 or earlier), you wish your pattern toolkit to continue to work in Visual Studio 2010, without updating to NuPattern (1.3.20.0).

Important: Pattern toolkits that were built with previous versions of VSPAT (1.2.19.0 or earlier) will continue to install and work in Visual Studio 2010, and will not require immediate migration, provided that: (1) The user of the pattern toolkit does not install another pattern toolkit that was built with a newer version of NuPattern, and (2) provided they do not install NuPattern (1.3.20.0 or later).

In projects where the development tools are strictly controlled this will likely not be an immediate issue. However, for development environments that are not strictly controlled, where users of Visual Studio are permitted to download and update their own extensions to Visual Studio, this will ultimately raise a compatibility issue.

It is therefore strongly recommended that all pattern toolkits built with previous versions of VSPAT (1.2.19.0 or earlier) are eventually migrated to NuPattern (1.3.20.0 or later) to avoid future backward compatibility issues.

Supported Migration Scenarios

There are 2 main migration scenarios for migrating existing toolkits to target Visual Studio 2010 or Visual Studio 2012.

These scenarios are documented in this release document and supported by this version of NuPattern. Other migration scenarios may exist, but the reader will need to determine the detailed steps to execute them.

1. Migrate an existing pattern toolkit to run in Visual Studio 2010
    - In this scenario you wish to migrate an existing pattern toolkit built with VSPAT (1.2.19.0 or earlier) in Visual Studio 2010, for use in Visual Studio 2010.
        - In this scenario, your built pattern toolkit will be installed into Visual Studio 2010.
        - In this scenario, your built pattern toolkit will never be installed into Visual Studio 2012.
2. Migrate an existing pattern toolkit to run in Visual Studio 2012
    - In this scenario you wish to migrate an existing pattern built with VSPAT (1.2.19.0 or earlier) in Visual Studio 2010, for use in Visual Studio 2012.
        - In this scenario, your built pattern toolkit will be installed into Visual Studio 2012.
        - In this scenario, your built pattern toolkit will never be installed into Visual Studio 2010.

Note: Previous versions of VSPAT (1.2.19.0 or earlier) supported building pattern toolkits that targeted only Visual Studio 2010. In the new version of NuPattern (1.3.20.0) pattern toolkits can be built and targeted at either Visual Studio 2010 or Visual Studio 2012.
Note: Newer versions of NuPattern in the future may support the building and targeting of pattern toolkits for both Visual Studio 2010 and Visual Studio 2012, but in this release this is currently not a documented scenario. Keep up to date with this scenario at the project site: http://nupattern.codeplex.com

Migration FAQ

Do existing pattern toolkits that were built with a previous version of VSPAT(1.2.19.0 or earlier) require migration?

Eventually yes, it is strongly recommended, see the previous section on Who Needs to Migrate? for more details about this issue, and then follow the Migration Notes section.

What happens to existing pattern toolkits built with a previous version of VSPAT (1.2.19.0 or earlier)?

For Visual Studio 2010: Existing pattern toolkits will continue to install and work in Visual Studio 2010 as before, provided a new pattern toolkit built with NuPattern (1.3.20.0 or later) is not installed into Visual Studio. See Who Needs to Migrate? for more details about this issue, and why migration to NuPattern (1.3.20.0 or later) for all pattern toolkits is still strongly recommended.

For Visual Studio 2012: Does not apply as no previous version of VSPAT (1.2.19.0 or earlier) supported building pattern toolkits for Visual Studio 2012 installation.

Will NuPattern update from the Visual Studio Gallery?

For Visual Studio 2010: Yes, authors of existing pattern toolkits (or those who have already installed NuPattern) can be notified of a new version of NuPattern from the Visual Studio Gallery, and they can choose to install the new version. At which point, they will need to migrate their existing pattern toolkit projects.

Note: Users of existing pattern toolkits will not be notified of updates from the Visual Studio Gallery.

For Visual Studio 2012: No, there was no previous version of NuPattern for Visual Studio 2012 to update.

Can I target my pattern toolkit to run in Visual Studio 2012?

For Visual Studio 2010: Yes. Pattern toolkits built with Visual Studio 2010 will not run properly in Visual Studio 2012 because these toolkits will have dependencies on Visual Studio 2010 that are not present in Visual Studio 2012. If you want your pattern toolkit to run in Visual Studio 2012, you must build (or migrate) your pattern toolkit to target Visual Studio 2012. See the Migration Notes section.

