**[File types and extensions in Visual Studio 2012](http://blog.280z28.org/archives/2013/04/165/" \o "Permanent Link: File types and extensions in Visual Studio 2012)**

The Visual Studio 2010 SDK introduced the new class [FileExtensionToContentTypeDefinition](http://blog.280z28.org/go.php?http://msdn.microsoft.com/en-us/library/microsoft.visualstudio.utilities.fileextensiontocontenttypedefinition.aspx). This post is all about why I *no longer use/export* this class, and what I now do instead. The steps in this post only associate specific, known extension(s) with a content type. Further integration into the user-customizable File Types settings and Open With… features requires additional work described in my answer to this Stack Overflow question: [Supporting user-specified file extensions in custom Visual Studio language service](http://blog.280z28.org/go.php?http://stackoverflow.com/questions/15263761).

The following requirements led me away from this attribute.

1. The Type & Member Dropdown Bars feature is not exposed through an MEF editor extension point. Supporting this feature requires providing an implementation of [IVsCodeWindowManager](http://blog.280z28.org/go.php?http://msdn.microsoft.com/en-us/library/microsoft.visualstudio.textmanager.interop.ivscodewindowmanager.aspx) (beyond the scope of this article).
2. To provide an implementation of IVsCodeWindowManager, you must provide an implementation of [IVsLanguageInfo](http://blog.280z28.org/go.php?http://msdn.microsoft.com/en-us/library/microsoft.visualstudio.textmanager.interop.ivslanguageinfo.aspx).
3. If you provide an implementation of IVsLanguageInfo, the [GetFileExtensions](http://blog.280z28.org/go.php?http://msdn.microsoft.com/en-us/library/microsoft.visualstudio.textmanager.interop.ivslanguageinfo.getfileextensions.aspx) and [GetLanguageName](http://blog.280z28.org/go.php?http://msdn.microsoft.com/en-us/library/microsoft.visualstudio.textmanager.interop.ivslanguageinfo.getlanguagename.aspx) methods of that interface will be used *instead* of the values given to FileExtensionToContentTypeDefinition.

To maximize your ability to support the full range of Visual Studio features in your extension, you can take the following steps to register your new language or file type.

**1. In your VSPackage.resx file, add the following values.**

* 100 – *Language name*
* 110 – *Package name*
* 111 – *Package description*

**2. Derive a package class from** [**Package**](http://blog.280z28.org/go.php?http://msdn.microsoft.com/en-us/library/microsoft.visualstudio.shell.package.aspx)**.**

|  |
| --- |
| [PackageRegistration(UseManagedResourcesOnly = **true**)]  [InstalledProductRegistration("#110", "#111", "1.0")]  [Guid("your guid here")]  **public** **class** ExamplePackage : Package  {  } |

**3. Implement IVsLanguageInfo.**

This is actually pretty simple.

|  |  |
| --- | --- |
|  | using System;  using System.Runtime.InteropServices;  using Microsoft.VisualStudio;  using Microsoft.VisualStudio.TextManager.Interop;    [Guid("your guid here")]  internal class ExampleLanguageInfo : IVsLanguageInfo  {  public int GetCodeWindowManager(IVsCodeWindow pCodeWin, out IVsCodeWindowManager ppCodeWinMgr)  {  ppCodeWinMgr = null;  return VSConstants.E\_NOTIMPL;  }    public int GetColorizer(IVsTextLines pBuffer, out IVsColorizer ppColorizer)  {  ppColorizer = null;  return VSConstants.E\_NOTIMPL;  }    public int GetFileExtensions(out string pbstrExtensions)  {  pbstrExtensions = ".e1;.e2";  return VSConstants.S\_OK;  }    public int GetLanguageName(out string bstrName)  {  bstrName = "Example";  return VSConstants.S\_OK;  }  } |

**4. Register the IVsLanguageInfo implementation**

Register the IVsLanguageInfo implementation using the [ProvideLanguageServiceAttribute](http://blog.280z28.org/go.php?http://msdn.microsoft.com/en-us/library/microsoft.visualstudio.shell.providelanguageserviceattribute.aspx) and [ProvideLanguageExtensionAttribute](http://blog.280z28.org/go.php?http://msdn.microsoft.com/en-us/library/microsoft.visualstudio.shell.providelanguageextensionattribute.aspx) registration attributes. Override the [Package.Initialize](http://blog.280z28.org/go.php?http://msdn.microsoft.com/en-us/library/microsoft.visualstudio.shell.package.initialize.aspx) method to provide the implementation. Here is the updated ExamplePackage class.

|  |  |
| --- | --- |
|  | using System;  using System.Runtime.InteropServices;  using Microsoft.VisualStudio.Shell;  using IServiceContainer = System.ComponentModel.Design.IServiceContainer;    [PackageRegistration(UseManagedResourcesOnly = true)]  [InstalledProductRegistration("#110", "#111", "1.0")]    [ProvideLanguageService(typeof(ExampleLanguageInfo), "Example", 100)]  [ProvideLanguageExtension(typeof(ExampleLanguageInfo), ".e1")]  [ProvideLanguageExtension(typeof(ExampleLanguageInfo), ".e2")]    [Guid("your guid here")]  public class ExamplePackage : Package  {  private ExampleLanguageInfo \_languageInfo;    protected override void Initialize()  {  base.Initialize();    \_languageInfo = new ExampleLanguageInfo();  ((IServiceContainer)this).AddService(typeof(ExampleLanguageInfo), \_languageInfo, true);  }  } |

**5. Create and export a** [**ContentTypeDefinition**](http://blog.280z28.org/go.php?http://msdn.microsoft.com/en-us/library/vstudio/microsoft.visualstudio.utilities.contenttypedefinition.aspx)**.**

**NOTE:** Since we are not using a [custom editor factory](http://blog.280z28.org/go.php?http://stackoverflow.com/questions/15263761), the content type name used here must match the language name used as the 2nd argument to ProvideLanguageExtensionAttribute above.

|  |  |
| --- | --- |
|  | using System.ComponentModel.Composition;  using Microsoft.VisualStudio.Utilities;    public static class ExampleDefinitions  {  /\* The language name (used for the language service) and content type must be the same  \* due to the way Visual Studio internally registers file extensions and content types.  \*/  [Export]  [Name("Example")]  [BaseDefinition("code")]  private static readonly ContentTypeDefinition ExampleContentTypeDefinition;  } |