

# Commanding in Silverlight with mvvm

By Jeremiah Redekop

<http://blogs.geniuscode.net/JeremiahRedekop>

[jeremiah@geniuscode.net](mailto:jeremiah@geniuscode.net)

# Evening Outline - 45min

- Mvvm Overview
  - ViewModel Locator
- Commanding Overview
- MVVMLight Framework
- Demo 1: Code-Behind
- Demo 2a Behaviours
- Demo 2b: Binding with Commanding
- Demo 3: Unit Testing ViewModel with Silverlight Test Framework
- Demo 4: OData StackOverflow.com example
- Q&A

# CODE!

- Available right now
- Follow along!
- URL:
- <http://bit.ly/bM2rZn>

The screenshot shows the VanSLUG Community Forum page. The header includes the site name and navigation links. The main content area displays an announcement for a session on November 24. The announcement is posted by Jeremiah Redekop and includes details about the session's format, topics, and speakers. A red arrow points to a link in the announcement text that says 'click here'.

**VanSLUG**  
Community Forum


Home  
Help  
Register  
Log In  
Search

Active Threads

You are here: [Home](#) > [Vancouver Silverlight User Group](#) > [Announcements](#) > Nov 24 Session: MVVM & Commanding + WP7

Page: 1/1

**Announcements**  
Nov 24 Session: MVVM & Commanding + WP7

Poster	Message
 Jeremiah Redekop	<p># Posted on: 19-Nov-2010 00:35:31.</p> <p><b>Date &amp; Time:</b> Wed, Nov 24, 2010, 6:30 PM (Welcome time 6:15 PM)</p> <p><b>Location:</b> Building SR3 room 1710, BCIT Burnaby Campus, 3700 Willingdon Ave, Burnaby, BC, V5G 3H2</p> <p><b>Registration:</b> <a href="#">Click here</a></p> <p>We are trying a new format whereby two 45 minutes long topics will be presented by two separate speakers.</p> <p><b>Topic 1:</b> MVVM and Commanding in Silverlight Mvvm is a paradigm that separates data access code from user interface code. It is a very popular architecture used by many Silverlight developers.</p> <p>We are going to explore how to call methods on the viewmodel from the view using a technique called commanding.</p> <p>We will also briefly touch on some of the benefits of this approach, such as unit testing.</p> <p><b>Speaker:</b> Jeremiah Redekop</p> <p>Jeremiah is partner &amp; solutions architect with GeniusCode, a .net startup which has recently discovered a patent-pending calculation technology.</p> <p>He took the Silverlight tour in May, 2010, and has since invested heavily in Silverlight.</p> <p><b>Topic 2:</b> Using your .NET &amp; Silverlight skills to program the Windows Phone 7 The Windows Phone 7 (WP7) has recently been released into the market. Those who are familiar with .NET and Silverlight will find themselves in familiar territory when programming for the WP7 platform. This session is really an introduction into what you need to start developing applications for the WP7. We are hoping to continue presenting more advanced topics in future.</p> <p><b>Speaker:</b> Medhat Elmasy</p> <p>Medhat Elmasy was born in Addis Ababa, Ethiopia. He holds a Masters Degree in Computing Science from Brunel University, England. Medhat moved to Canada from Egypt with his wife, son, and daughter in 1993. He is currently an instructor at the British Columbia Institute of Technology (BCIT). He specializes in web development and is involved in most of the ASP.NET and Silverlight courses at BCIT. Medhat is currently a contractor at Sierra Wireless Inc. He is a founder of the .NETBC user group and the current president of the VanSLUG (Silverlight) user group in Vancouver. Medhat is also co-owner of bcjobs.com. Dabbling in new technologies is a passion.</p> <p>The Vancouver VanSLUG Users Group is a community organization established on May 1, 2010. Our sponsors are Microsoft, BCIT, and INETA (<a href="http://www.ineta.org">www.ineta.org</a>). We focus on Silverlight. There are no fees for attending and all are welcome. Please inform anyone you know who might be interested. The users group registration site is located at <a href="http://vanslug.net">http://vanslug.net</a>.</p> <p># Posted on: 19-Nov-2010 00:39:05.</p> <p>Here is a link to the MVVM Commanding Code on GitHub: <a href="#">click here</a></p> <p>Once on the page, just click the download button to get the zipped contents 📄</p> <p>Slides are available <a href="#">here</a></p>

