# Setting up a Multi-Device Windows Solution

## Introduction

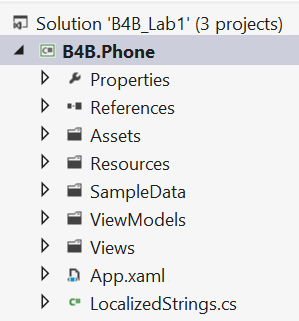
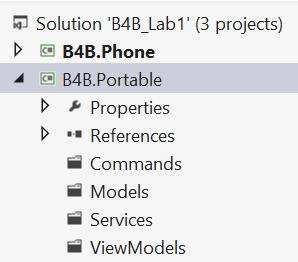
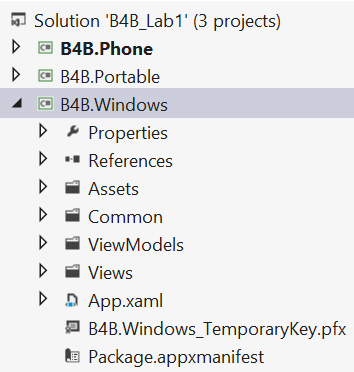
Visual Studio supports a great structure for building apps concurrently for Windows and Windows Phone. In this lab, we will start with an application shell and build out the structure of a properly architected Windows 8 / Windows Phone 8 app.

## Pre Requisite

* [Windows 8](http://technet.microsoft.com/en-us/evalcenter/hh699156.aspx) or [Windows 8.1](http://technet.microsoft.com/en-US/evalcenter/hh699156.aspx).
* Visual Studio 2013 or 2012 Professional, Premium

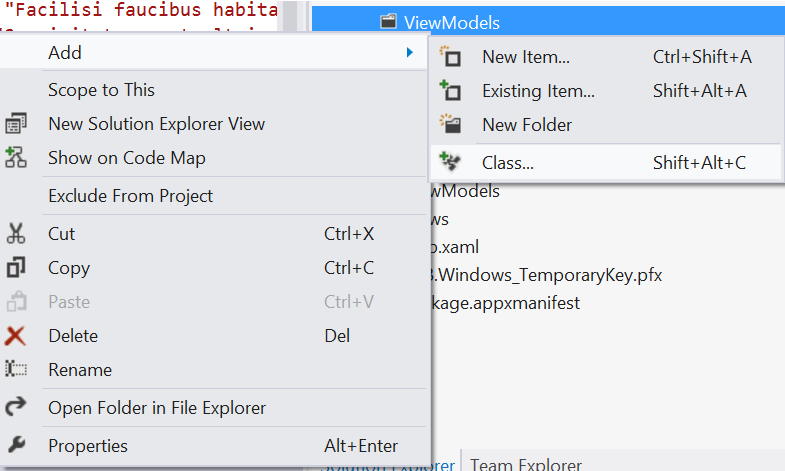
## Step by Step

1. Open up the solution B4B\_Lab1. What you see here should look familiar from the demo we’ve just seen. This is a slightly modified version of
   1. A new PCL (Portable Class Library
   2. A Windows Phone 8 “Databound App” template project
   3. A blank Windows 8 Store app