For Visual Studio 2012: Yes. All pattern toolkits built in Visual Studio 2012 run correctly in Visual Studio 2012. However, these toolkits will not install or work correctly in Visual Studio 2010.

How do I build my pattern toolkit to target both VS2010 and VS2012?

This is currently not supported in this version of NuPattern. Multi-targeting pattern toolkits is limited to general issues in multi-targeting any VSIX extension to run in multiple versions of Visual Studio. A pattern toolkit is just another VSIX extension, and as such has build-dependencies, and sub-dependencies that are not necessarily available in both versions of Visual Studio. Whilst multi-targeting a pattern toolkit may be technically possible, it is not currently supported in this version of NuPattern.

Note: The NuPattern project is investigating future approaches and techniques that may help pattern toolkit builders multi-target their toolkit to either Visual Studio 2010 or to Visual Studio 2012 or both, using a single codebase for the pattern toolkit. Keep up to date with progress on this issue at the project site: http://nupattern.codeplex.com

Technically, why do we need to migrate pattern toolkits, what has changed in this version?

This new version of NuPattern has had to undergo numerous major changes in its binary compatibility, in its dependencies, and in its ownership.

This version of NuPattern now delivers a new version for Visual Studio 2012, which brings changes in how pattern toolkits (VSIX extensions) are built and deployed. Generally speaking, a Visual Studio eXtension (VSIX) that is built with Visual Studio 2010 is not compatible with Visual Studio 2012 without some changes to its binary dependencies and registration information. It is because of these general Visual Studio compatibility issues that two versions of the NuPattern extensions will need to be deployed, targeted separately at Visual Studio 2010 and at Visual Studio 2012.

In addition, a major dependency of NuPattern to date has been the 'Feature Extension Runtime' extension, which is a sub-component of the 'Feature Builder Power Tool' extension. Support for the 'Feature Builder Power Tool' extension from Microsoft has ended, and there will be no release of the Power Tool for Visual Studio 2012. Therefore, this dependency has necessarily been absorbed and packaged into this version of NuPattern.

In addition, the ownership of the NuPattern project has recently been transferred from Microsoft to 'The Outercurve Foundation' where it benefits from being a supported open source project by the community. This has necessarily had to change the registration and ownership and identification of the binary deliverables of NuPattern, and again these cannot be technically compatible with previous versions of VSPAT (1.2.19.0 or earlier). However, the channel which NuPattern is being released on (Visual studio Gallery and CodePlex) remains the same.

These changes have necessarily led to a new major version of NuPattern being released that is unfortunately not backwardly compatible with previous versions of VSPAT (1.2.19.0 or earlier), and not compatible with pattern toolkits built with a previous version of VSPAT (1.2.19.0 or earlier).

The long term goal of this version of NuPattern (1.3.20.0 and later) is to replace previous installations of VSPAT (1.2.19.0 or earlier) and to upgrade any versions of pattern toolkits built with previous versions of VSPAT (1.2.19.0 or earlier).

Once this difficult migration phase is complete, no such detailed migration will be required again.

As this version of NuPattern is not backwardly compatible with previous versions, any toolkits that were built with previous versions of VSPAT (1.2.19.0 or earlier) will no longer work correctly if installed

alongside toolkits that are built with this version of NuPattern. This is why we strongly recommend migration of any existing toolkits.

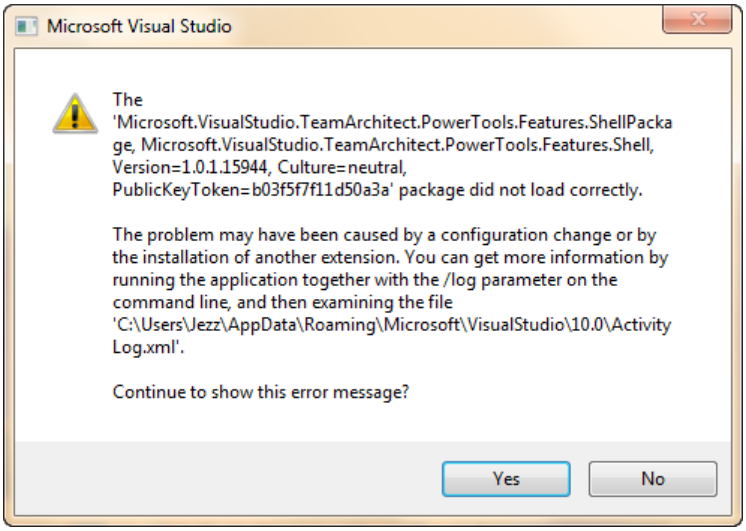
What happens to existing pattern toolkits that already embed the 'Pattern Toolkit Manager' from VSPAT 1.2.19.0?

