

Rachel Lim's Blog

Simplifying programming into something I can understand

Navigation with MVVM

When I first started out with MVVM, I was lost about how you should navigate between pages. I'm a firm believer in using ViewModels to do everything (unless it's View-specific code), and that the UI is simply a user-friendly interface for your ViewModels. I did not want to create a button on a page that has any kind of code-behind to switch pages, and I didn't like the idea of my navigation being spread out throughout all the ViewModels.

I finally came to realize the solution was simple: I needed a ViewModel for the Application itself, which contained the application state, such as the `CurrentPage`.

Here is an example that builds on the [Simple MVVM Example](https://rachel53461.wordpress.com/2011/05/08/simplemvvmexample/) (<https://rachel53461.wordpress.com/2011/05/08/simplemvvmexample/>).

The ViewModel

Usually I name the ViewModel `ApplicationViewModel` or `ShellViewModel`, but you can call it whatever you want. It is the startup page of the application, and it is usually the only page or window object in my project.

It usually contains

```
List<ViewModelBase> PageViewModels
ViewModelBase CurrentPage
ICommand ChangePageCommand
```

Here is an example `ApplicationViewModel` that I would use to go with the [Simple MVVM Example](https://rachel53461.wordpress.com/2011/05/08/simplemvvmexample/) (<https://rachel53461.wordpress.com/2011/05/08/simplemvvmexample/>).

```
public class ApplicationViewModel : ObservableObject
{
    #region Fields

    private ICommand _changePageCommand;

    private IPageViewModel _currentPageViewModel;
    private List<IPageViewModel> _pageViewModels;
```

```
#endregion
```

```
public ApplicationViewModel()  
{  
    // Add available pages  
    PageViewModels.Add(new HomeViewModel());  
    PageViewModels.Add(new ProductsViewModel());  
  
    // Set starting page  
    CurrentPageViewModel = PageViewModels[0];  
}
```

```
#region Properties / Commands
```

```
public ICommand ChangePageCommand  
{  
    get  
    {  
        if (_changePageCommand == null)  
        {  
            _changePageCommand = new RelayCommand(  
                p => ChangeViewModel((IPageViewModel)p),  
                p => p is IPageViewModel);  
        }  
  
        return _changePageCommand;  
    }  
}  
  
public List<IPageViewModel> PageViewModels  
{  
    get  
    {  
        if (_pageViewModels == null)  
            _pageViewModels = new List<IPageViewModel>();  
  
        return _pageViewModels;  
    }  
}  
  
public IPageViewModel CurrentPageViewModel  
{  
    get  
    {  
        return _currentPageViewModel;  
    }  
    set  
    {  
        if (_currentPageViewModel != value)  
        {  
            _currentPageViewModel = value;  
            OnPropertyChanged("CurrentPageViewModel");  
        }  
    }  
}
```

```

#endregion

#region Methods

private void ChangeViewModel(IPageViewModel viewModel)
{
    if (!PageViewModels.Contains(viewModel))
        PageViewModels.Add(viewModel);

    CurrentPageViewModel = PageViewModels
        .FirstOrDefault(vm => vm == viewModel);
}

#endregion
}

```

This won't compile right away because I've made some changes to it. For one, all my PageViewModels now inherit from an IPageViewModel interface so they can have some common properties, such as a Name.

I also created a new HomeViewModel and HomeView since its hard to demonstrate navigation unless you have at least 2 pages. The HomeViewModel is a blank class that inherits from IPageViewModel, and the HomeView is just a blank UserControl.

In addition, I added an s to ProductsViewModel since it really deals with multiple products, not a single one.

An added advantage to having a ViewModel to control the application state is that it can also be used to handle other application-wide objects, such as Current User, or Error Messages.

The View

I also need an ApplicationView for my ApplicationViewModel. It needs to contain some kind of Navigation that shows the list of PageViewModels, and clicking on a PageViewModel should execute the ChangePage command.

It also needs to contain a control to display the CurrentPage property, and I usually use a ContentControl for that. This allows me to use DataTemplates to tell WPF how to draw each IPageViewModel.

```

1  <Window x:Class="SimpleMVVMExample.ApplicationView"
2      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" (http://schemas
3      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" (http://schemas
4      xmlns:local="clr-namespace:SimpleMVVMExample"
5      Title="Simple MVVM Example" Height="350" Width="525">
6
7      <Window.Resources>
8          <DataTemplate DataType="{x:Type local:HomeViewModel}">
9              <local:HomeView />
10         </DataTemplate>
11         <DataTemplate DataType="{x:Type local:ProductsViewModel}">
12             <local:ProductsView />
13         </DataTemplate>
14     </Window.Resources>
15
16     <DockPanel>
17         <Border DockPanel.Dock="Left" BorderBrush="Black" BorderThickness="0,0,0,0">
18             <ItemsControl ItemsSource="{Binding PageViewModels}">
19                 <ItemsControl.ItemTemplate>
20                     <DataTemplate>
21                         <Button Content="{Binding Name}"
22                             Command="{Binding DataContext.ChangePageCommand, RelativeSource={RelativeSource FindAncestor, AncestorType=Window, AncestorLevel=1}}"
23                             CommandParameter="{Binding}"
24                             Margin="2,5"/>
25                     </DataTemplate>
26                 </ItemsControl.ItemTemplate>
27             </ItemsControl>
28         </Border>
29
30         <ContentControl Content="{Binding CurrentPageViewModel}" />
31     </DockPanel>
32 </Window>

```

In this example, I'm using an [ItemsControl](https://rachel53461.wordpress.com/2011/09/17/wpfx-itemscontrol-example/) (<https://rachel53461.wordpress.com/2011/09/17/wpfx-itemscontrol-example/>) to display my PageViewModels. Each item is drawn using a Button, and the Button's Command property is bound to the ChangePageCommand.

Since the Button's DataContext is the PageViewModel, I used a RelativeSource binding to find the ChangePageCommand. I know that my Window is the ApplicationView, and it's DataContext is the ApplicationViewModel, so this binding looks up the VisualTree for the Window tag, and gets bound to Window.DataContext.ChangePageCommand.

Also note that I am putting DataTemplates in Window.Resources to tell WPF how to draw each IPageViewModel. By default, if WPF encounters an object in it's visual tree that it doesn't know how to handle, it will draw it using a TextBlock containing the .ToString() method of the object. By defining a DataTemplate, I am telling WPF to use a specific template instead of defaulting to a TextBlock.