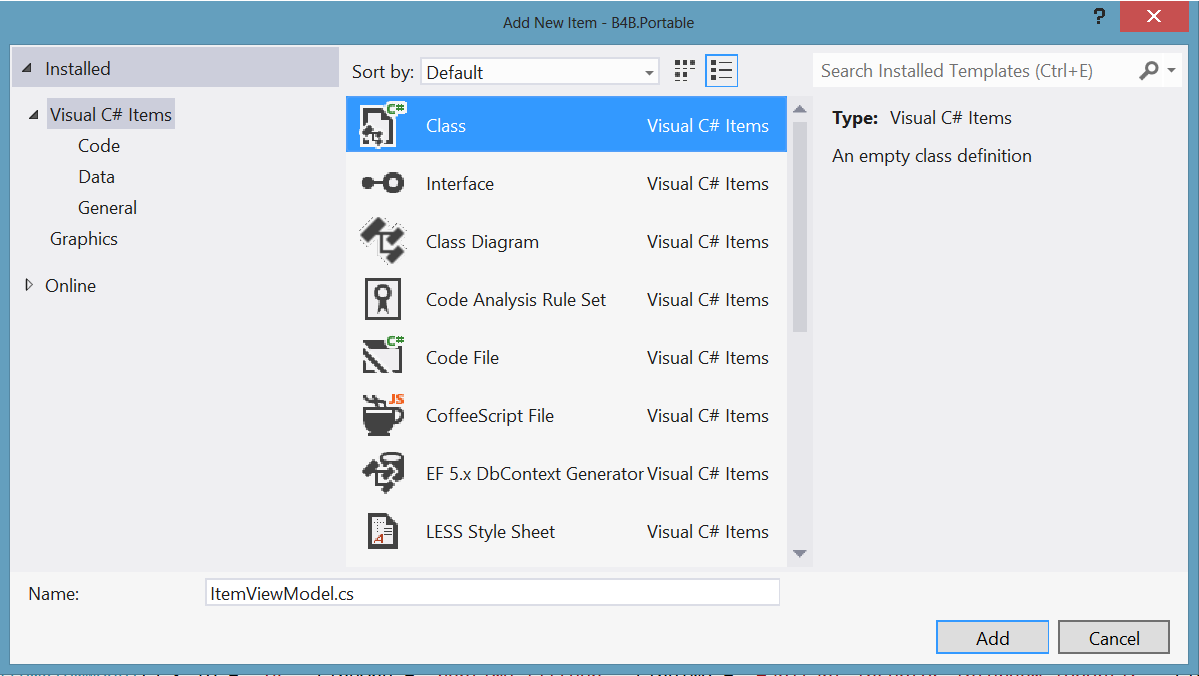
  

Figure

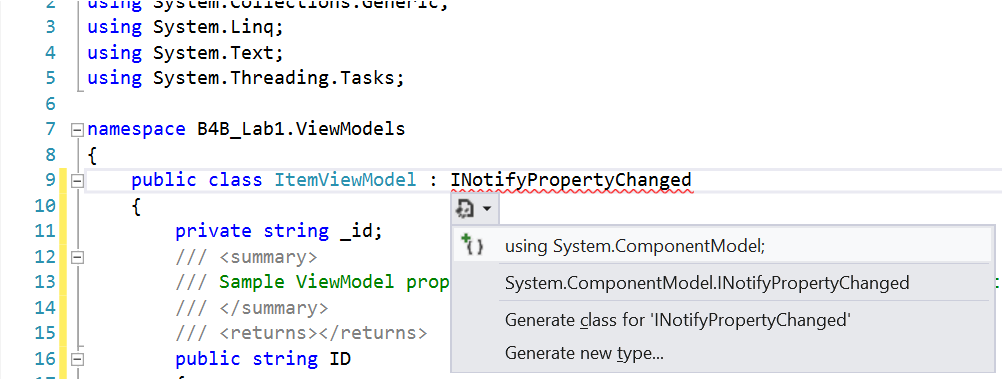
1. The Windows Phone app is already set up with some sample data that we can use to see how a databound application might work. Open up **B4B.Phone** and open the **MainViewModel.cs**. Scroll down to line 64 and see the **LoadData** method. This populates the viewModel with data that will display at runtime.
2. Right now this code is duplicated in both projects. We will move it to a portable class library, where we can call this data from either application.
3. The first step is going to be to add the appropriate data objects into the PCL. Open the **B4B.Portable** project and right-click on the **ViewModels** folder and navigate to **Add -> Class…**



1. Name it **ItemViewModel.cs**

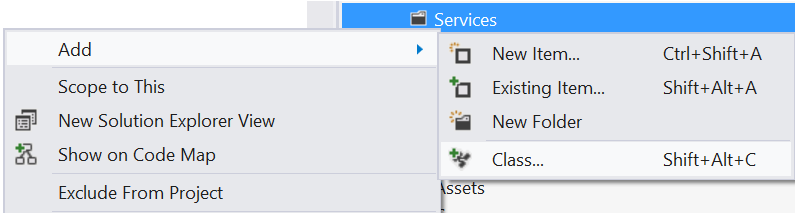


1. Copy the class code (from **line 13** to **line 108**) from **ItemViewModel.cs** in the Windows Phone project, located at **B4B.Phone/ViewModels/ItemViewModel.cs**. Paste it into the **ItemViewModel.cs** in your PCL project, replacing **lines 9-11**. Place the cursor on the **INotifyPropertyChanged** error and press **Alt-Shift-F10** to resolve the namespace. Select **using System.ComponentModel**