For Visual Studio 2010: If a Visual Studio 2010 user already has a toolkit installed in Visual Studio, and then either installs NuPattern ('NuPattern Toolkit Builder' extension) or installs a pattern toolkit built with the new version of NuPattern in Visual Studio 2010, then the older version of the 'Pattern Toolkit Manger' will be upgraded automatically for them. Their toolkit should continue to work as before.

For Visual Studio 2012: Does not apply as no previous version of VSPAT (1.2.19.0 or earlier) supported building pattern toolkits for Visual Studio 2012 installation.

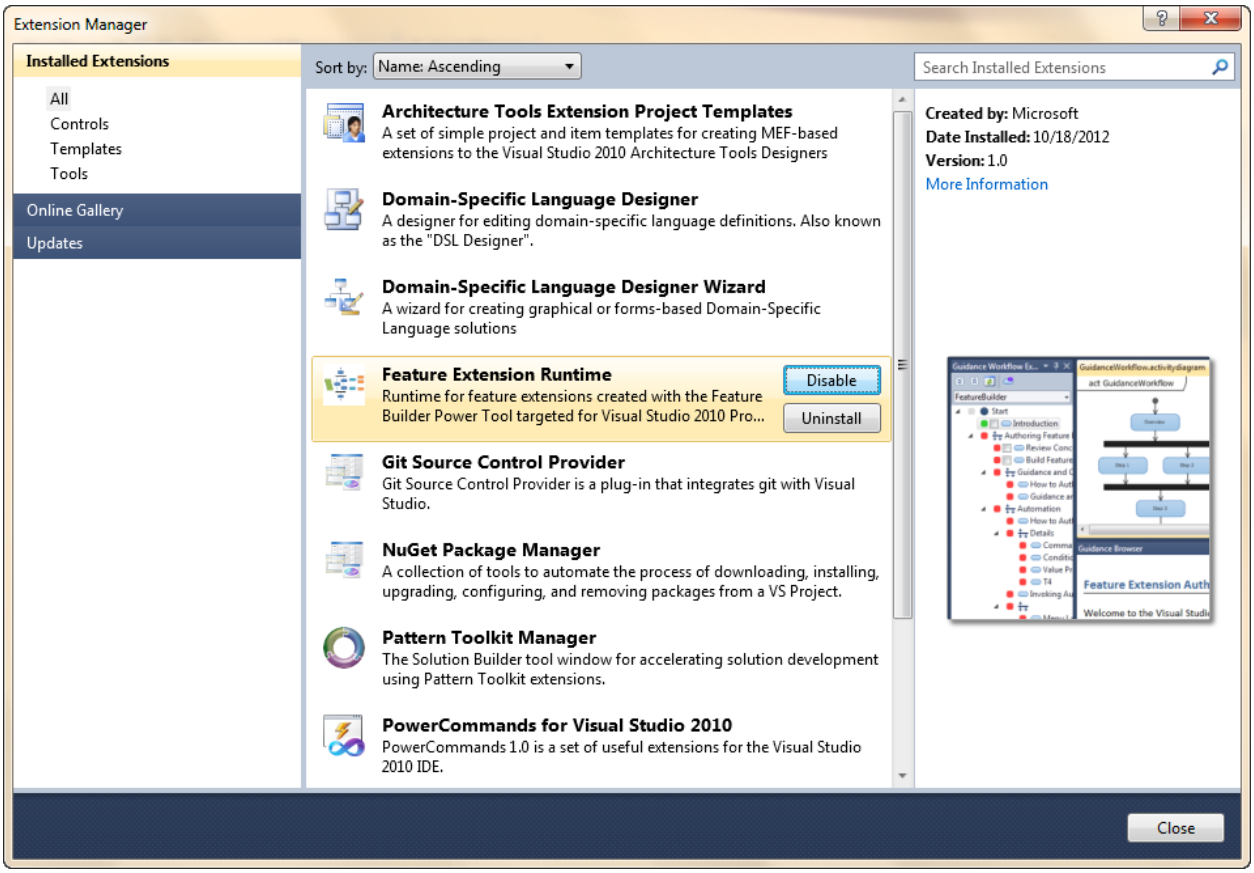
What happens if the 'Feature Extension Runtime' extension (or 'Feature Builder Power Tool' extension are already installed when I install this version of NuPattern?

