651.288.7000 info@intertech.com

Building a Universal Windows Platform (UWP) Application (Part 1) – Using Template10

by **admin** | Sep 29, 2016

I just rolled off a client recently that needed to build a Universal Windows Platform (UWP) application as part of their hardware and software solution. For those who aren't yet familiar with UWP, you can check out this article by Tyler Whitney.

As many of you developers out there are aware, sometimes you have to build or bring with you a number of application infrastructure items before you can even get started with the core application logic. For example, you might need some helpers, services and base classes that make your job easier or allow you to start with your base patterns, such as, MVC, MVVM, etc.

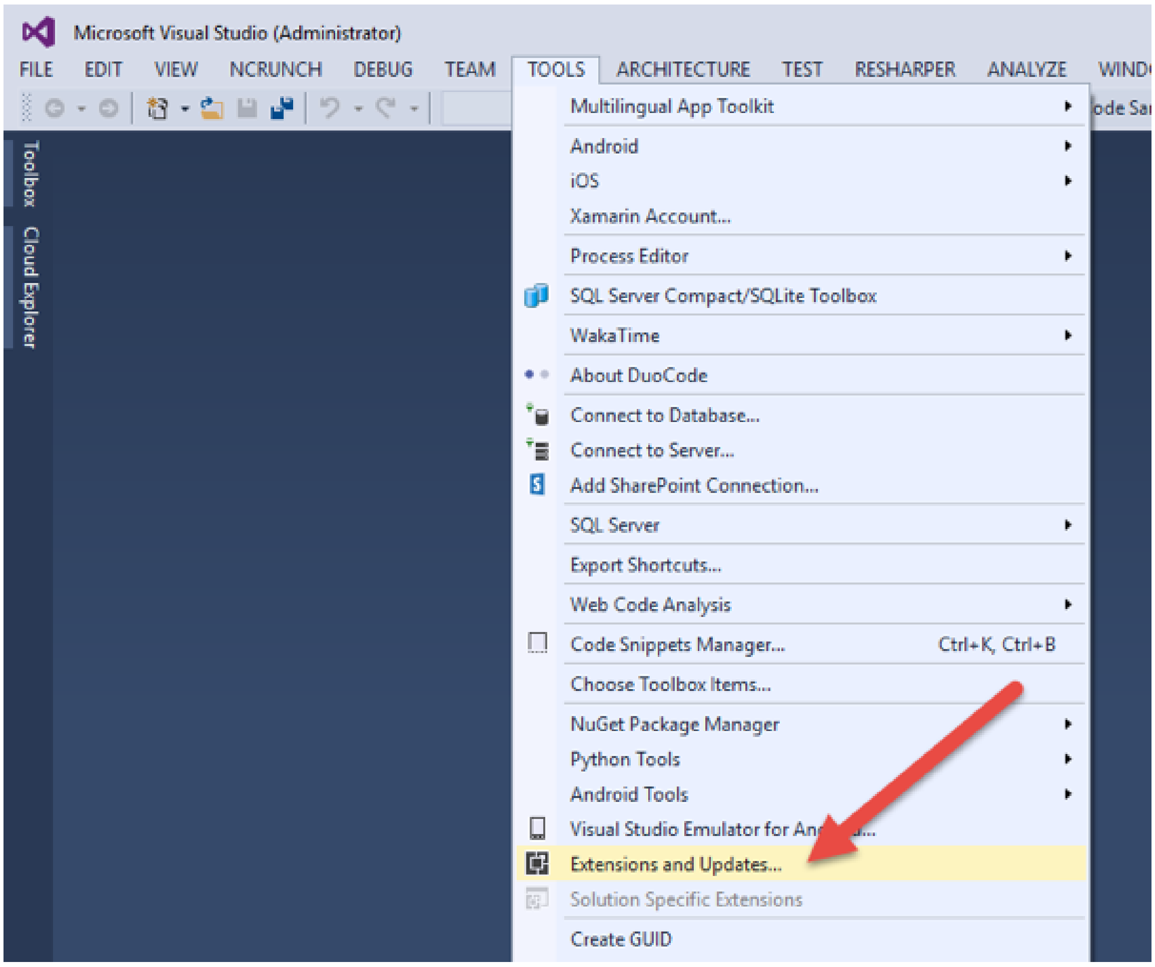
Well, in order to get a jump-start in this direction, a group of Microsoft Evangelists, who have gone through the same troubles, decided to create an open source library of very useful items for UWP development called Template 10. In this first part of this blog series, I will show you how to building a Universal Windows Platform application using this library can be done effectively.