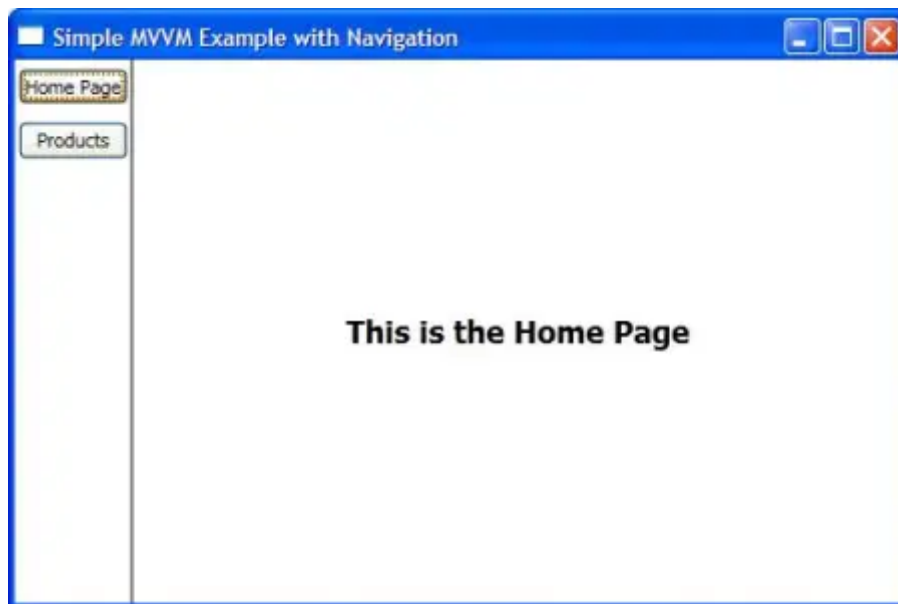
If you are continuing from the Simple MVVM Example, I moved the ProductView out of a ResourceDictionary and into a UserControl to make this simpler.

Starting the Example

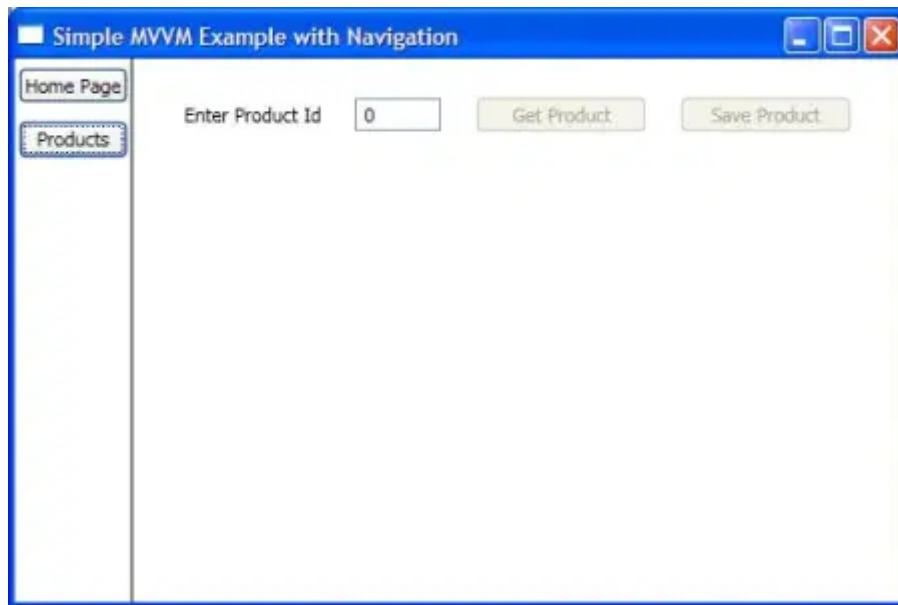
The last thing to do is change App.xaml to make ApplicationView and ApplicationViewModel our startup, instead of ProductView/ProductViewModel.

```
1 public partial class App : Application
2 {
3     protected override void OnStartup(StartupEventArgs e)
4     {
5         base.OnStartup(e);
6
7         ApplicationView app = new ApplicationView();
8         ApplicationViewModel context = new ApplicationViewModel();
9         app.DataContext = context;
10        app.Show();
11    }
12 }
```

Run the project and you should see something that looks like the images below, which quickly switches the CurrentPage when clicking on the Navigation buttons.



[Home \(https://rachel53461.files.wordpress.com/2011/12/home.jpg\)](https://rachel53461.files.wordpress.com/2011/12/home.jpg)



[Products \(https://rachel53461.files.wordpress.com/2011/12/products.jpg\)](https://rachel53461.files.wordpress.com/2011/12/products.jpg).

Summary

And there you have it. A simple navigation example with MVVM.

You can download the source code for this sample from [here \(http://www.mediafire.com/?21g3er1ant6m99a\)](http://www.mediafire.com/?21g3er1ant6m99a).

Once you get more comfortable with WPF, I would recommend looking into using a Messaging System, such as [MVVM Light's Messenger \(http://www.galasoft.ch/mvvm/\)](http://www.galasoft.ch/mvvm/), or [Microsoft Prism's EventAggregator \(http://msdn.microsoft.com/en-us/library/ff921122\(v=pandp.20\).aspx\)](http://msdn.microsoft.com/en-us/library/ff921122(v=pandp.20).aspx) to broadcast `ChangePage` commands from any `ViewModel` so you wouldn't need to find the `ApplicationViewModel` to execute the `ChangePageCommand`, however that's for another day.

[<< Back – A Simple MVVM Example](http://rachel53461.wordpress.com/2011/05/08/simplemvvmexample/)

[\(http://rachel53461.wordpress.com/2011/05/08/simplemvvmexample/\)](http://rachel53461.wordpress.com/2011/05/08/simplemvvmexample/).

[>> Next – Communication between ViewModels](http://rachel53461.wordpress.com/2011/06/05/communication-between-viewmodels-with-mvvm/)

[\(http://rachel53461.wordpress.com/2011/06/05/communication-between-viewmodels-with-mvvm/\)](http://rachel53461.wordpress.com/2011/06/05/communication-between-viewmodels-with-mvvm/).

This entry was posted on Sunday, December 18th, 2011 at 3:57 pm and is filed under [MVVM](#), [WPF](#). You can follow any responses to this entry through the [RSS 2.0](#) feed. You can [leave a response](#), or [trackback](#) from your own site.