For Visual Studio 2010: When Visual Studio 2010 starts, or while using Visual Studio, the following error is reported multiple times:



You must either [Disable] or [Uninstall] the 'Feature Extension Runtime' extension (and any and all extensions related to the 'Feature Builder Power Tool') in 'Extension Manager'.

Continued…



For Visual Studio 2012: Does not apply, as no previous version of the 'Feature Extension Runtime' or Feature Builder Power Tool' supported installation into Visual Studio 2012.

Migration Notes

These manual actions need to be followed in the stated order to migrate existing pattern toolkit projects to the current version of NuPattern (v.1.3.20.0), from VSPAT (**v.1.2.19.0** or earlier)**.**

Since the current version of NuPattern supports both Visual Studio 2010 and Visual Studio 2012, some migration steps may be different depending on which version of Visual Studio you wish to target your pattern toolkit to. The migration steps are tailored for each version of Visual Studio.

Note: In this version of NuPattern, you must migrate your existing pattern toolkit project and its solution to the version of Visual Studio you wish to target your pattern toolkit for.

Individual migration steps with be marked for the Visual Studio that the pattern toolkit will target:

- **VS2010** – Migration steps specific only to pattern toolkits targeted to Visual Studio 2010
- **VS2012** – Migration steps specific only to pattern toolkits targeted to Visual Studio 2012

Important: Before starting the migration , it you are strongly recommend to back up your source code, or use source control, to manage any changes you make in migrating your toolkits projects.

Pattern Toolkit Solutions:

Pre Work:

Note: Before opening your pattern toolkit project or solution in Visual Studio.

(Optional) If your toolkit solution contains one or more DSL projects (unusual for most pattern toolkit projects)	
VS2012	Recommend you use the DSL Tool Migration wizard to first convert your DSL projects. Found in: **%ProgramFiles(x86)%\Microsoft Visual Studio 11.0\VSSDK\VisualStudioIntegration\Tools\DSLTools\DslProjectsMigrationTool.exe**. Please consult the **MigrationGuide.mht** in the same folder.
(Optional) If your toolkit project file use the following targets files (unusual for most pattern toolkit projects)	
VS2012	Change: $(MSBuildExtensionsPath)\Microsoft\VisualStudio\TextTemplating\v10.0\Microsoft.TextTemplating.targets To: $(MSBuildExtensionsPath)\Microsoft\VisualStudio\v11.0\TextTemplating\Microsoft.TextTemplating.targets
Replace all occurrences of “Microsoft.VisualStudio.Patterning” in all files.	
VS2010, VS2012	Replace all occurrences of “Microsoft.VisualStudio.Patterning” with “NuPattern” in all files of all types in the solution.
Remove all occurrences of “using NuPattern.Extensibility;” in all code files.	
VS2010, VS2012	Remove all occurrences of “using NuPattern.Extensibility;” in all code files (or generated code file) in the solution.
Remove all occurrences of “using Microsoft.VisualStudio.TeamArchitect.PowerTools.*;” in all code files.	
VS2010, VS2012	Remove all occurrences of “using Microsoft.VisualStudio.TeamArchitect.PowerTools.*” in all code files (or generated code file) in the solution.
Rename all occurrences of “IFxr*” in all code files.	
VS2010, VS2012	Replace all occurrences of the interfaces that begin with IFxr* with I*

Solution file (.sln):*

Note: Before opening your pattern toolkit project or solution in Visual Studio.

Open in 'XML View':

Add/Modify the following section to the end of the file, before the last 'EndGlobal' line.	

VS2010, VS2012	`GlobalSection(ExtensibilityGlobals) = postSolution`
	`  GuidanceWorkflows = 9f6dc301-6f66-4d21-9f9c-b37412b162f6:Creating Pattern Toolkits:1.3.20.0`
	`EndGlobalSection`

Solution Builder File (.slnbldr):*

Note: Before opening your pattern toolkit project or solution in Visual Studio.