Learn more about Template 10 here.

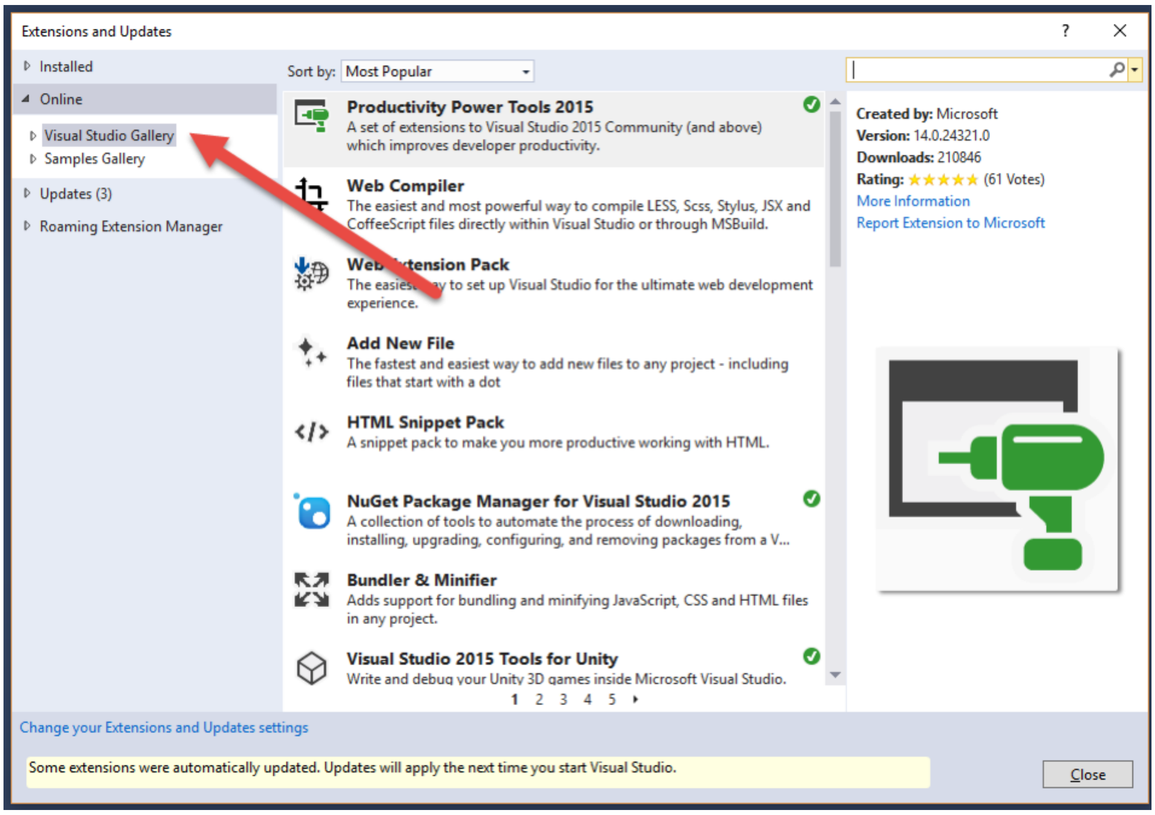
Install Templates

So, to get this started, lets first install templates.

The first thing you will want to do is start Visual Studio 15. From the **main menu**, you can **select TOOLS->Extensions and Updates…**


Microsoft Visual Studio (Administrator)
FILE EDIT VIEW NCRUNCH DEBUG TEAM TOOLS ARCHITECTURE TEST RESHARPER ANALYZE
Toolbox
Cloud Explorer
Multilingual App Toolkit
Android
iOS
Xamarin Account...
Process Editor
SQL Server Compact/SQLite Toolbox
WakaTime
About DuoCode
Connect to Database...
Connect to Server...
Add SharePoint Connection...
SQL Server
Export Shortcuts...
Web Code Analysis
Code Snippets Manager... Ctrl+K, Ctrl+B
Choose Toolbox Items...
NuGet Package Manager
Python Tools
Android Tools
Visual Studio Emulator for Android...
Extensions and Updates...
Solution Specific Extensions
Create GUID

The Extensions and Updates dialog will be displayed. From here, **select Online->Visual Studio Gallery**.