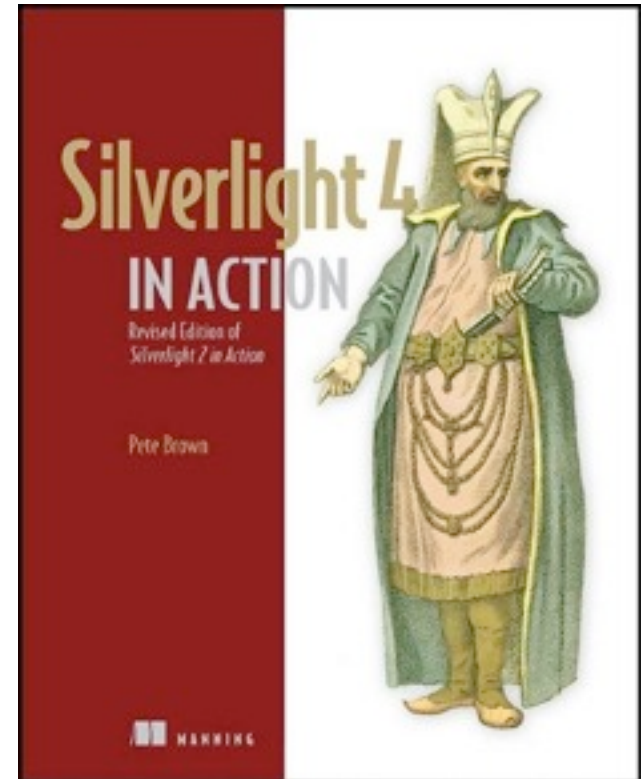
- Link in forums:
- Announcements > Nov 24 Session: MVVM & Commanding WP7

# Silverlight 4 Resources

- Silverlight TV
- Book: Silverlight 4 In Action - Pete Brown
- Official Silverlight Forums
- Silverlight User Group (meetings & forums)

# Post Questions now

- Twitter: #vanslug
- Forum: Event Announcement Forum
- eligible for:
  - Silverlight 4 In Action Book (\$60)
  - LINQPad License (\$25)