Open in 'XML View':

Modify any <product> elements where definitionName="PatternToolkit" to the following	
VS2010, VS2012	`<product definitionName="PatternToolkit" extensionId="9f6dc301-6f66-4d21-9f9c-b37412b162f6" extensionName="NuPattern Toolkit Builder" author="NuPattern" version="1.3.20.0" >`
Modify any <product> elements where definitionName="AutomationLibrary" to the following:	
VS2010, VS2012	`<product definitionName="PatternToolkitLibrary" extensionId="97bd7ab2-964b-43f1-8a08-be6db68b018b">`
Replace any <reference> elements where kind="NuPattern.Extensibility.References.SolutionArtifactLinkReference" to the following:	
VS2010, VS2012	`<reference kind="NuPattern.Runtime.References.SolutionArtifactLinkReference" />`
Replace any <reference> elements where kind="NuPattern.Extensibility.References.GuidanceReference" to the following:	
VS2010, VS2012	`<reference kind="NuPattern.Runtime.References.GuidanceReference" />`

Pattern Toolkit Project:

Toolkit Project File (csproj):

Note: Before opening your pattern toolkit project or solution in Visual Studio.

Open in 'XML View':

Add the following "PropertyGroup" as the first "PropertyGroup" in the file:	

VS2010	`<PropertyGroup>`
	`<MinimumVisualStudioVersion>10.0</MinimumVisualStudioVersion>`
	`<VisualStudioVersion Condition="'$(VisualStudioVersion)' == ''">10.0</VisualStudioVersion>`
	`<VSToolsPath Condition="'$(VSToolsPath)' == ''">$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v$(VisualStudioVersion)</VSToolsPath>`
	`</PropertyGroup>`

VS2012	`<PropertyGroup>`
	`<MinimumVisualStudioVersion>11.0</MinimumVisualStudioVersion>`
	`<VisualStudioVersion Condition="'$(VisualStudioVersion)' == ''">11.0</VisualStudioVersion>`
	`<VSToolsPath Condition="'$(VSToolsPath)' == ''">$(MSBuildExtensionsPath32)\Microsoft\VisualStudio\v$(VisualStudioVersion)</VSToolsPath>`
	`</PropertyGroup>`

Add the following "PropertyGroup" and "Import" immediately below the first "PropertyGroup" in the file:	

VS2012	`<PropertyGroup>`
	`<IncludeAssemblyInVSIXContainer>true</IncludeAssemblyInVSIXContainer>`
	`<IncludeDebugSymbolsInVSIXContainer>false</IncludeDebugSymbolsInVSIXContainer>`
	`<IncludeDebugSymbolsInLocalVSIXDeployment>true</IncludeDebugSymbolsInLocalVSIXDeployment>`
	`<CopyBuildOutputToOutputDirectory>false</CopyBuildOutputToOutputDirectory>`
	`<CopyOutputSymbolsToOutputDirectory>false</CopyOutputSymbolsToOutputDirectory>`
	`</PropertyGroup>`
	`<Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props" Condition="Exists('$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props')" />`

<table>
<tr><td colspan="2">Change all “Microsoft.VisualStudio.*.dll” version 10.0 assembly references to version 11.0 assembly references.</td></tr>
<tr><td>VS2012</td><td>

```
<Reference Include="Microsoft.VisualStudio.Shell, Version=11.0.0.0,
Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL"
/>
            <Reference Include="Microsoft.VisualStudio.ExtensibilityHosting,
Version=11.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a,
processorArchitecture=MSIL" />
```

</td></tr>
<tr><td colspan="2">Change all “Microsoft.VisualStudio.*.10.0.dll” assembly references to “Microsoft.VisualStudio.*.11.0.dll” assembly references. e.g.</td></tr>
<tr><td>VS2012</td><td>

```
<Reference Include="Microsoft.VisualStudio.Shell.Immutable.11.0,
Version=11.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a,
processorArchitecture=MSIL" />
```

Change all “Microsoft.VisualStudio.Patterning.*” assembly references to “NuPattern.*.dll” assembly references. e.g.

```
    <Reference Include="NuPattern.Runtime.Interfaces">
      <HintPath>$(PatternToolkitRuntime)\NuPattern.Runtime.Interf
aces.dll</HintPath>
      <Private>false</Private>
  </Reference>
```

</td></tr>
<tr><td colspan="2">Remove all “Microsoft.VisualStudio.TeamArchitect.PowerTools.*.dll” assembly references. e.g.</td></tr>
<tr><td>VS2010, VS2012</td><td>

```
<Reference Include="Microsoft.VisualStudio.TeamArchitect.PowerTools"/>
<Reference Include="Microsoft.VisualStudio.TeamArchitect.PowerTools.Features"/>
```

</td></tr>
<tr><td colspan="2">(Optional) Change the value of the “TargetFrameworkVersion” property:</td></tr>
<tr><td>VS2012</td><td>

```
<TargetFrameworkVersion>v4.5</TargetFrameworkVersion>
```