Extensions and Updates

Installed
Online
Visual Studio Gallery
Samples Gallery
Updates (3)
Roaming Extension Manager

Sort by: Most Popular

Productivity Power Tools 2015
A set of extensions to Visual Studio 2015 Community (and above) which improves developer productivity.

Web Compiler
The easiest and most powerful way to compile LESS, Scss, Stylus, JSX and CoffeeScript files directly within Visual Studio or through MSBuild.

Web Extension Pack
The easiest way to set up Visual Studio for the ultimate web development experience.

Add New File
The fastest and easiest way to add new files to any project - including files that start with a dot

HTML Snippet Pack
A snippet pack to make you more productive working with HTML.

NuGet Package Manager for Visual Studio 2015
A collection of tools to automate the process of downloading, installing, upgrading, configuring, and removing packages from a V...

Bundler & Minifier
Adds support for bundling and minifying JavaScript, CSS and HTML files in any project.

Visual Studio 2015 Tools for Unity
Write and debug your Unity 3D games inside Microsoft Visual Studio.

1 2 3 4 5

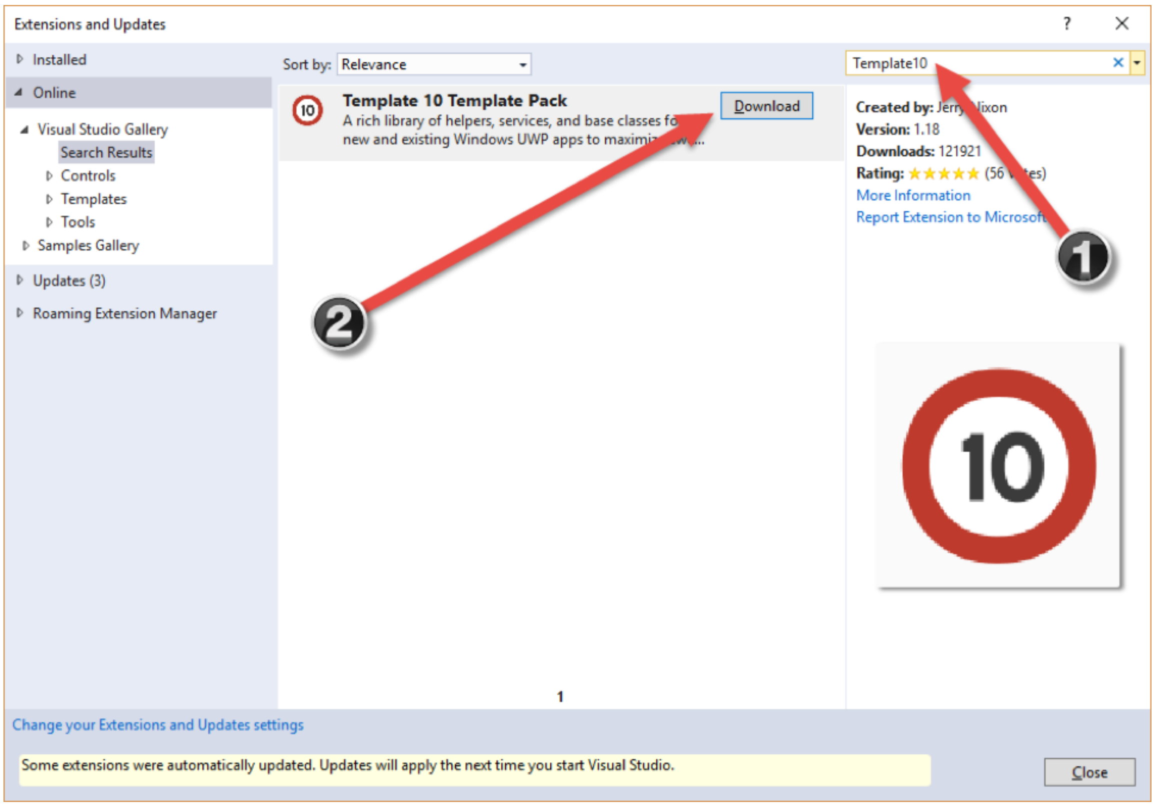
Created by: Microsoft
Version: 14.0.24321.0
Downloads: 210846
Rating: ★★★★★ (61 Votes)
More Information
Report Extension to Microsoft

Change your Extensions and Updates settings

Some extensions were automatically updated. Updates will apply the next time you start Visual Studio.