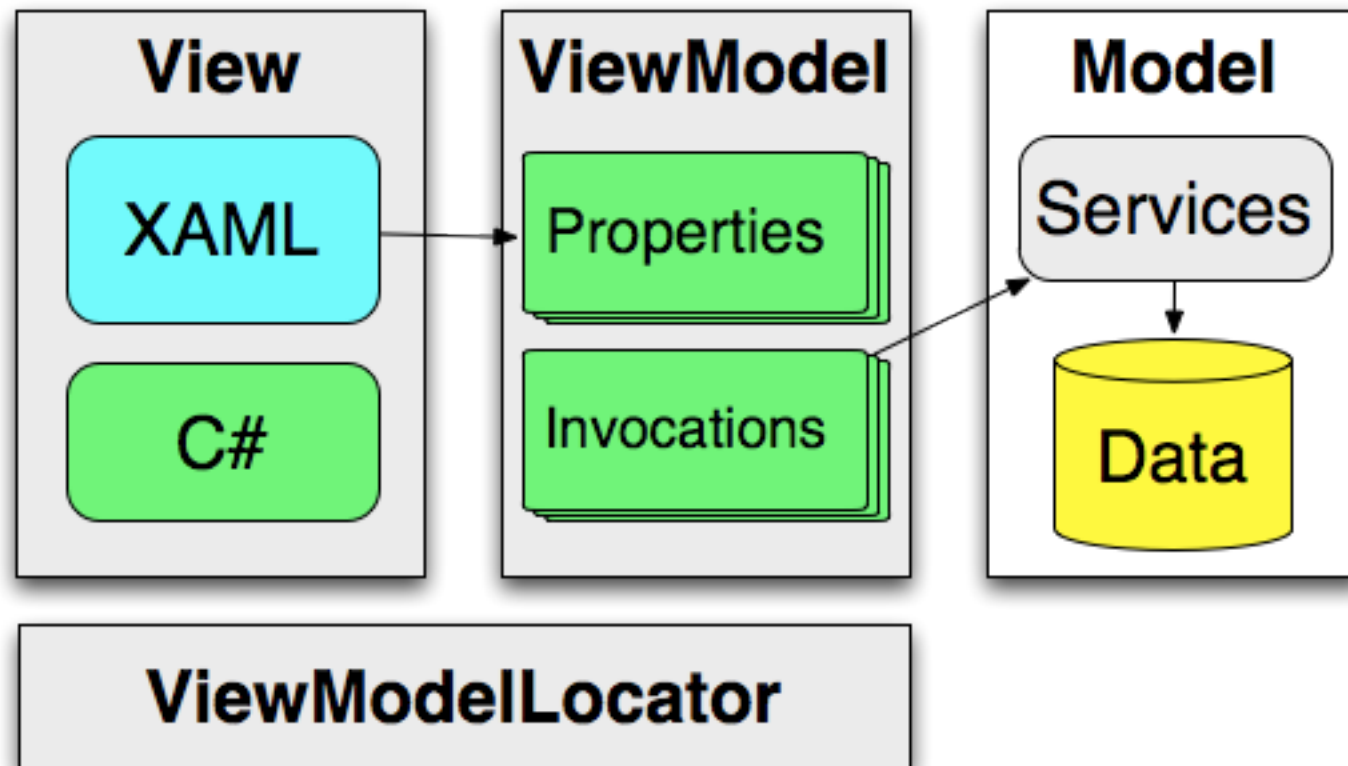


# Important Disclaimer

- There is no best, or “right” way for MVVM!
- MVVM is just a pattern
- There are pros & cons to any approach: use what works
- Understand the underlying technologies
- Reality Check: your client doesn't care about mvvm

# What is Mvvm?

- Separation of Concerns



# What is the ViewModel?

- DataContext of the View
- Connection to Model
- Encapsulates & Isolates Logic



# Where does ViewModel Come From?

- The View
  - Code
    - View's code behind creates ViewModel in constructor or event
  - Markup (XAML)
    - ViewModel is declared in the markup
    - Exactly the same result as code-behind
    - Allows for design time data preview
- External Source (ViewModel Locator)
  - Locator is declared in Markup, exposes ViewModel
  - Allows sharing of ViewModel between Views
  - ViewModel can outlive the View

# ViewModel Locator as Resource

```
<Application x:Class="CommandingWithMvvm.App"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:vm="clr-namespace:CommandingWithMvvm.ViewModels">
  <Application.Resources>
    <ResourceDictionary>
      <vm:ViewModelLocator x:Key="VMLocator"/>
      <ResourceDictionary.MergedDictionaries>
        <ResourceDictionary Source="Assets/Styles.xaml"/>
      </ResourceDictionary.MergedDictionaries>
    </ResourceDictionary>
  </Application.Resources>
</Application>
```

# Getting the ViewModel

- ViewModel set as DataContext in View, through the Resource:ViewModel Locator

```
<navigation:Page x:Class="CommandingWithMvvm.Binding1"  
    DataContext="{Binding Binding1.ViewModel, Source={StaticResource  
    VMLocator}}"
```

- ViewModel Locator exposes the ViewModel as a property

```
public Binding1.ViewModel Binding1.ViewModel
```

# What is Commanding?

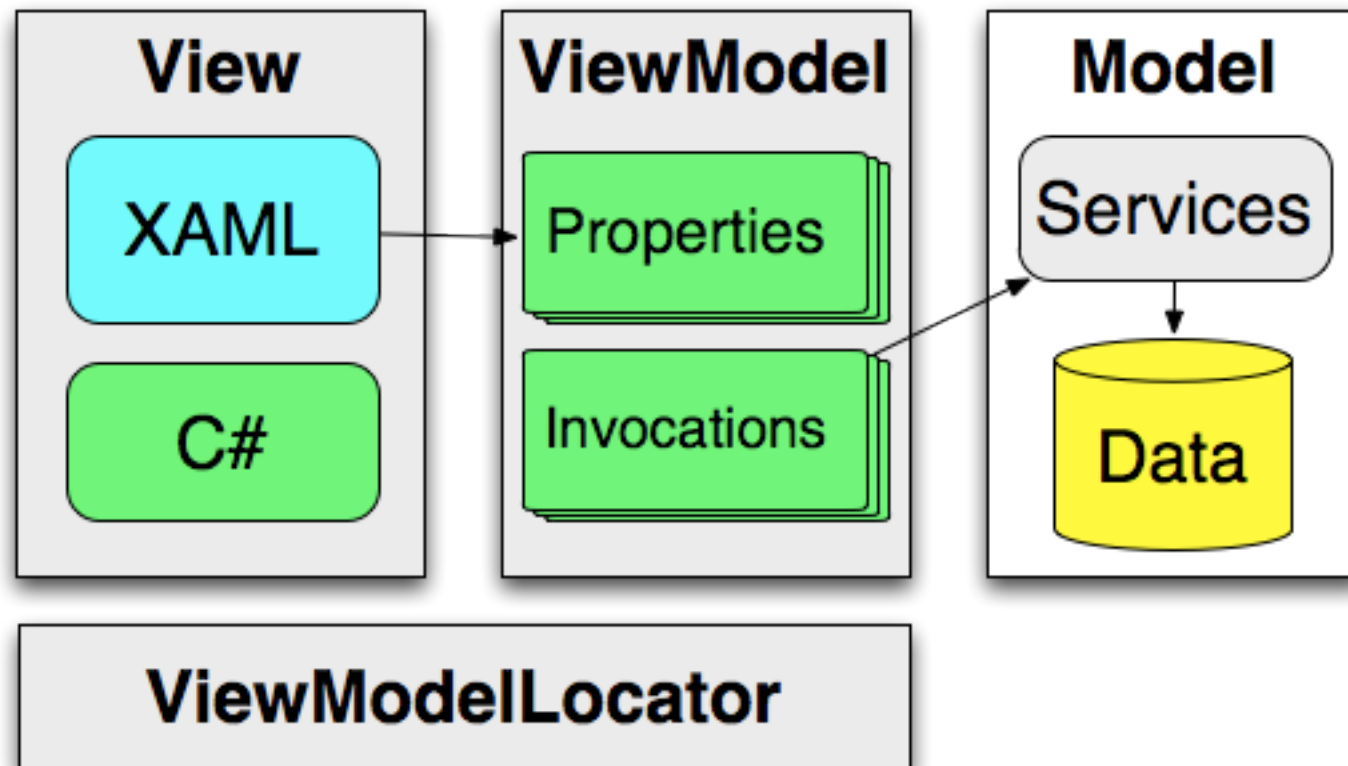
- Button bound to Property (ICommand) on DataContext
- No code-behind on view is required
- Mouse click invokes bound command
- **XAML:**

```
<Button Content="Bound to ViewModel Command"  
Command="{Binding IncrementCount}" />
```