</td></tr>
<tr><td colspan="2">Change the path to the linked ‘PatternToolkitManager’ VSIX:</td></tr>
<tr><td>VS2010, VS2012</td><td>

```
    <Content Include="$(PatternToolkitBuilder)\NuPatternToolkitManager.vsix">
      <Visible>false</Visible>
      <Link>NuPatternToolkitManager.vsix</Link>
      <FixedLink>
      </FixedLink>
      <IncludeInVSIX>true</IncludeInVSIX>
    </Content>
```

</td></tr>
</table>

Ensure the correct properties are set for the project reference to the “Automation” project:	
VS2010, VS2012	<ItemGroup> <ProjectReference Include="..\<ToolkitProjectName>.Automation\<ToolkitProjectName>.Automation.csproj"> <Name><ToolkitProjectName>.Automation</Name> <IncludeOutputGroupsInVSIX>BuiltProjectOutputGroup</IncludeOutputGroupsInVSIX> <IncludeOutputGroupsInVSIXLocalOnly>DebugSymbolsProjectOutputGroup</IncludeOutputGroupsInVSIXLocalOnly> </ProjectReference> </ItemGroup>
Change the “Import” to the “Microsoft.VsSDK.targets”:	
VS2010, VS2012	<Import Project="$(VSToolsPath)\VSSDK\Microsoft.VsSDK.targets" />
Change the “Import” to the “Microsoft.VisualStudio.Patterning.Authoring.PatternToolkitVersion.targets”:	
VS2010, VS2012	<Import Project="$(localappdata)\Microsoft\MSBuild\NuPattern\NuPattern Toolkit Builder\NuPattern.Authoring.PatternToolkitVersion.targets" />
Replace the assembly reference to “NuPattern.Runtime.Interfaces”:	
VS2010, VS2012	<Reference Include="NuPattern.Runtime.Extensibility"> <HintPath>$(PatternToolkitRuntime)\NuPattern.Runtime.Extensibility.dll</HintPath> </Reference>
Replace the assembly reference to “NuPattern.Common.Presentation”:	
VS2010, VS2012	<Reference Include="NuPattern.Presentation"> <HintPath>$(PatternToolkitRuntime)\NuPattern.Presentation.dll</HintPath> </Reference>
(Optional) Replace the assembly reference to “NuPattern.Extensibility”:	
VS2010, VS2012	<Reference Include="NuPattern.Runtime.Extensibility"> <HintPath>$(PatternToolkitRuntime)\NuPattern.Runtime.Extensibility.dll</HintPath> </Reference>
Add the assembly reference to “NuPattern.Common”:	
VS2010, VS2012	<Reference Include="NuPattern.Common"> <HintPath>$(PatternToolkitRuntime)\NuPattern.Common.dll</HintPath> </Reference>
Replace the CatalogName attribute in the AssemblyInfo.cs file.	
VS2010, VS2012	[assembly: VsCatalogName(NuPattern.ComponentModel.Composition.Catalog.DefaultCatalogName)]

Source.extension.tt

(Optional) Remove all text template directives	
VS2010, VS2012	Remove all (@Template, @Assembly, @Import) at the start of the file. Leave the @Output and @Include directives.
Update the 'Include' directive to include 'source.extension.gen.ttinclude' instead of 'source.include.t4'	
VS2010, **VS2012**	`<#@ Include File="source.extension.gen.ttinclude" #>`
Update <SupportedProducts> and <SupportedFrameworkRuntimeEdition> elements:	

VS2010, VS2012	`<SupportedProducts>`
	`  <VisualStudio Version="<#=supportedVsVersion#>">`
	`  <Edition>Pro</Edition>`
	`  </VisualStudio>`
	`  </SupportedProducts>`
	`  <SupportedFrameworkRuntimeEdition MinVersion="<#=supportedFrxMinVersion #>" MaxVersion="<#=supportedFrxMaxVersion#>"/>`

Update <Reference> to 'Pattern Toolkit Manager' VSIX:	

VS2010, VS2012	`<Reference Id="<#=managerVsixIdentifier#>" MinVersion="<#=managerVsixVersion#>">`
	`  <Name><#=managerVsixName#></Name>`
	`  <VsixPath><#=managerVsixFileName#></VsixPath>`
	`</Reference>`

Update <MefComponent> elements	

VS2010, VS2012	From:
	`<MefComponent>|<#= productName #>|</MefComponent>`
	`<MefComponent>|<#= productName #>.Automation|</MefComponent>`
	To:
	`<MefComponent>|%CurrentProject%|</MefComponent>`
	`<MefComponent>|<#=libraryProjectName #>|</MefComponent>`

Update the <CustomExtension Type="PatternModel" …> element	
VS2010, VS2012	`<CustomExtension Type="NuPattern.Toolkit.PatternModel"`

All Project Template and Item Templates Files (*.vstemplate):

Change the Namespace, PublicKeyToken, and add the Version and Culture attributes to all <WizardExtension> elements for assemblies beginning with 'Microsoft.VisualStudio.Patterning'. e.g.	

