Designer View Over Xml Editor

This sample demonstrates how to create an extension that provides a WPF-based Visual Designer for editing XML files with a specific schema (XSD) in coordination with the Visual Studio XML Editor.  
  
In this sample we implement a basic view for a *.vstemplate* file.

# Requirements

* Visual Studio 2010
* Visual Studio 2010 SDK

# Getting Started

1. Download and unzip the sample
2. Open the solution file
3. Build the solution
4. Open the Visual Studio experimental instance by pressing F5

# Files

* **VsTemplateDesignerPackage.cs** – registers the designer, via **ProvideXmlEditorChooserDesignerView**, as the preferred editor view for files with the *.vstemplate* extension and indicated schema
* **EditorFactory.cs** – determines if the document to be edited already exists (was already opened in the XML Editor view), rather than assuming it must be created; creates the designer’s EditorPane as the new Editor
* **EditorPane.cs** – creates the sited designer control and associated XmlModel for the file and text buffer
* **IViewModel.cs** – expresses the interface needed to bind the designer controls to the XmlSchema
* **ViewModel.cs** – implements **IViewModel** and manages the events that synchronize data between the fields in the designer and the underlying XML document, which may also be seen in the XML Editor
* **VsDesignerControl.xaml[.cs]** – implements the WPF controls expressing the designer interface and binds them to the ViewModel
* **VsTemplateSchema.cs** – XML schema file generated via   
  xsd.exe vstemplate.xsd /classes /e /n:MyNameSpace

# To test the sample functionality

1. On the **File** menu, click **Open**
2. Browse to the **TestTemplates** sub-directory within the solution.
3. Select and **Open** one of the files.
4. A new tab opens with the contents of the file laid out in the fields of a WPF form.
5. On the **View** menu, click **Code**
6. An additional tab opens with the contents of the file formatted by the XmlEditor
7. Editing in either tab will also update the contents of the other tab (although you must tab out of a form field before the update is sent to the underlying document).

# Additional Resources