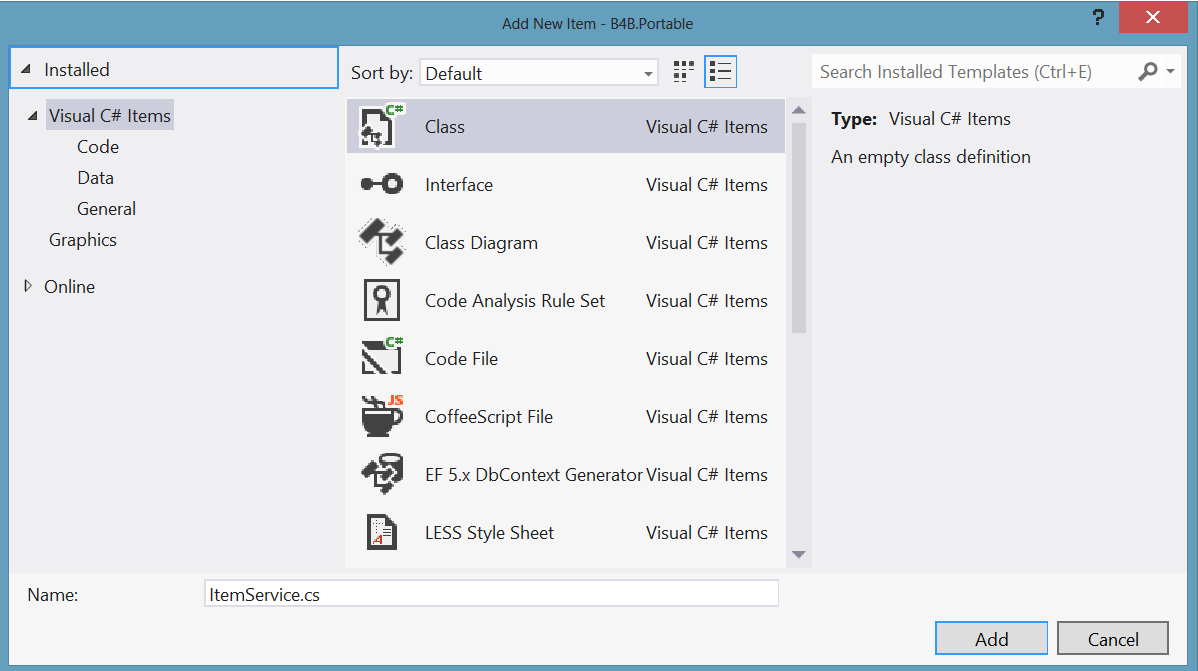


1. Now that our data object is in place, we need to add a data service to serve the data to our two applications. In future labs, we will build a service that connects to an internet service to deliver this data, but for now, we will just serve up the sample data created by out template.

Right-click on the **Services** folder in the **B4B.Portable** project and navigate to **Add -> Class…**



1. Name the new file **ItemService.cs**



1. Add the following code.

public static class ItemService

{

public static ObservableCollection<ItemViewModel> GetItems()

{

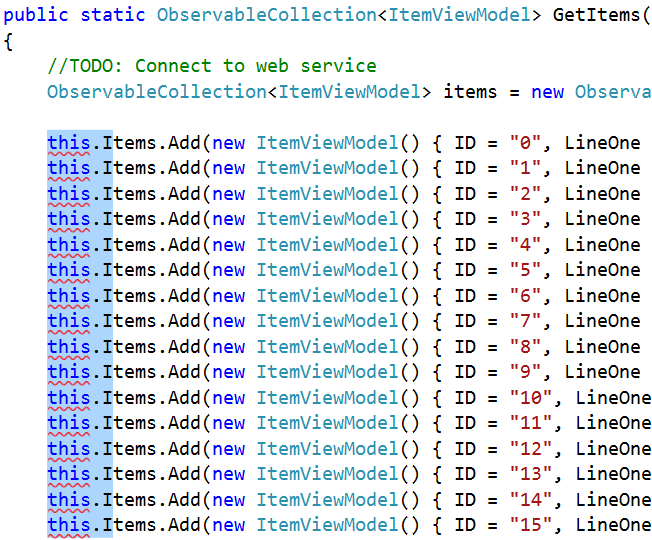
//TODO: Connect to web service

ObservableCollection<ItemViewModel> items = new ObservableCollection<ItemViewModel>();

}

}

1. When **ObservableCollection** and **ItemViewModel** don’t resolve, use the **Alt-Shift-F10** command to add the appropriate namespaces.
2. Go back to the **MainViewModel.cs** in the **B4B.Phone** project and find the **LoadData** method. Copy the code inside this from **line 64 to line 79** and paste it just below the items declaration inside the **GetItems** method in your **ItemService** file.
3. Hold **Alt** and select the following section of the code.