VS2010, VS2012	`<WizardExtension>`
	`  <Assembly>NuPattern.VisualStudio.TemplateWizards, Version=1.0.0.0, Culture=neutral, PublicKeyToken=24c7786d4a8b1a88</Assembly>`
	`  <FullClassName>NuPattern.VisualStudio.TemplateWizards.InstantiationTemplateWizard</FullClassName>`
	`</WizardExtension>`

Change the <FullClassName> value for the <WizardExtension> elements for the 'ElementReplacementsWizard. e.g.	

VS2010, VS2012	`<WizardExtension>`
	`  <Assembly>NuPattern.VisualStudio.TemplateWizards, Version=1.0.0.0, Culture=neutral, PublicKeyToken=24c7786d4a8b1a88</Assembly>`
	`  <FullClassName>NuPattern.VisualStudio.TemplateWizards.ElementReplacementsTemplateWizard</FullClassName>`
	`</WizardExtension>`

All Text Template Files (.t4):*

Add the <#@ Assembly Name="NuPattern.Common.dll" #> directive to the top of the text template	
VS2010, VS2012	`<#@ Assembly Name="NuPattern.Common.dll" #>`
Add the <#@ Import Namespace="NuPattern.Runtime.ToolkitInterface" #> directive to the top of the text template	
VS2010, VS2012	`<#@ Import Namespace="NuPattern.Runtime.ToolkitInterface" #>`

All Wizard and Wizard Page XAML Files (.xaml):*

In Wizard XAML and WizardPage XAML pages, modify the namespace that includes the path '/visualstudiopatterning/2010/' at the top of the file.	
VS2010, VS2012	`xmlns:p="http://schemas.microsoft.com/nupattern/2012/xaml/"`
In Wizard XAML pages (not WizardPage XAML files), add the following element as the first child element of the <WindowWizard>, before the <WizardWindow.Pages> element:	

VS2010, VS2012	`<p:WizardWindow.Resources>`
	`<ResourceDictionary>`
	`<ResourceDictionary.MergedDictionaries>`
	`<ResourceDictionary`
	`Source="/NuPattern.Presentation;component/Resources/CommonStyles.xaml" />`
	`</ResourceDictionary.MergedDictionaries>`
	`</ResourceDictionary>`
	`</p:WizardWindow.Resources>`

All Project Template Project files (.csproj)*

Add the following "Import" to the top of the file, before the first <PropertyGroup> element.	

VS2012	`<Import`
	`Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props`
	`"`
	`Condition="Exists('$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Co`
	`mmon.props')" />`

Pattern Toolkit Automation Project:

Toolkit Automation Project File (csproj):

Note: Before opening your pattern toolkit project or solution in Visual Studio.

Open in 'XML View':

Add the following "Import" to the top of the file, before the first <PropertyGroup> element.	
VS2012	`<Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props" Condition="Exists('$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props')" />`
Change all "Microsoft.VisualStudio.*.dll" version 10.0 assembly references to version 11.0 assembly references.	
VS2012	`<Reference Include="Microsoft.VisualStudio.ExtensibilityHosting, Version=11.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL" />`
Change all "Microsoft.VisualStudio.*.10.0.dll" assembly references to "Microsoft.VisualStudio.*.11.0.dll" assembly references.	

VS2012	`<Reference Include="Microsoft.VisualStudio.Shell.Immutable.11.0, Version=11.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL" />`
	`<Reference Include="Microsoft.VisualStudio.Modeling.Sdk.11.0, Version=11.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL" />`
	`<Reference Include="Microsoft.VisualStudio.Modeling.Sdk.Diagrams.11.0, Version=11.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL" />`

Change all "Microsoft.VisualStudio.Patterning.*" assembly references to "NuPattern.*.dll" assembly references. e.g.	

