PkgDef Editor

This sample demonstrates how to create a package of Editor Extensions that provide across-the-board support of a language without implementing a full-blown language service.

Most features are implemented as some form of **ITagger**.  
  
In this sample we implement support for *.pkgdef* files, which are used to register VSPackages and other items with Visual Studio or Visual Studio Shell applications. Included are:

* Colorizing of keys, value names, values, comments, string tokens, and guids
* Syntax validation (showing errors with squiggles and QuickInfo)
* Token validation (showing errors with squiggles and QuickInfo)
* Token completion (when a ‘$’ is typed) and QuickInfo
* Supports the continuation line syntax (i.e., ‘\’ at the end of a line)
* Highlights matching braces and quotes
* Outline collapsing of values beneath sections (keys)
* List errors in the Error List window (and allows double click jump to the associated line)

# 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. Run solution in the Visual Studio experimental instance by pressing F5

# Test the Functionality

1. On the **File** menu, click **Open**
2. Browse to the **TestPkgDefs** sub-directory within the solution.
3. Select and **Open** one of the files.
4. A new tab opens with the contents of the file colorized. (Yes, there are errors in *blog.pkgdef*)
5. Try typing and hovering over various language elements.

# Files

* **PkgDefLanguage.cs** – implements **PkgDefTaggerProvider**, which provides the core of the language processing, handling continuation lines and returning tags for each language element
* **PkgDefLanguageTokens.cs** – enumerates the token types and some associated string constants
* **ErrorTagger.cs** – implements an **ITaggerProvider** for tags of type **ErrorTag**; uses the **PkgDefTaggerProvider** for parsing and filters and translates the tokens appropriately; also implements a buffer change event handler that watches for changes and idle time (see **BufferIdleEventUtil.cs**) to update the **Error List** window with error tasks
* **Classifier\Classifier.cs** – implements an **ITaggerProvider** for tags of type **ClassificationTag**; uses the **PkgDefTaggerProvider** for parsing and filters and translates the tokens appropriately
* **Classifier\ClassificationFormat.cs** – defines the **EditorFormatDefinition**s that are associated with each tag returned by the classifier
* **Classifier\ClassificationType.cs** – defines the Content Type and File Extensions supported by the classifier, along with the classification types
* **OutlineTagger.cs** – implements an **ITaggerProvider** for tags of type **IOutliningRegionTag** using its own parser that only looks for top-level regions to collapse; also uses a buffer change event handler (see **BufferIdleEventUtil.cs**) to scan the entire document during idle time (rather than try to track the possible effect of each change on each region)
* **BufferIdleEventUtil.cs** – implements a buffer change event tracker with a timer than accumulates changes until the user stops typing
* **BraceMatching.cs** – implements an **IViewTaggerProvider** for tags of type **TextMarkerTag** that monitors changes in the typing caret, watching for braces and quotes, so that it can highlight the matching item
* **Intellisense\QuickInfoController.cs** – implements a **IIntellisenseController** needed to translate mouse hover events into QuickInfo sessions
* **Intellisense\QuickInfoControllerProvider.cs** – implements the factory for the **IIntellisenseController** for pkgdef files
* **Intellisense\PkgDefTokenQuickInfoSource.cs** – provides QuickInfo data for pkgdef tokens (string substitution)
* **Intellisense\CompletionController**.cs – implements an **IOleCommandTarget** that watches for appropriate typing events to initiate (tokens start with ‘$’), filter, and finish token completion
* **Intellisense\CompletionSource.cs** – provides the actual list of completion items