1. Type “i” to change all the

this.Items.Add

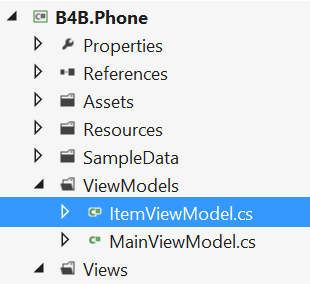
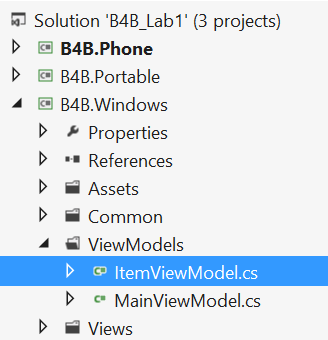
references into

items.Add

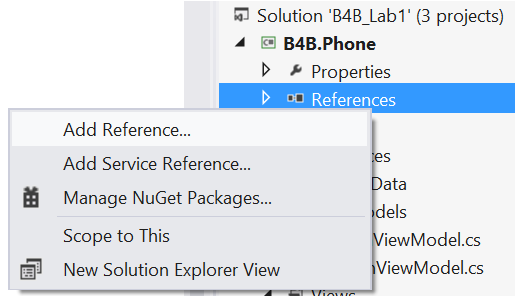
1. At the bottom of the method, add

return items;

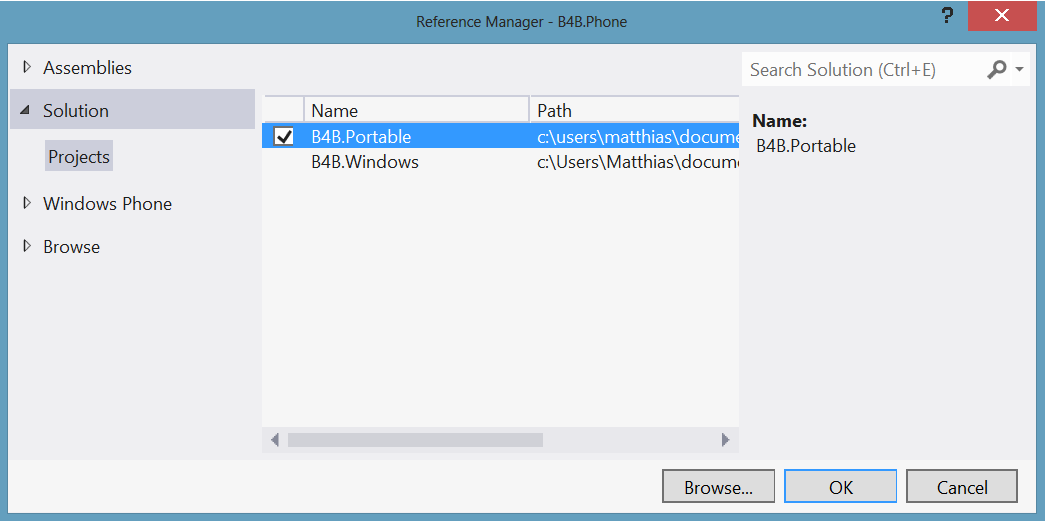
1. We now have a PCL service that can be called from both our projects. In both projects, delete the **ItemViewModel.cs** file. We don’t need it there now that we’ve migrated it to our PCL.

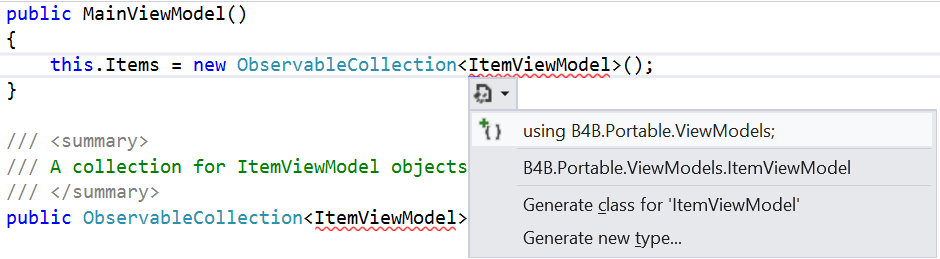
1. For both projects, right-click on the “References” folder and select **Add References…**.



1. Select **Solution -> Projects** and then select the **B4B.Portable** library. Press “OK”.



1. In the **MainViewModel.cs** (in both projects) and MainPage.xaml.cs (in B4B.Phone) your **ItemViewModel** references now need a reference. Use the **Alt-Shift-F10** command to add the **B4B.Portable.ViewModels** reference to the PCL.



1. Inside the **LoadData** method in each project, replace the sample data creation code with

public void LoadData()

{

// Sample data; replace with real data

Items = ItemService.GetItems();

NotifyPropertyChanged("Items");

this.IsDataLoaded = true;

}

1. Now both our apps are running off the same data and the same objects contained within our PCL. Run the application to verify that it is running. Change the data in the PCL to see the updates reflecting in both projects.
2. Congratulations! You’ve completed the first lab to write applications that span the range of Windows devices.

## Conclusion