- *IncrementCount is an property (ICommand) on the DataContext*

# What is Commanding?

- ViewModel invokes the service via Command



# The crux of Commanding

- ViewModel has logic to communicate with model
- ViewModel exposes commands that can be invoked
- View binds commands to UI Elements for the user to invoke

# ICommand Interface

- Recently added in Silverlight 4
- Same contract as WPF

```
public interface ICommand
{
    event EventHandler CanExecuteChanged;
    bool CanExecute(object parameter);
    void Execute(object parameter);
}
```

# RelayCommand

- MVVM Light - <http://mvvmlight.codeplex.com>
- Implementation of ICommand that relays functionality through lambdas

```
public class RelayCommand : ICommand
{
    public RelayCommand(Action execute, Func<bool> canExecute);
    public void RaiseCanExecuteChanged();
    public event EventHandler CanExecuteChanged;
    public bool CanExecute(object parameter);
    public void Execute(object parameter);
}
```



# RelayCommand: Example

- Create Action to call method
- Create Func as predicate
- Create Command with delegates in constructor

```
private void InitializeCommands()
{
    // Action that command will perform
    Action incrementAction = () => PerformIncrementCount();
    // Predicate for whether or not action is allowed to execute
    Func<bool> incrementPredicate = () => Count >= 0;

    // Set Command property to new RelayCommand Object
    IncrementCount = new RelayCommand(incrementAction, incrementPredicate);
}
```

# Demo

- Increment Number on the screen
- No Ria Services
- No Network Access
- Simple example to demonstrate:
  - commands
  - viewmodels

# Demo 1: No ViewModel

- update values using c#
- View contains Variables
- View contains Logic
- Logic executed via event handlers

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    //Update private variable
    count++;
    //Refresh Text Box from private value
    SetTextBoxValue();
}
```

# Demo 2: ViewModel

- ViewModel assigned through ViewModel Locator
- Textbox bound to property on ViewModel
- 2 parts:
  - Triggers
  - Commanding

# Demo 2a: Triggers & Actions

- Mimicks code behind, but in XAML
- Requires Expression blend 4 SDK
  - Microsoft.Expression.Interactions.dll
  - System.Windows.Interactivity.dll

```
<Button Content="Behavior Button" >
  <i:Interaction.Triggers>
    <i:EventTrigger EventName="Click">
      <si:CallMethodAction MethodName="PerformIncrementCount"
        TargetObject="{Binding}" />
    </i:EventTrigger>
  </i:Interaction.Triggers>
</Button>
```

# Demo 2b: MvvmLight Commanding

- Button bound to command on viewmodel