110 Responses to *Navigation with MVVM*

A Grateful Programmer says:

December 30, 2018 at 6:43 am

Um...wow...I've been pulling my hair out trying to figure this out. I've just started WPF and was struggling with how to create the Master/Detail design. Whenever I clicked a menu item, only the viewmodel's ToString() would show. So, it would show something like "ViewModels.ViewAViewModel" or something like that. I stumbled across some answers of yours on S.O. and followed this blog via there. It looks like a wealth of knowledge. Thank you for taking the time to do this. It really truly helps and I greatly appreciate your hard work and effort.

Reply

estebanqv says:

November 28, 2016 at 4:30 pm

Hello Rachel,

I'm in my first MVVM steps. I followed your Navigation tutorial. I have a small question:

How can I integrate a Login Window in this navigation sample, knowing that after the user is logged the options in navigation will change?

Thank you for your answer

Reply

Rachel says:

February 22, 2017 at 3:30 pm

There's multiple ways to do this. Here's one example. Best way to figure it out is just try it, and if you get stuck post a question on Stack Overflow with your code, expected result, and actual result



Reply

Dios says:

August 19, 2016 at 1:11 pm

Hello Rachel. Excellent article and blog. Two questions,

Can the ViewModels list inside ApplicationViewModel be avoided? Because I think a huge list with concrete ViewModels will become a pain for a developer if the project become more complex.

Also do you still implementing your projects with that way?

Reply

Rachel says:

February 22, 2017 at 3:36 pm

Hi Dios, sadly my current job does not have me working with WPF, so I haven't used it much over the past few years. Yes there is ways around that, but I wanted to keep this very simple for the purpose of this blog post. If you look around on Google or Stack Overflow, I'm sure you'd find some more advanced examples 😊

Reply

Roy says:

June 7, 2016 at 1:10 pm

Thanks for this post. I could figure out the way to work with the approach shared (first one). This is helpful. Good luck with the future posts.

Reply

Vijay says:

May 23, 2016 at 7:31 am

Could you provide complete code or project for Navigation with MVVM

Reply

Rachel says:

June 10, 2016 at 2:24 pm

Hi Vijay, the code should be linked at the end if you're looking to download a copy

Reply

Mayola Pizzulo says:

May 16, 2016 at 3:09 am

Very nice post. I simply stumbled upon your blog and wished to say that I have really loved surfing around your weblog posts. After all I'll be subscribing in your feed and I hope you write again soon! |

Reply

Jan Ivar Carlsen says:

January 16, 2016 at 4:49 pm

Thank you, this was very helpful!

Reply

www says:

December 18, 2015 at 6:14 am

Thank you very much!

Reply

Peter Dunn says:

October 29, 2015 at 5:41 am

Thank you for time and clear example. I highlighted the selected button with a Style when selecting (code below), but i need help keeping the button highlighted while I work in the selected UserControl.

Thanks for any help,

Peter

Reply

Rachel says:

November 27, 2015 at 5:02 pm

Hi Peter, try posting a question on stackoverflow.com with the relevant bits of code and someone can probably help you out. If not, just post the question link here and I'll try to take a look at it when I get time.

Reply

eavestn says:

October 22, 2015 at 8:52 pm

Rachel, is there anyway to use your method of updating the VM and pulling the .xaml DataTemplate configurations out into the App.xaml?

Reply

eavestn says:

October 22, 2015 at 9:06 pm

Also, did you ever write anything on using a messaging service?

Reply

Rachel says:

November 27, 2015 at 4:59 pm

Yes, I wrote Communication between View Models with MVVM which is just a basic overview of messaging systems, and Simplifying PRISM's EventAggregator which was a helper class that was easy to use for smaller projects.

Rachel says:

November 27, 2015 at 4:58 pm

I don't see why you couldn't do that 😊 If you need help with it, I'd recommend posting a question on stackoverflow.com with the relevant bits of code, and it will probably get answered quickly. If not, just send me the link and I'll see if I can take a look at it sometime

Reply

Anne Jan Pool says:

October 14, 2015 at 7:37 pm

We have several applications all working according to your tutorial using the ViewModelLocator of MVVM Light and the Messenger functionality of MVVM Light. Works wonderful (thanks for the clear explanation).

Each ViewModel is related to one View. But every time the user navigates to a page (View) a new instance of the View is created (the ViewModels are singletons, constantly listening to Messages) and the old one is not garbage collected. So memory gets fuller and fuller and the application slows down. How can I make a View be garbage collected after it has been unloaded?

I really hope that you can help me.

Reply

Rachel says:

October 22, 2015 at 3:25 pm

Hi Anne,

I'm not a big fan of the ViewModelLocator, so I'm not sure if I could help you there.

My best guess is to double-check any event handlers to make sure nothing is holding the View in memory, or see if you can draw the View using DataTemplates rather than new instances. Its hard to tell though with my limited knowledge of ViewModelLocators and without seeing your code.

I'd recommend posting a question on StackOverflow with some sample code to demonstrate the problem, and someone will probably be able to help you there. Feel free to post a link to the question here too, and perhaps I can take a look at it.

Good luck!

Reply

sd says:

September 17, 2015 at 12:25 pm

Thank you ms Rachel you for this simple easy to understand tutorial! I kindly ask you this:

What would be the approach to have a “Home page” button also *inside* the products page (imagine it just below the “get product” button) which of course will navigate to the home page?

Thank you, keep the good work! 😊

Reply.

sd says:

September 17, 2015 at 12:35 pm

hmm just noticed the last paragraph “I would recommend looking into using a Messaging System, such as MVVM Light’s Messenger, or Microsoft Prism’s EventAggregator to broadcast ChangePage commands from any ViewModel”

as i am learning MVVM i would like to avoid ready made frameworks so i can really get the knowledge of what is happening (and then later i can use frameworks). So I just would be glad if you could provide any tips on broadcasting these type of events/commands without using framework. (*gets ready for google-fu*)

Thank you again! 😊

Reply.

Rachel says:

September 18, 2015 at 6:37 pm

I’m sorry to say that I never bothered to make my own, so am not sure if I could help direct you in the right direction. I know there are many variations of such messaging frameworks, some Microsoft ones and some 3rd party, although the only ones I’ve used is MVVM Light’s Messenger, Prism’s EventAggregator, and CAB’s [EventSubscription] and [EventPublication] attributes.

Rachel says:

September 18, 2015 at 6:46 pm

In this case, I would recommend using an event broadcast system like MVVM Light’s Messenger or Prism’s EventAggregator. Alternatively, your ApplicationViewModel could set something like a ProductViewModel.HomeCommandDelegate at the time it is created

Reply.

