Release Notes VSPAT 1.3.20.0

This minor release of VSPAT is a compatibility release for Visual Studio 2012 that continues to work for Visual Studio 2010.

It requires critical manual changes to all existing Pattern Toolkits for them to be used in Visual Studio 2012, as well as used in Visual Studio 2010.

This version of VSPAT now embeds the ‘Feature Extension Runtime’ extension which was previously installed by VSPAT, and is installed by the ‘Feature Builder Power Tool’. This release of VSPAT will not work correctly, nor will any pattern toolkits built with any version of VSPAT work when the ‘Feature Extension Runtime’ extension is also installed in Visual Studio 2010.

**IMPORTANT**: The ‘Feature Extension Runtime’ extension is automatically installed when the ‘Feature Builder Power Tool’ extension is installed, or when any Feature Extension built by the ‘Feature Builder Power Tool’ is installed, or when any Pattern Toolkit built by a prior version of VSPAT is installed.

The ‘Feature Extension Runtime’ extension must be disabled or uninstalled in order to use VSPAT, or any toolkit created with the current version of VSPAT.

# What happens to toolkits that embed the Runtime from 1.2.19.0?

# Do toolkits that were built with a previous version need migrating?

Yes, if you wish to use this version of VSPAT, or with VS2012.

# Will VSPAT update from the Visual Studio Gallery?

# What are the main differences in this version? Signing key? Owner?

# What happens if I want to target my toolkit for VS2010, or VS2012?

# What happens if the FERT is already installed or is installed after this version?

Note: Only applies to Visual Studio 2010.

When Visual Studio starts, or while using Visual Studio, the following error is reported:



This error may appear multiple times while using Visual Studio.

You must either [Disable] or [Uninstall] the ‘Feature Extension Runtime’ extension in ‘Extension Manager’.



# What’s New?

* This version of VSPAT works in Visual Studio 2012, as well as in Visual Studio 2010.
* Pattern Toolkits that were built with a previous version of VSPAT (version 1.2.19.0 or earlier) will need to be rebuilt with this version of VSPAT in order to work.
* Very minor bug fixes.

# Migration Notes

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

**WARNING**: The VSPAT extensions are now publically [available online](http://visualstudiogallery.msdn.microsoft.com/332f060b-2352-41c9-b8dc-95d8ad21329b) at the [Visual Studio Gallery](http://visualstudiogallery.msdn.microsoft.com/), and existing toolkit users with existing installed pattern toolkits (built against previous versions of VSPAT) are free to download and install the latest version of VSPAT for creating their own pattern toolkits. Should they either install the latest version of VSPAT ( the ‘Pattern Toolkit Builder’ extension) or install newer pattern toolkits, then their existing pattern toolkits solution (built with a previous version of VSPAT) will no longer work for them. We strongly recommend upgrading the existing pattern toolkits.

## Who Needs to Migrate?

You need to migrate existing pattern toolkits in either of the following scenarios:

1. As a user, you are using a pattern toolkit (built against a previous version VSPAT), and wish to be upgraded to the latest version of the pattern toolkit on your machine.
   * This upgrade will require your to manually:
     1. Obtain a newer version of your pattern toolkit from the orginal author, who has already upgraded it to the latest version of VSPAT (see next scenario for details).
     2. Uninstall your existing pattern toolkit
     3. Uninstall all VSPAT related extensions (i.e. the ‘Pattern Toolkit Runtime’ extension)
     4. Manually install the latest version of the pattern toolkit.
   * Note: This kind of upgrade will not happen automatically for a user.
2. As an author, you are currently developing an existing pattern toolkit (built against a previous version VSPAT), and wish to upgrade it to the latest version of VSPAT.
   * This upgrade will require you to manually:
     1. Uninstall the current version of VSPAT (i.e. the ‘Pattern Toolkit Builder’, ‘Pattern Toolkit Library Suppport’, and ‘Pattern Toolkit Runtime’ extensions)
     2. Install the latest version of ‘Pattern Toolkit Builder’ extension.
     3. Migrate the pattern toolkit to the latest version of VSPAT.
   * Note: this kind of upgrade will not happen automatically for an author.

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

Migration notes (In progress):

* Convert projects using DSLConverterTool
* **Do we really need this step? Our customers don't typically have actual DSLs in toolkit projects, and my experience with the DSL migration tool teaches me that we don’t actually need this step for pattern toolkit projects – I don’t think? But I can make a note for those toolkit authors that do.**
* Changed target file: $(MSBuildExtensionsPath)\Microsoft\VisualStudio\TextTemplating\v10.0\Microsoft.TextTemplating.targets

To:

$(MSBuildExtensionsPath)\Microsoft\VisualStudio\v11.0\TextTemplating\Microsoft.TextTemplating.targets

**OK, again I don’t think is is typical for most toolkit projects, at least the out-of-the-box pattern toolkit projects does not put these targets in the project file. But I will put a note in the migration notes for anyone who actually put this import in their project file.**

* Updated VSPAT assemblies references to find on the correct folder (updated VSPAT version)
* **This will be handled automatically when the new VSPAT version is installed.**
* Fix references to VSSDK 10 assemblies on XAMLs files
* **Confirm, this is definitely a manual step at this point in time.**
* Make sure there are not references to .10.0 assemblies in the code
* **Yes, I will leave a note about this in the migration guide as well.**
* NOT REMOVE VSSDK.targets file, that's the one who install the toolkit into the ExpHive and generate the VSIX
* **We will be replacing the standard Import: <Import Project="$(MSBuildExtensionsPath)\Microsoft\VisualStudio\v10.0\VSSDK\Microsoft.VsSDK.targets" />**
* **with something that pre-calculates the targets path for them, so in the future they wont have to worry about this. Something like this: <Import Project="$(VsSDKTargetPath)\Microsoft.VsSDK.targets" />**

### Pattern Toolkit Solutions:

#### Solution file (\*.sln)

* + Open in ‘XML View’:
  + Modify the following:

GlobalSection(ExtensibilityGlobals) = postSolution

Features = 9f6dc301-6f66-4d21-9f9c-b37412b162f6:Creating Pattern Toolkits:1.3.20.0

EndGlobalSection

#### Solution Builder File (\*.slnbldr)

* + Open in ‘XML View’:
  + Modify any <product> elements where DefinitionName=”PatternToolkit” to the following:

<product definitionName="PatternToolkit" extensionName="Pattern Toolkit Builder" version="1.3.20.0">

### Pattern Toolkit Project:

#### Toolkit Project File (csproj):

* + Unload project, and edit the XML:
  + (Optional) Remove:

<Import Project$(MSBuildExtensionsPath)\Microsoft\VisualStudio\v10.0\VSSDK\Microsoft.VsSDK.targets" />

Reload the project

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

* + Add version and update the 'PublicKeyToken' strings from all <WizardExtension> elements for assemblies beginning with ‘Microsoft.VisualStudio.Patterning’. e.g.

<WizardExtension>  
    <Assembly>Microsoft.VisualStudio.Patterning.Library, Version=1.3.20.0, PublicKeyToken=31bf3856ad364e35</Assembly>

#### Source.extension.tt

* + (Optional, for VS2012 Compatibility) Add a <SupportedVersion>: