

Installing the MVVM Light Toolkit and Additional Components

Laurent Bugnion
@LBugnion
<http://www.galasoft.ch>



pluralsight 
hardcore developer training

Outline

- **Using the assemblies from Nuget**
- **Installing the whole package in Visual Studio**
- **Using the code snippets**
- **Creating a new project using the project templates**
- **Discovering the Windows 8.1 project template**
- **Using the item templates**

Using the Assemblies from Nuget



What is Nuget?

- **Package management system**
 - Developer can define a package
 - → including dependencies to other packages.
 - User adds references to package
 - → Automatically pulls dependencies
 - → Update mechanism
- **Available as command line tool**
 - Ideal for automated build process
- **Available as Visual Studio extension**
 - Ideal for humans

Advantages of Nuget

- **Packages can contain more than just binaries**
 - Content files
 - Scripts to modify the target project
- **Manage dependencies per project**
 - Dependencies are committed to source control
 - Easy to restore a working environment
- **Easy to handle updates**
- **Nuget server can also be local to a firm, etc**
- **Main disadvantage:**
 - Requires the PC to be online when creating the project.

Installing Nuget

- **Already available in VS2012 and VS2013**
- **Must be installed in VS2010**

MVVM Light "Libs only" Vs "Full"

- **"MVVM Light libraries only" installs only the DLLs**
 - Core
 - Extras
 - Dependency to CommonServiceLocator
- **"MVVM Light" installs files in addition**
 - MainViewModel
 - ViewModelLocator
 - Modifies App.xaml
 - Dependency to "MVVM Light libraries only"
- **PCL**
 - Portable class libraries

Installing the Full Toolkit



Additional Tools

- **Project and item templates**

- Allow to get started faster and avoid "plumbing"
- Templates implement the MVVM Light architecture

- **Code snippets**

- To avoid writing repetitive code
- A choice of the most commonly used constructs

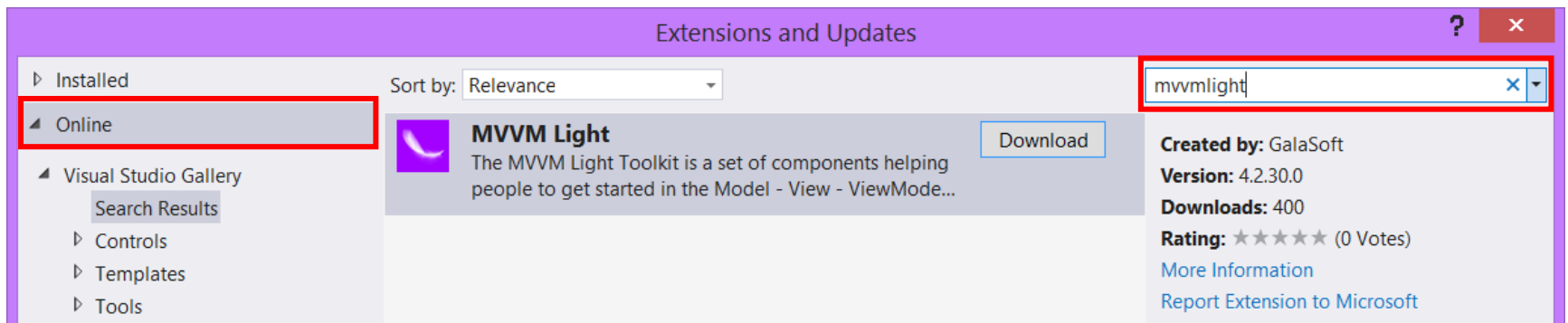
- **First please uninstall previous versions**

- <http://mvvmlight.net/cleaning>

Installation Process

- **In Visual Studio:**

- Tools, Extensions and Updates, Online, Search for "mvvmlight"



- **Or download the VSIX**

- <http://mvvmlight.codeplex.com> and then Downloads
- Make sure to pick the right VSIX!

Installation Process (Express)

- **Download the VSIX**

- <http://mvvmlight.codeplex.com> and then Downloads
- Make sure to pick the right VSIX!

Code Snippets

- **Simply type "mvvm"**

- The list of all snippets shows in Intellisense
- Select a snippet
- "Tab" to expand
- "Tab" to select the different fields

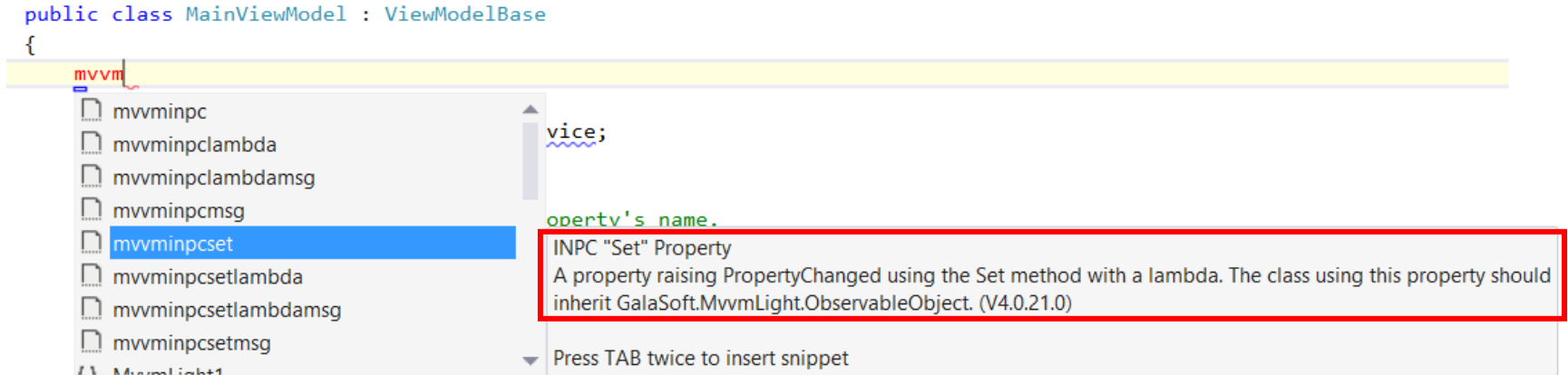
- **Grouped by:**

- **inpc**: Adds an "observable property" (INotifyPropertyChanged)
- **locatorproperty**: Adds a ViewModel property to the ViewModelLocator
- **prop**: Adds an Attached or Dependency Property (WPF style)
- **relay**: Adds a RelayCommand
- **slprop**: Adds an Attached or Dependency Property (Silverlight style)
- **vm**: Adds an access to the ViewModel inside the View (DataContext)

- **Example:**

- **mvvmrelaygeneric**: Adds a RelayCommand with parameter (generic)

Code Snippets



■ Known issues:

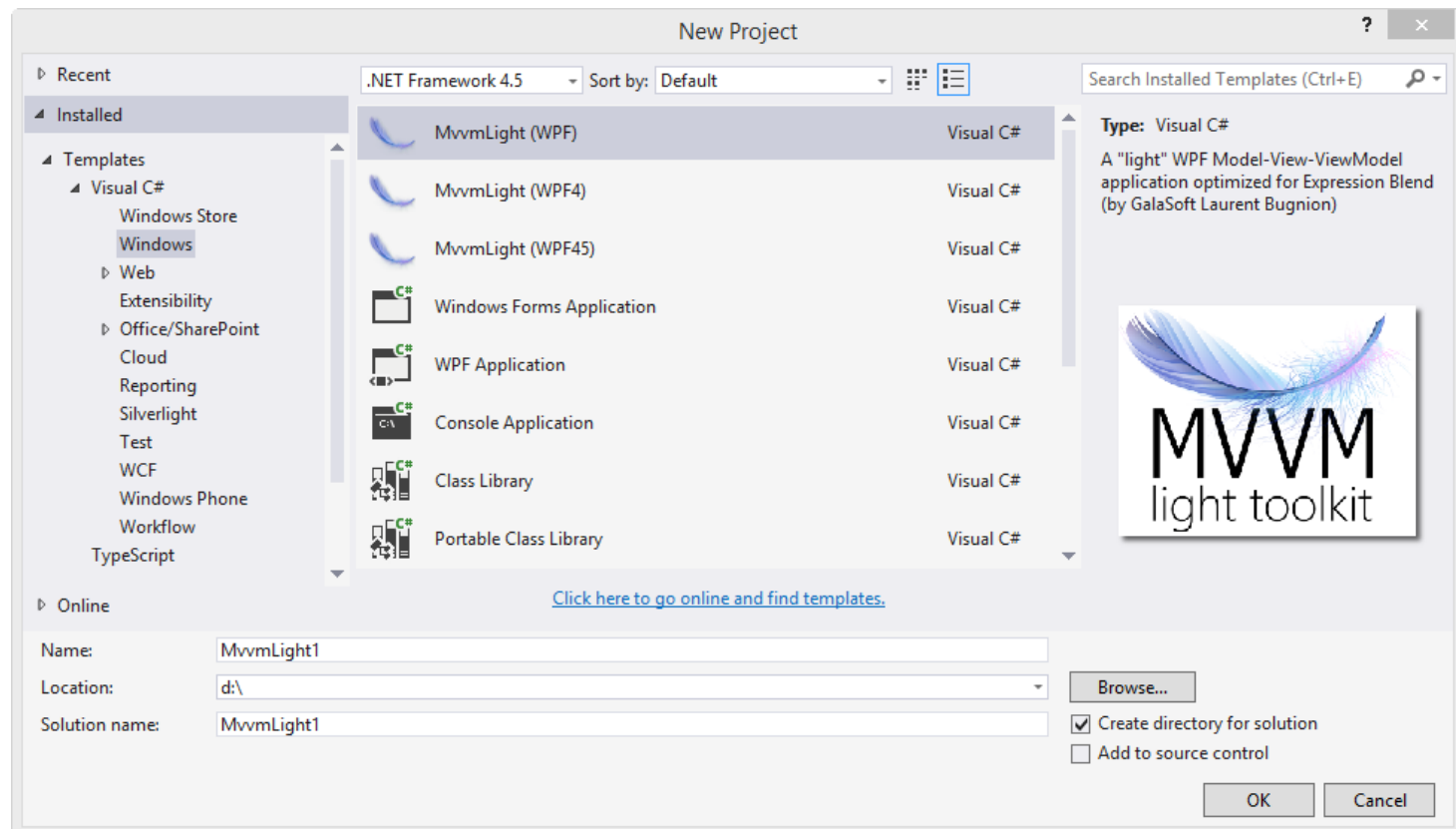
- ❑ Conflicts with Resharper snippets
- ❑ If you don't see the MVVM Light snippets, try deactivating the Resharper snippets

The "prop" Snippets

- **mvvmpropdp**
 - Declares a dependency property in WPF
- **mvvmpropa**
 - Declares an attached property in WPF
- **mvvmslpropdp**
 - Declares an attached property in Silverlight, Windows Phone and Windows 8
- **mvvmslpropa**
 - Declares an attached property in Silverlight, Windows Phone and Windows 8

The Project Templates

- **File, New Project**
 - Available for WPF, Silverlight, Windows Phone, Windows Store



Summary

- **Using the assemblies from Nuget**
- **Installing the whole package in Visual Studio**
- **Using the code snippets**
- **Creating a new project using the project templates**
- **Discovering the Windows 8.1 project template**
- **Using the item templates**