RodAtHome says:

September 10, 2015 at 1:33 am

Reblogged this on Doctor Who and commented:

This is an excellent series of articles by Rachel on the MVVM programming pattern. Well worth reading.

Reply.

RodAtHome says:

September 9, 2015 at 8:32 pm

Very good article on the MVVM design pattern and navigation within a WPF app. Thank you for posting!

I’ve got one question; where is the definition for the IPageViewModel interface?

Reply.

Rachel says:

September 9, 2015 at 9:39 pm

Thanks! And the IPageViewModel interface should be in the sample code in the same folder as the .csproj. Make sure you have this version though, and not the version from the previous article 😊

Reply

RodAtHome says:

September 10, 2015 at 1:30 am

OK, thank you!

Franz says:

August 24, 2015 at 9:49 am

Hello,

I tried to add some Views. The buttons are shown but when I click on one of them I only get the Name of my ViewModel class? How can I connect my ViewModel with the correct View? And why is it working for 3 out of 4?

Thank you in advance!

Reply

Rachel says:

August 24, 2015 at 1:45 pm

Hi Franz,

Most likely you do not have a DataTemplate defined your your ViewModel's object type, WPF is rendering the object using the default template which is just text containing the .ToString() of the object.

If you add your DataTemplates it should work correctly.

Good luck!

Rachel

Reply

Samambaia Jr. says:

June 25, 2015 at 9:44 pm

Hi, I'm a absolute begginer in MVVM and I would like to know about send data between UserControl. The both UC are on the same page. Something like List and Detail. I would like to click on a list item and show the result in the UserControl detail. How do I do? You could give me a tip or an example?

Reply

Rachel says:

August 24, 2015 at 1:59 pm

Hi Samambaia,

In your example you'd ideally have a single ViewModel with both a Collection of objects, and a SelectedObject property, and each UserControl would get bound to those. If the two objects are in completely separate areas of the application and it would be difficult to have them share the same

parent object model, I'd recommend looking into WPF's event system, such as MVVM Light's Messenger object or Microsoft Prism's EventAggregator.

Thanks,
Rachel

Reply

lam says:

May 29, 2015 at 7:24 pm

Hi Rachel,

Thanks for showing this useful article which is similar to what I am trying to achieve currently: use a group of radio buttons (command binding) to switch views with MVVM. However, I failed to get it work. Is it possible for you to illustrate with your sample code to show how it can be done? Thanks.

Reply

Rachel says:

June 18, 2015 at 9:15 pm

Hi lam,

I would suggest using a ListBox that has its style overwritten to use RadioButtons so it keeps track of the selected item easily. Then you could bind the ItemsSource to your AvailableScreens, and bind the SelectedItem to whatever screen is currently selected.

If you need help with such a solution, try posting a question containing the details on <http://www.stackoverflow.com>, and leave a link here for me and I'll take a look (assuming someone else doesn't help you out first!).

Good luck with it!

Reply

carpi1968 says:

February 6, 2015 at 2:12 am

Hi Rachel,

Wonderful post!

Would there be a simple way to have the old view fading out and the new view fading in?

Any suggestion would be highly appreciated

Reply

Rachel says:

August 24, 2015 at 1:56 pm

You'll probably want to look into WPF Animations, and animate the Opacity property. Good luck!

Reply

avots says:

November 18, 2014 at 10:53 am

Hi, unfortunately the code can not be downloaded for some reason..

The message I get is something like "You connected to wired.meraki.com, but you are likely not currently connected to a Cisco Meraki appliance."

Can you please check this?

Thanks!

Reply

Rachel says:

November 26, 2014 at 4:17 pm

Hi Avots, I'm not sure why it isn't working. I just double-checked the link and it seems OK, so perhaps it's an issue with your internet connection? I don't know what Meraki is, but it sounds like something specific to your network.

Reply

esdee says:

October 13, 2014 at 7:39 pm

excellent mvvm series

thanks a lot!

Reply

Woj says:

October 13, 2014 at 5:20 pm

Thanks for the awesome example. How would one go about doing the view navigation using MVVM Light Messenger class??

Reply

Rachel says:

October 13, 2014 at 9:02 pm

Hi Woj,

Usually I'd have my ViewModels raise an event (Messenger.Default.Send) whenever it wants to change the current screen, and my main AppViewModel would subscribe to receive these messages and change the currentContent whenever it receives one.

Reply

Woj says:

October 15, 2014 at 7:29 pm

So would the Send event need to be in the constructors of all my viewmodels and the register event in the constructor of the main app view model? Or would the Register event need to be in the ChangePageCommand?

Rachel says:

October 28, 2014 at 1:08 pm

Its been a while since I've used MVVM Light, however I believe it uses a singleton instance of the Messenger class, so wherever you want to change pages you'd call something like Messenger.Default.Send(...) and pass it whatever event args you want. The Register event only needs to be hooked up in whatever class is in charge of actually changing the view, such as your main app view model.

dancllick says:

September 18, 2014 at 7:04 pm

Hi Rachel, great articles, thanks!

One question: I wanted to remove the "button sidebar" from the ApplicationView Windows. So, I implemented the navigation, as follows:

-Instead having a collection of viewmodel, when application starts I set the ApplicationWiew's Data Context to a new instance of my first view, passing to the constructor the ApplicationViewModel itself:

```
CurrentPageViewModel = new HomeViewModel(this)
```

so I'm injecting the ApplicationViewModel on another viewmodel.

-To perform navigation I created a Command that sets the CurrentPageViewModel to a new instance of another viewmodel:

```
_applicationViewModel.ChangeViewModel(new AboutViewModel());
```

Is this a good way to accomplish navigation in this scenario?

Thanks!

Daniele

Reply

Rachel says:

September 19, 2014 at 7:31 pm

Hi Daniele, I don't see any problem with using that approach if that is what works with your application style.

A few things to be careful of though is to ensure it makes sense for your ViewModels to know of each other in this way, and that's its OK to create a new ViewModel every time that method is called.

I've always been a big fan of being practical first and foremost, so if this works for you and is practical for your framework, go for it!

Rachel

Reply

BenjaminBenjamin Gale says:

September 13, 2014 at 12:06 pm

Hi Rachel,

I notice you are adding the page viewModels in your ApplicationViewModel. I can see some issues with this:

1. The ApplicationViewModel knows about all of the concrete sub-viewModels.
2. If the sub-ViewModels have a lot of properties or contain their own sub-viewModels then there could be a lot of viewModels open at application launch increasing the resource usage of the application.