VS2010, VS2012	`<Reference Include="NuPattern.Runtime.Interfaces">`
	`<HintPath>$(PatternToolkitRuntime)\NuPattern.Runtime.Interfaces.dll</HintPath>`
	`<Private>false</Private>`
	`</Reference>`

(Optional) Change the value of the "TargetFrameworkVersion" property:	
VS2012	`<TargetFrameworkVersion>v4.5</TargetFrameworkVersion>`
Change the "Import" to the "Microsoft.VisualStudio.Patterning.Authoring.PatternToolkitVersion.targets":	
VS2010, VS2012	`<Import Project="$(localappdata)\Microsoft\MSBuild\NuPattern\NuPattern Toolkit Builder\NuPattern.Authoring.PatternToolkitVersion.targets" />`

Replace the assembly reference to “NuPattern.Runtime.Interfaces”:	

VS2010, VS2012	`<Reference Include="NuPattern.Runtime.Extensibility">`
	`  <HintPath>$(PatternToolkitRuntime)\NuPattern.Runtime.Extensibility.dll</HintPath>`
	`</Reference>`

(Optional) Replace the assembly reference to “NuPattern.Extensibility”:	

VS2010, VS2012	`<Reference Include="NuPattern.Runtime.Extensibility">`
	`  <HintPath>$(PatternToolkitRuntime)\NuPattern.Runtime.Extensibility.dll</HintPath>`
	`</Reference>`

Add the assembly reference to “NuPattern.Common”:	

VS2010, VS2012	`<Reference Include="NuPattern.Common">`
	`  <HintPath>$(PatternToolkitRuntime)\NuPattern.Common.dll</HintPath>`
	`  <Private>false</Private>`
	`</Reference>`

Replace the assembly reference to “NuPattern.Common.Presentation”:	

VS2010, VS2012	`<Reference Include="NuPattern.Presentation">`
	`  <HintPath>$(PatternToolkitRuntime)\NuPattern.Presentation.dll</HintPath>`
	`</Reference>`

Replace the CatalogName attribute in the AssemblyInfo.cs file.	
VS2010, VS2012	`[assembly: VsCatalogName(NuPattern.ComponentModel.Composition.Catalog.DefaultCatalogName)]`

All Automation Classes (Commands, Conditions, ValueProviders etc.) (.cs)*

Refactor the tracer declaration ITraceSource . e.g.	
VS2010, VS2012	`private static readonly ITracer tracer = Tracer.Get<MyClass>();`

All Automation Commands (.cs)*

Rename the base class from ‘FeatureCommand’ to ‘NuPattern.Runtime.Command’. e.g.	
VS2010, VS2012	`public class MyCommand : NuPattern.Runtime.Command`

Pattern Toolkit Solution

Final Steps

VS2010, VS2012:

- Reload the Solution (and all Projects)

1. Open and save all <ToolkitProject>\PatternModel.patterndefinition files. (This will automatically upgrade this file and its *.diagram files, and save a backup file).
2. Delete the following generated files:
    - <ToolkitProject>\source.include.t4. (Note: This file is nested under <ToolkitProject>\source.extension.tt)
    - <ToolkitProject>\GeneratedCode\Guidance\GuidanceWorkflow.cs
    - All files in: <ToolkitAutomationProject>\GeneratedCode\<ToolkitProjectName>*.cs
3. In Solution Builder, if you have defined any toolkit guidance:
    - Right-click on the '<ToolkitName>/Assets/Guidance' element, and select 'Build Guidance'
4. In Solution Builder, if the 'TransformOnBuild' property of the '<ToolkitName>' element and '<ToolkitName>/Automation/Library' elements are set to 'Never', then manually:
    - Right-click on the '<ToolkitName>' element and 'Transform Templates'
    - Right-click on the '<ToolkitName>/Automation/Library' element and 'Transform Templates'
    - Right-click on the '<ToolkitName>/Toolkit Info' element and select 'Transform Toolkit Info'.
5. In Solution Explorer, if you have any additional generated code files (*.tt files other than 'source.extension.tt'), then transform those manually.
6. The following types and methods have been changed and will no longer compile. You will need to manually find & replace all of the following occurrences:

Old Type/Method	New Type/Method
FeatureCommand	Runtime.Command
IFeatureManager	IGuidanceManager
ITraceSource	ITracer
ITraceSource.TraceInformation	ITracer.Info
ITraceSource.TraceWarning	ITracer.Warning
ITraceSource.TraceError	ITracer.Error

7. Clean & Rebuild the Solution