```
<Button Content="Bound to ViewModel Command"  
Command="{Binding IncrementCount}" />
```

# Demo 3: Unit Testing

- Install Silverlight Toolkit April 2010
- Silverlight Unit Testing Framework
- All code executed on SL client
- Logic on ViewModel can be tested without UI

# OData - Open Data Protocol

- producers:

SQL Azure	eBay	Facebook
Stack Overflow	Netflix	custom

- client apis:

Javascript	PHP	Java
.Net	Silverlight / WP7	iPhone

- OData Application Support:

Excel	LINQPad	Java
-------	---------	------



# LINQPad

- Written by: Joseph Albahari.
- Query Datasources using LINQ, without visual studio
  - Will auto-create db context for your li n against.
- Execute any c# / vb.net code as script
- Easy to use
- FREE to use!
  - C# Intellisense for (\$39)



# Demo 4: Odata & Silverlight

- Create a linq query in LINQPad
- Wire up ViewModel command to call query

```
from t in Posts  
where t.Tags.Contains("Silverlight") && t.Score > 30  
select t
```

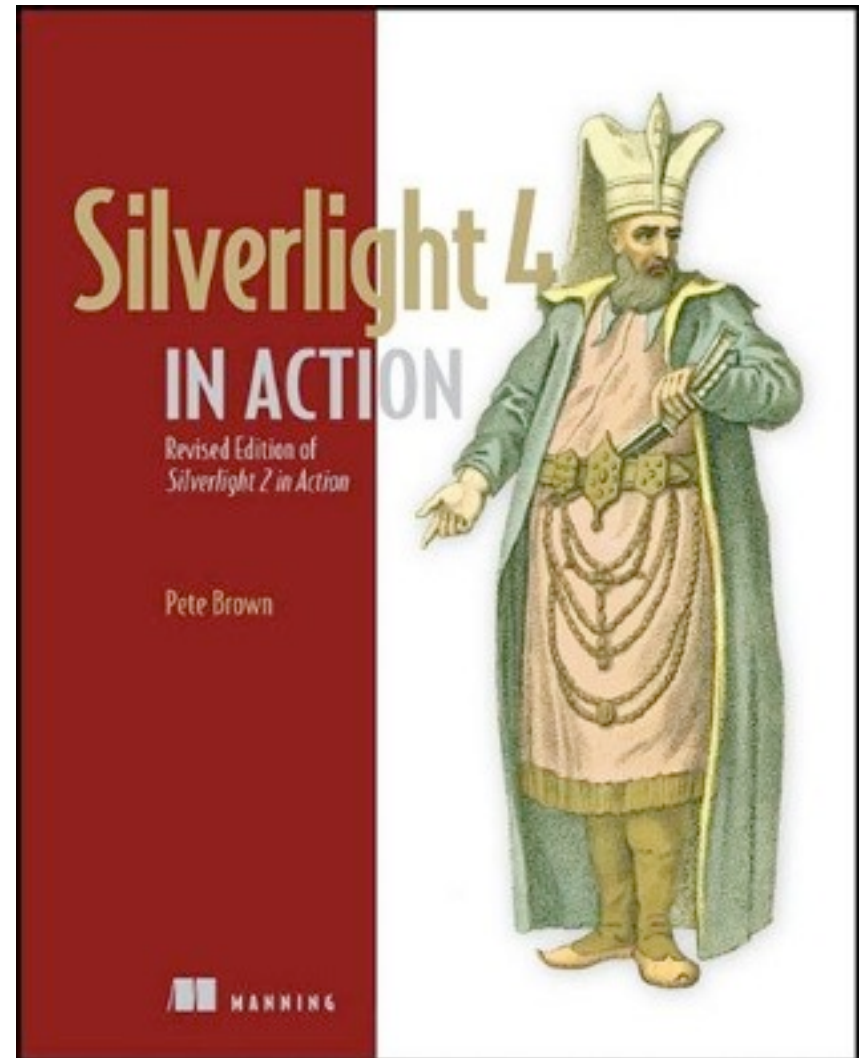
- Search box is bound to Query Command Parameter


# Summary

- Mvvm allows separation of logic & model activity from view
- Commanding & Behaviours allow for XAML invocation of ViewModel Code
- Commanding exposes bindable properties
- ViewModels can be unit tested

# Q & A

- Twitter #vanslug
  - Forums
  - Live
- 
- Silverlight 4 in action



- 
- Thank you for your involvement!
  - Please put your questions & feedback on the forum:
  - <http://forum.vanslug.net>

# *Vanslug*

- Thank you for your involvement!
- Please put your questions & feedback on the forum:
- <http://forum.vanslug.net>