Take unit testing, for example. If I was testing the ApplicationViewModel, the entire object graph, including the HomeViewModel and ProductsViewModel would be in memory. In a real world application you could have 10+ pages, each of which could have their own pages (and so on). This seems a bit heavy handed where at most I should have the AppViewModel and any mocked dependencies in memory during the test of the ApplicationViewModel.

Also, what happens if the viewModels being used as pages have their own dependencies? Who is responsible for resolving these? and if the page viewModels also have their own set of page viewModels, each with their own dependencies...

I love the simplicity and elegance of using this approach but in my use of it I have come across the issues detailed above. I'm not currently sure of the best way of overcoming these hurdles either.

Obviously this is an example, so I understand why the article does not address these points but I was wondering if you have come across these problems in any of the WPF applications you have written and if so what solutions you would recommend.

Reply

Rachel says:

September 19, 2014 at 7:37 pm

Hi Benjamin,

A typical solution I see is the use of a NavigationService that is responsible for handling the creation and management of ViewModels and their associated Views.

Typically it only creates a ViewModel when it's requested for display the first time, and can handle other cases depending on your requirements, such as memory management, ViewModels that go out of scope, etc.

In the past I will also typically use of an event system too, such as PRISMs EventAggregator or MVVM Light's Messenger class, so the ViewModels don't need to have a direct reference to each other.

Hope that helps answer your questions!

Rachel

Reply

Franz says:

August 2, 2014 at 10:07 pm

Hi Rachel,

Thank you for this beautifully written article which I am so glad I stumbled upon. I think that this is a good model for an application that I am currently converting, as a learning experience, from Delphi 7/VCL to F#/WPF/MVVM.

Your article has helped me better understand these new technologies. I am pretty sure that I will refactor the app in its present state, with one main window view/VM and two (independent) tabs, into multiple (ultimately three) views/VM's as per your design since that suits so much better (from both a coding and usage perspective).

Regards,
Franz

Reply

Michael Aliotti says:

July 3, 2014 at 1:42 pm

Hi Rachel. Great blog post! I do have some questions, though:

1) How would you handle passing parameters from one ViewModel to the next? In other words, a command is triggered on ViewModel1 and the application switches to ViewModel2, but ViewModel2 needs a parameter from ViewModel1.

2) When instantiating the list of ViewModels, what would be the best approach for ViewModels that rely on parameters from other ViewModels? Should they be added to the list with default parameters? Could the default parameters be changed when the ViewModel they depend on passes the parameters to them?

Any help is much appreciated!

Reply

Rachel says:

July 3, 2014 at 2:52 pm

Hi Michael,

Typically we would use some form of messaging system to communicate between ViewModels. I'd recommend Microsoft PRISM's EventAggregator or MVVM Light's Messenger, or if you really wanted to you could also build your own instead.

I have a brief overview of them here if you're interested : [Communication Between ViewModels with MVVM](#)

Good luck with your project!

Rachel

Reply

Michael Aliotti says:

July 3, 2014 at 3:07 pm

Thanks Rachel! I will check out your recommendations.

peter says:

May 28, 2014 at 8:38 pm

I noticed that in your ApplicationViewModel you have a list of ViewModels, which is composed by a new instance of each ViewModel. Can this method lead to performance issues or is this the standard way?

Thank You

Reply

Rachel says:

June 10, 2014 at 1:36 pm

I'm sure it could, and you could lazy-load the ViewModels as needed instead of creating them all initially. It depends on how big your ViewModels are 😊

Reply

Syn says:

March 3, 2014 at 2:32 pm

This is an excellent article, thank you.

Reply

Deno says:

January 18, 2014 at 6:13 pm

How does this app gets the correct View by the ViewModel? Where is it stated that the HomeViewModel is connected with HomeView?

I added for example an another ViewModel to the PageViewModels list, and it automatically adds the button with the correct newly created custom View... I just don't get it, how is that possible? I've never stated that this new ViewModel is connected with the new View.

```
Command="{Binding DataContext.ChangePageCommand, RelativeSource={RelativeSource  
AncestorType={x:Type Window}}}"  
CommandParameter="{Binding}"
```

Is it because of this? What's Relative AncestorType={x:Type Windows} doing? And why is CommandParameter binding to nothing?

C# and MVVM looks really fun way of programming, but it's pretty abstract technique for me...

Reply

Rachel says:

January 19, 2014 at 6:17 pm

Hi Deno,

The ContentControl has it's Content property bound to the CurrentPageViewModel. By default, WPF doesn't know how to draw a ViewModel, however I have defined a DataTemplate in the .Resources of the application to tell WPF that it should draw anything of type HomeViewModel with HomeView, and ProductsViewModel with ProductsView.

The Command binding only ties the button click event with the ChangePageCommand in the ViewModel. Because of the way the DataContext behind the UI controls gets inherited, we need to use a RelativeSource binding to tell it to look in Window.DataContext for the ChangePageCommand. The empty binding is simply a shortcut way to refer to the current DataContext, so it is passing in the selected IPageViewModel to that command.

If you're new to WPF, I'd suggest reading my articles [Understanding the change in mindset when moving from Winforms to WPF](#) and [What is this "DataContext" you speak of?](#). I think they may help answer some of your questions related to bindings, the DataContext, and MVVM.

Reply

Deno says:

January 21, 2014 at 12:21 pm

Wow thanks for you quick reply.. and this blogs contains very useful information, I didn't knew about that huge basic WPF article. I've improved a lot since I asked that question but I'm still a beginner, so I'm still going to look into it... thank you!

rajesh says:

November 11, 2014 at 12:40 pm

Hi Rachel,

You have mentioned in the article that "Button's DataContext is the PageViewModel, so you used a RelativeSource binding to find the ChangePageCommand".

I am struggling to understand how it is that Button's DataContext is PageViewModel? Can you please shed some light on it.

I have read your article explaining DataContext, from which I can infer that if a DataContext is not set for an element, it inherits it from it's Parent element. Here I see that DataContext is not specified for any of the element except for Window element(specified in App.xaml.cs) whose DataContext is ApplicationViewModel. So, for me Button's Datacontext should be ApplicationViewModel.

What am I thinking wrong here?

Rachel says:

November 13, 2014 at 4:05 pm

Hi Rajesh, to understand the DataContext inheritance here you need to understand how an ItemsControl works.

An ItemsControl will take each item in the ItemsSource, create a copy of the UI objects in the ItemsPanelTemplate, and assign the DataContext of those UI objects equal to the item in the ItemsSource. You can read more about the ItemsControl [here](#) if you're interested.

So in this case, the ItemsControl is bound to a collection of PageViewModels, so it will create an instance of the ItemsPanelTemplate for each PageViewModel and assign the DataContext equal to PageViewModels[n] on each one.

Hope that helps,
Rachel

Walter says:

September 20, 2013 at 8:01 am

Thanks for this example,

I never used my vm for navigation before, and have a question about the ApplicationView.

I like to use (restyle) radiobuttons for switching views. This way the user can see which view is selected. How about setting the right radiobutton to IsChecked when changing the view in the viewmodel.

Walter

Reply

Rachel says:

September 21, 2013 at 5:01 pm

Hi Walter,

Typically for something like that where I want to track the SelectedItem, I would use a ListBox that is styled to use RadioButtons. You can find an example [here](#).

Good luck with it,
Rachel

Reply

Farid says:

June 4, 2013 at 8:29 pm

Hi Rachel,

Thanks a lot. It's exactly what i've been searching during this week!
Now I would like to use builtin Ribbon control to load my viewmodels.
I've tried to modify your source to do that but my Ribbon button is always disabled???

Reply

Rachel says:

June 4, 2013 at 8:59 pm

Hi Farid, you might want to post a question on StackOverflow about it, along with a small sample of your source code.

I haven't worked with the Ribbon control before, and wouldn't be of much help without seeing your code.

Thanks,
Rachel

Reply

Farid says:

June 4, 2013 at 9:27 pm

Thanks!

Naveed Quadri says:

May 18, 2013 at 12:02 pm

Hi Rachel,

I am (WPF Newbie) working on a navigation based application, where i have abt 10 views. I am trying to understand why are we not using the NavigationService built in WPF.

I would really appreciate it if you could tell me or point to a resource on this topic.

Thanks

Naveed

Reply

Rachel says:

June 15, 2013 at 4:20 pm

Hi Naveed,

Honestly, I have not used the NavigationService before. I can't remember my exact reasons for it, but I believe it was because I thought it mixed the data and UI layers too much.

I always found it easier to let my application (ViewModels) control the application flow, and leave the UI side of things out of the application layers entirely.

For me, my UI layer is simply a visual representation of my application layer, and I didn't like having my application layer worry about how to draw each page.

Reply

Mitesh says:

April 10, 2013 at 12:20 am

Hi!

My question is related to following article where you posted your answer.

Ref: <http://stackoverflow.com/questions/7446825/open-child-window-inside-a-parent-window-in-wpf-using-mvvm>

I am using MVVM framework and the technique you have mentioned in your following post

<https://rachel53461.wordpress.com/2011/12/18/navigation-with-mvvm-2/>

This is the same technique I am using to implement my WPF application. I am building new WPF application and want to use pure MVVM i.e. no code behind.

Now I want to open dialog window or Child Window on top of application window using a button click on current application window. In that window I want to show some data in data grid and want to give facility to filter those data in grid to search specific data in grid and then want to give user a functionality where user can select specific row from grid and hit "OK" or "Cancel".

Now my problem is How can I open this window on top of application window? How can I know that which record user have selected? and once user click "OK" or "Cancel" I want that window to go away and do some operation based on selection.

Do you have any sample for this scenario? Or could you give any sample for this scenario?

I can share my solution with you if you want.

Thanks in Advance.

Mitesh.

Reply

Rachel says:

June 15, 2013 at 4:26 pm

Hi Mitesh,

For me it typically depends on what kind of dialog it is.

If it is something like a File/Folder dialog, I usually refer to that from the ViewModel layer.

If it is something like a popup window that contains child content, I usually use a custom Popup Control in my UI layer, and manage properties like IsPopupVisible and PopupContent from the ViewModel layer.

And a 3rd solution is to have a class that is solely responsible for managing dialogs, and use a messaging system to trigger when dialogs appear and what content they should contain.

In your example, I would probably go with the 2nd option of having the ViewModel contain the dialog content (including the SaveCommand and CancelCommand), and have the XAML be written in such a while that it provides a visual representation of your Dialog data when needed.

Reply

AdamAdam says:

April 4, 2013 at 10:51 am

Thank-you for this article, it really helped me! ☺

Reply

Adam says:

April 8, 2013 at 4:14 am

I'm trying to work out why when I switch pages it keeps the previous page running because when I added a MessageBox to the Background worker in my Overview page it kept calling even on other pages and when I clicked Overview again it just added another Overview because I was now getting 2 notices incrementing on clicking Overview button every loop, and so on.

Has anybody experienced this issue?

Reply

Rachel says:

April 8, 2013 at 12:46 pm

Hi Adam,

Make sure you're changing to the existing ViewModel, and not creating a new one each time to switch to the page. And don't forget that changing pages doesn't remove or dispose of the old page at all – it just changes the active page.

If you're having problems with your code, I'd encourage you to post a question on stackoverflow.com and include the relevant parts of your code. You should get an answer fairly quickly, or you can email me the link to your question if you want.

Good luck with it =)

Alex F says:

January 26, 2013 at 1:36 am

Hello,

I'm using your example and wanted to know how I can pass parameter to my user control? My application will have a menu on the left (TreeView most likely) and when user clicks an item in context menu of the tree, I load a User Control on the right side. The user control will need an ID or more parameters to be passed to it so it can populate its View.

I would appreciate it if you could let me know how to implement this.

Thanks.

Alex

Reply

Rachel says:

January 28, 2013 at 5:05 pm

I would have the Model object containing the data for your UserControl in your ViewModel, and simply set the DataContext to be the object in your ViewModel.

Its hard to post a lot of code in these comments, but ideally I'd have my ViewModel contain an ObservableCollection of MyObject classes for the TreeView.ItemsSource to bind to, and a SelectedObject property of type MyObject. Then I would have the UserControl simply bind it's DataContext to SelectedObject.

Reply

Alex F says:

January 29, 2013 at 10:37 pm

Thanks for your reply. I'm still not clear how to do this. Is it possible to point me to an example where TreeView item event will switch between User Controls and able to pass parameter to user control so it can get populated?

Thanks,
Alex

Rachel says:

January 30, 2013 at 1:16 pm

The best example I can find is [this answer of mine on StackOverflow](#). It uses Page objects, however you could do something similar with your objects.

If you have further questions, I'd recommend posting them on that site along with a short code sample demonstrating your problem. They're pretty good at helping you figure out your problem quickly.

Arishtat says:

January 8, 2013 at 1:45 pm

Hmm, if you add a menu and bind the command to a MenuItem in a similar fashion, the MenuItem is disabled even though the CanExecute method always returns true. Can't quite figure out why.

Reply

Rachel says:

January 8, 2013 at 2:52 pm

Hi Arishtat,

Perhaps you could post a question on [StackOverflow](#) with a code sample outlining the problem, and send me a link. Its hard to tell what the problem would be without seeing the code, however a common problem I see with Menus is they're sometimes not part of the same Visual Tree, so the DataContext isn't what you'd expect it to be.

Reply

Amit Sharma says:

December 27, 2012 at 9:56 pm

Nice post Rachel. While using this, I'm facing a problem (or may be it is by design) – each time I switch back to a different viewmodel, a new VIEW instance is created.

Binding on mainview xaml –

When I switch back and forth to let's ActiveViewModel – X, the view associated with X is re-instantiated (I can verify it by seeing view constructor getting hit in debugger).

It leads to subtle bugs as my view has a DataGrid and if user changed some filter, groups – on re-instantiating the view, all these changes are gone.

Note – my viewmodel data is retained each time so I don't have same issue as other folks above.

Reply

Rachel says:

January 2, 2013 at 8:56 pm

Hi Amit,

By default, WPF unloads objects that are not visible, so it sounds like your UserControl is getting unloaded when its hidden, then a new copy of it is loaded when you switch back, which causes all changes made to properties that are not bound to something in the DataContext to be reset.

What I usually end up doing to avoid this behavior is reusing a custom TabControl control which stores the ContentPresenter of each TabItem when you switch tabs, and reloads the same ContentPresenter when you switch back. This preserves the control's state when it's not visible, and reloads it just as it was when you navigated away from it.

An example of the code and how it works can be found in [this StackOverflow answer of mine](#). Perhaps I'll write this solution up as my next blog post, as I think the original site I got the idea and most of the code from no longer exists 😊

Good luck with your project,
Rachel

Reply

Kissack says:

October 10, 2012 at 3:15 pm

Hello Rachel,

I'm a total beginner with WPF and MVVC. Followed your example but having a little problem. Once I leave one view and return to it again the ViewModel's constructors for that view is called and loses its state. Why is that and how can I avoid it ?

Reply

Rachel says:

October 10, 2012 at 7:44 pm

Hi Kissack,

WPF unloads resources which are not in use, which includes non-visible UI objects, so it sounds like your View is initializing a new ViewModel as its DataContext when the View loads, which is usually not something I recommend.

Check your View's constructor, Loaded event, or XAML and see if the DataContext is being set anywhere to a new instance of your ViewModel.

Reply

Kissack says:

October 11, 2012 at 9:54 am

Hi Rachel, the view was one older experimental, where a data context was being set in XAML, so the ViewModel was recreated. When I removed it, the state is maintained. Thank you for your time...

Lora says:

December 11, 2012 at 2:33 pm

Hi, I have the same problem, would you mind to explain it? I don't know how to set the DataContext to not lose the state. Thanks.

Rachel says:

December 12, 2012 at 9:00 pm

Hi Lora,

If your current bound controls appear to be losing their state, then you're probably resetting the DataContext they are bound to somewhere, such as setting the DataContext manually in the code-behind your controls.

Usually you do not want to ever set the DataContext from the code behind your UserControls, as you are supposed to pass the DataContext to your UserControl at the time it gets used, and setting it explicitly in the code-behind will prevent that from happening.

If you're struggling to understand the DataContext, I'd recommend reading a recent blog post I wrote: [What is this "DataContext" you speak of?](#)

Goodluck with it,
Rachel

Jeff says:

July 3, 2012 at 9:10 pm

Thanks Rachel!

Reply

Jeff says:

June 28, 2012 at 6:31 pm

Nice example. Working on a POS design where the initial screen is a keypad login. So, in this example, I'm struggling with how you would start out with a login view. Once the user logs in then navigate to the home page???

Reply

Rachel says:

June 28, 2012 at 8:03 pm

In the past when I've done a login screen, I show and validate the login before the main application even launches in the OnStartup method of app.xaml.cs


```
protected override void OnStartup(StartupEventArgs e)
{
    base.OnStartup(e);

    var login = new LoginDialog();
    var loginVm = new LoginViewModel();

    login.DataContext = loginVm;
    login.ShowDialog();

    if (!login.DialogResult.GetValueOrDefault())
    {
        Environment.Exit(0);
    }

    // Providing we have a successful login, startup application
    var app = new ShellView();
    var context = new ShellViewModel(loginVm.CurrentUser);
    app.DataContext = context;
    app.Show();
}
```

Hope this helps =)

Reply

Jeff says:

June 28, 2012 at 9:15 pm

Ok, makes sense. Question, per your suggestion I added a logindialog window and that appears on startup. However, after logging in ok, the app just closes (ShellView never appears). Anything come to mind?

Rachel says:

June 29, 2012 at 3:58 am

Set the Application.ShutdownMode to something other than the default value of OnLastWindowClose. I usually use OnExplicitShutdown

Hal says:

September 11, 2013 at 8:06 am

Hey Rachel

What should I pass in my login fuction (logindialog window) to close it and then open ShellView, was trying different solutions and nothing happened to open ShellView...

I even added "Application.ShutdownMode: like you wrote before.

Thanks in advance.

Rachel says:

September 12, 2013 at 12:42 pm

Hi Hal, if you're using ShowDialog to show the login window, then the code in your OnStartup should resume once your login dialog returns a DialogResult

Nilow says:

April 24, 2012 at 11:24 am

Nice example,

Tried it in my application, because I didn't want to use Prism for this small app.

The only thing I want to change, is instead of putting all the buttons on the shellview / applicationview, I just want to register the views in the shellview. Then adding buttons on different views, which set the IPageViewModel CurrentPageViewModel on the shellview to the appropriate view.

But my problem is that I cannot reach the List PageViewModels from my viewmodels, and set the CurrentPageViewModel from there.

How can I solve this?

Reply

Rachel says:

April 25, 2012 at 6:44 pm

Hi Nilow,

Usually I use some kind of Event System for communication between ViewModels. Typically I'll have my ShellViewModel listen for something like ChangePageEvents, and any button which should change the page simply broadcasts a ChangePageEvent message, which will get picked up and executed by the ShellViewModel.

I wrote a brief overview of what event systems are and how they work at [the link above](#), although someday I hope to write a more comprehensive guide to how to use one.

Reply

Vidar Lund says:

April 23, 2012 at 3:07 pm

Rachel,

Thanks for your excellent examples. You explain things without assuming that the audience knows a lot. I especially liked your comment that the ViewModels are the application, forget about the Views, they are just a distraction.

Wish that you did this example using MVVM Light but I'll sort it out. Believe that the main difference is that the logic you have put in ApplicationViewModel needs to be in the ViewModelLocator which is loaded upon starting the application in MVVM Light.

Reply

Vidar Lund says:

April 25, 2012 at 8:11 am

I sorted it out. The solution was to make a top level ShellVM that handles the navigation just like your ApplicationVM. Trying to put the navigation logic in the VMLocator was a mistake.

Reply

Rachel says:

April 25, 2012 at 6:36 pm

Glad you got it sorted out. I don't actually use a ViewModelLocator if I can help it, because I feel like it's a bit limiting and I've had some issues with it in the past.

Benjamin Gale says:

March 26, 2012 at 11:06 am

Thanks for the excellent article.

I thought the use of a DataTemplate to discover which view to use for each viewmodel was really clever!

This really helps to keep the View separated from the ViewModel and means that a different Window could use a different View for the same ViewModel without any changes to the ViewModel.

Do you have any ideas about how the lifetime of the each view is managed by WPF? If i had a large number of views to switch between, for example, how could I dispose of the views no longer in use?

Reply

Rachel says:

March 26, 2012 at 3:13 pm

Hi Benjamin,

WPF will automatically unload the View once it's no longer visible. For example, if you have a TabControl with ViewA in one tab, and ViewB in another tab, then switching from the ViewA tab to the ViewB tab will automatically unload ViewA and load ViewB.

Reply

Benjamin Gale says:

March 26, 2012 at 3:48 pm

Hi Rachel, thank you for replying.

Does your comment above apply to all WPF views or is it specific to the TabControl?

E.G. in your code example, does clicking the "Home" button completely dispose of the products user control?

Are you aware of any situations where this automatic behavior could cause unexpected memory leaks?

Rachel says:

March 26, 2012 at 5:51 pm

It should apply to all controls, not just the TabControl. WPF will only load controls that are currently visible on the screen.

Benjamin Gale says:

March 26, 2012 at 6:10 pm

I used WPF Inspector to check this out earlier and it appears the visual tree does change when I switch between the home and the products pages.

Thank you again for your reply it was much appreciated. I look forward to your next blog post.

Ricardo Duran says:

February 2, 2012 at 10:28 pm

Hello Rachel,

Thanks for your example of navigation.

I am new in WPF and MVVM, I am learning.

I followed your example and every thing work fine.

In my mainwindow, I have four buttons to call differents user control, like Home page, products, devices and device list.

I the "Device List" I have a button. When I click the button, a listbox needs to be filled with data.

My problem is: When I click the button, nothing happend, but if I click a button to call for example the Home Page and then click the button to call "Device List" again, I can see now the listbox with all information that I read from tyhe database.

Do you have any idea of this problem?

Thanks in advanced for your help

Ricardo

Reply

Rachel says:

February 3, 2012 at 3:40 am

Hi Ricardo,

It's hard to tell what the issue is without seeing some code, however I am assuming the Button is on the DeviceView, so the click event should be handled in the DeviceViewModel via a RelayCommand.

Perhaps you can post a question on <http://www.stackoverflow.com> outlining your problem, along with some sample code, and someone can help you out.

Goodluck with it,

Rachel

Reply

Mark Wiskochil says:

January 19, 2012 at 8:08 pm

Thanks for an easy to understand but very powerful and useful article. I am using this method on my current project. It works very well. The only change I made was to check if the view model you want to change to is the one you are currently on, I don't change the view model.

Reply

Rachel says:

January 19, 2012 at 9:10 pm

Hi Mark, glad you like the article!

Definitely alter the shell class to suit your needs, and don't stick with what I have here! Its not really meant to be a concrete class anyways – instead its more of an abstract idea which you can alter to fit the navigation style of whatever project you're working on. It's usually different for every project I work on, however all of them navigate with some kind of Application or Shell ViewModel that has a similar concept to this.

Reply

Radu says:

January 3, 2012 at 4:37 pm

Rachel,

I really like a lot your approach, it is definitely a way out to solve the navigation issue I currently have. In one of your posts on stackoverflow related to navigation, you recommended to get rid of defining DataContext in the UserControl:

<http://stackoverflow.com/questions/8677500/wpf-mvvm-load-an-usercontrol-at-runtime>

How can I solve the "Blendability" (sample data visible in Expression Blend) of the user controls in this way? Currently I followed the method described in MVVM Light toolkit. Having a Locator class, by using the Unity IoC it separates design time data from the runtime data so Expression displays sample data in my user controls.

Once I switch to the ContentControl and DataTemplate way, I cannot have design data anymore, as the DataContext which I currently use would break the approach you suggest. Can you please help me with an idea related to this matter? I still want to have my user controls blendable.

Many thanks,

Radu

Reply

Rachel says:

January 3, 2012 at 5:25 pm

Hi Radu,

I don't use Expression Blend a lot, but I believe you can use the d: prefix to specify a design-time DataContext. This will allow your UserControls to have a DataContext when in the Designer, but not hard-code the DataContext into your UserControl

```
<UserControl
    mc:Ignorable="d"
    d:DataContext="{Binding DesignTimeDataContext}"
    ... >
```

I'm not positive, but I believe the design-time markup extensions were added in VS2010, so I think ViewModelLocators were frequently used prior to that to solve the Blendability problem you're describing. Personally I don't like using ViewModelLocators at all (Some of the reasons can be found [here](#))

Good luck with your project,

Rachel

Reply

Bob says:

December 21, 2011 at 9:43 am

Very nicely written article Rachel, this helped me out no end with navigation in MVVM.

The code displayed above for the ApplicationViewModel has a list to hold the view models, I could be wrong but neither of the declarations have the generic type stated, should it not be something like:

```
private List _pageViewModels;
```

If so I think the property needs to be changed to follow suit.

Thanks :)!

Reply

Rachel says:

December 21, 2011 at 12:34 pm

Thanks Bob, you're right. When I copy/pasted the code it got rid of the > and < characters, and everything between them. I'll fix that.

Reply

Prasanth Guruprasad says:

December 20, 2011 at 11:34 am

Goodness, you are still using Windows XP? When are you planning to move to Win 7 (and Win 8)?

Reply

Rachel says:

December 20, 2011 at 12:37 pm

lol I have Windows 7 at work, but both my home computers are XP. They're built for gaming so slightly more expensive and I haven't really had a need to replace them yet.

Reply

[Blog at WordPress.com.](#)