Close

Next, **type Template10** into the **search box**. You should see a *Template 10 Template Pack* search result item in the list. **Select** the **item** and **click Download**.



Extensions and Updates

Installed
Online
Visual Studio Gallery
Search Results
Controls
Templates
Tools
Samples Gallery
Updates (3)
Roaming Extension Manager

Sort by: Relevance

Template10

Template 10 Template Pack
A rich library of helpers, services, and base classes fo... new and existing Windows UWP apps to maximi...

Download

Created by: Jerry Nixon
Version: 1.18
Downloads: 121921
Rating: ★★★★★ (56 Votes)
More Information
Report Extension to Microsoft

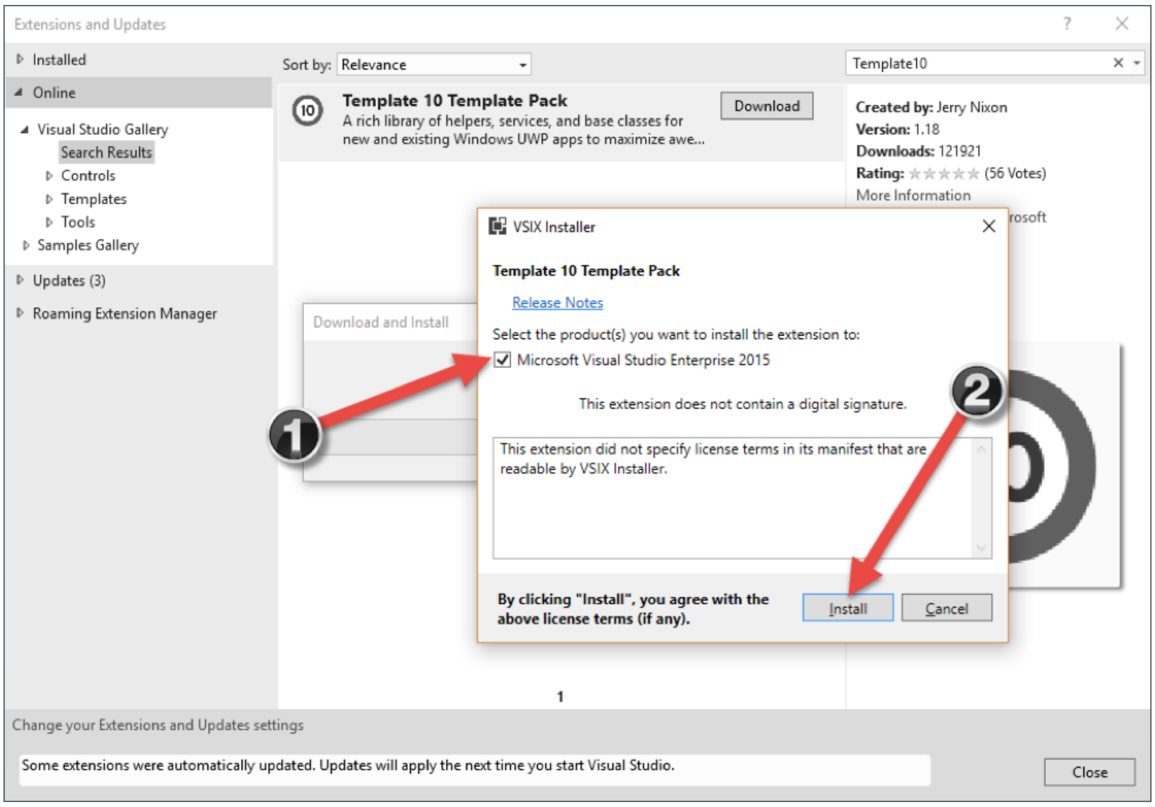
1

Change your Extensions and Updates settings

Some extensions were automatically updated. Updates will apply the next time you start Visual Studio.

Close

A VSIX Installer dialog will be displayed to confirm that you would like to install the Template 10 Template Pack. Make sure that the **checkbox** for the supported **version** of Visual Studio is **selected**. Once you confirmed the version, you should be able to **select Install**.



Extensions and Updates

Sort by: Relevance

Template10

Template 10 Template Pack
A rich library of helpers, services, and base classes for new and existing Windows UWP apps to maximize awe...

Created by: Jerry Nixon
Version: 1.18
Downloads: 121921
Rating: ★★★★★ (56 Votes)
More Information

VSIX Installer

Template 10 Template Pack

Release Notes

Select the product(s) you want to install the extension to:
☑ Microsoft Visual Studio Enterprise 2015

This extension does not contain a digital signature.

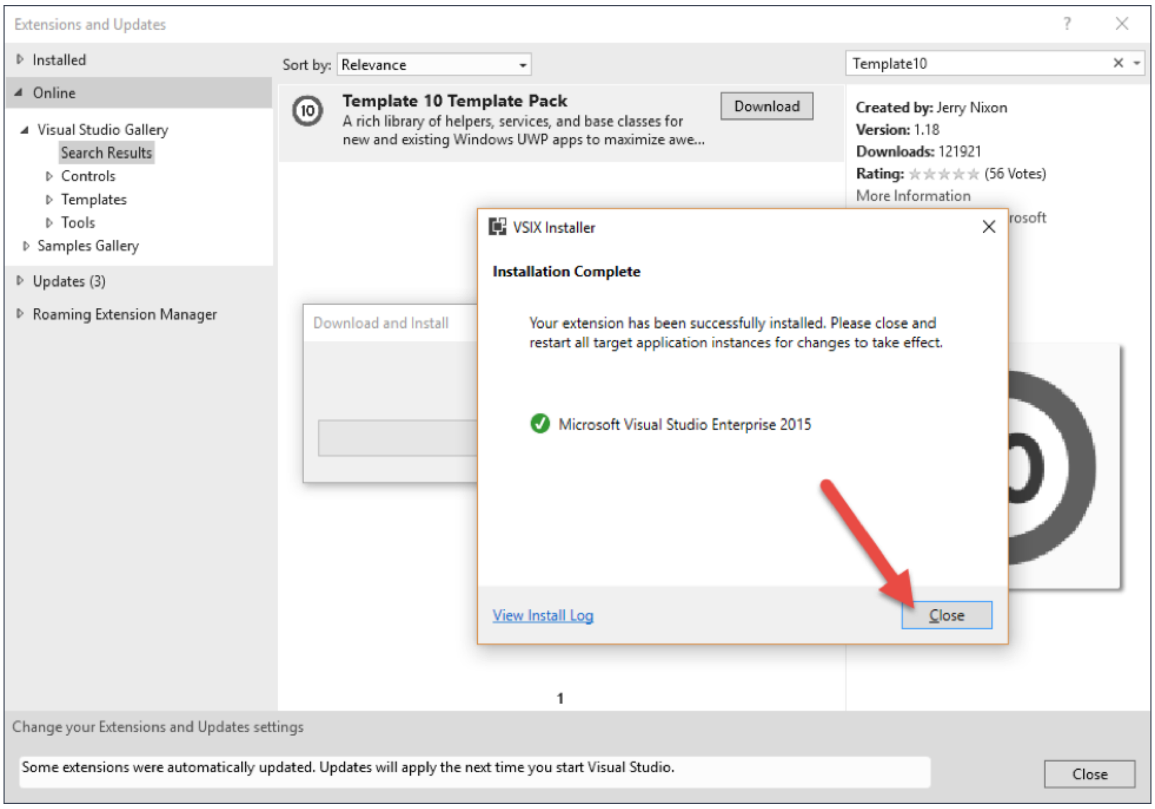
This extension did not specify license terms in its manifest that are readable by VSIX Installer.

By clicking "Install", you agree with the above license terms (if any).

Change your Extensions and Updates settings

Some extensions were automatically updated. Updates will apply the next time you start Visual Studio.

Once the installation completes, you will see the following dialog. **Select Close**.



Extensions and Updates

Sort by: Relevance

Template10

Template 10 Template Pack
A rich library of helpers, services, and base classes for new and existing Windows UWP apps to maximize awe...

Created by: Jerry Nixon
Version: 1.18
Downloads: 121921
Rating: ★★★★★ (56 Votes)
More Information

VSIX Installer

Installation Complete

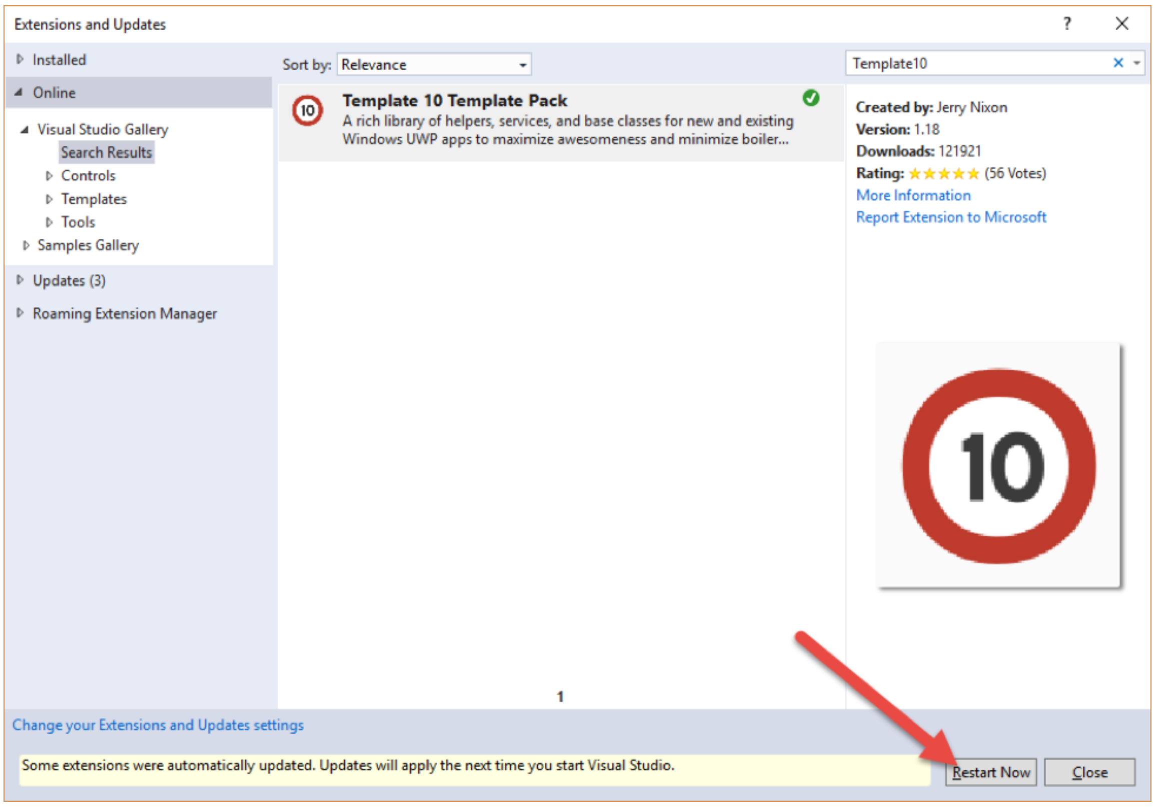
Your extension has been successfully installed. Please close and restart all target application instances for changes to take effect.

✓ Microsoft Visual Studio Enterprise 2015

Change your Extensions and Updates settings

Some extensions were automatically updated. Updates will apply the next time you start Visual Studio.

Visual Studio will now need to be restarted. **Choose Restart Now**.



Template 10 Template Pack
A rich library of helpers, services, and base classes for new and existing Windows UWP apps to maximize awesomeness and minimize boiler...

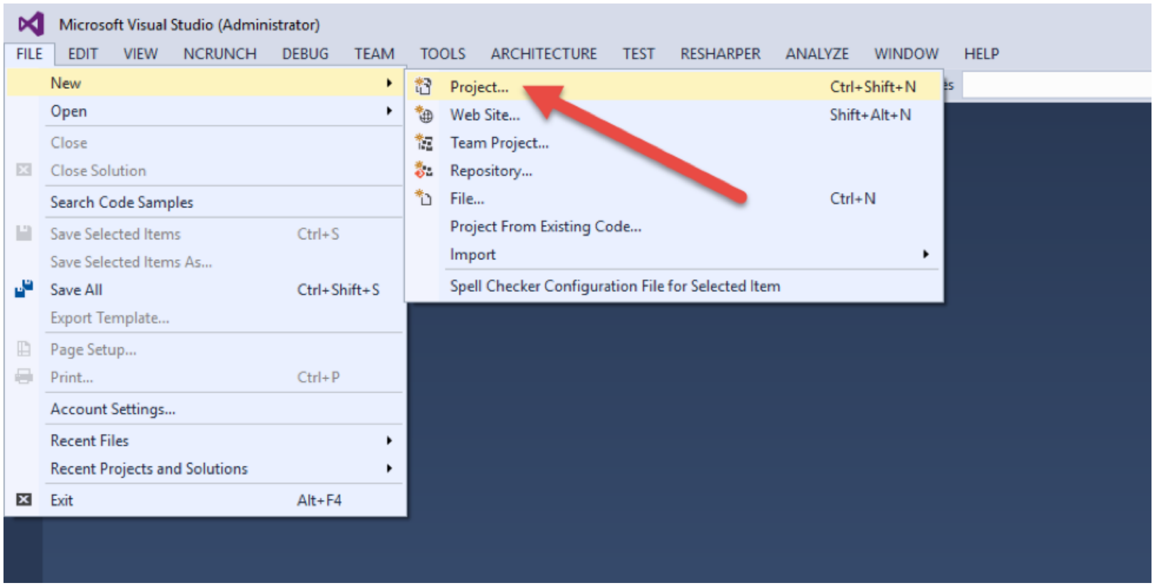
Created by: Jerry Nixon
Version: 1.18
Downloads: 121921
Rating: ★★★★★ (56 Votes)
More Information
Report Extension to Microsoft

Change your Extensions and Updates settings

Some extensions were automatically updated. Updates will apply the next time you start Visual Studio.

Create New Project

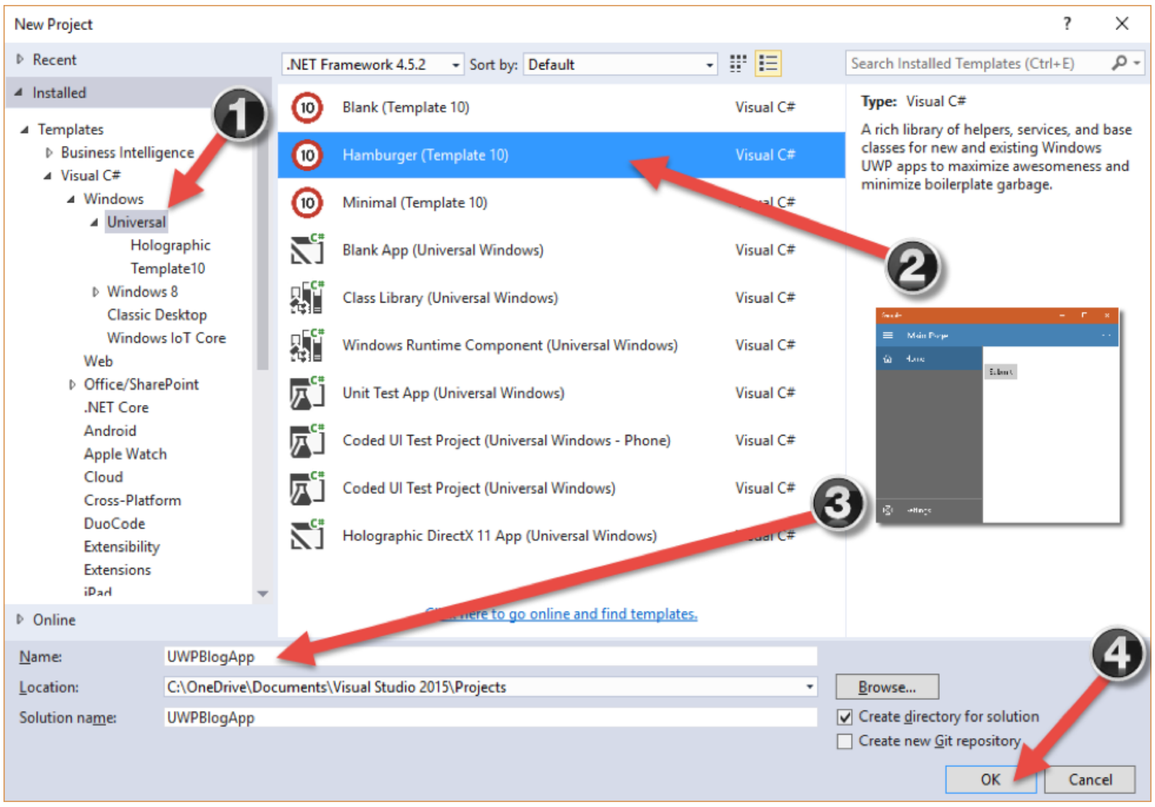
Once Visual Studio has restarted, you will want to create a project with the template. So, **Select FILE->New->Project…**


Microsoft Visual Studio (Administrator)
FILE EDIT VIEW NCRUNCH DEBUG TEAM TOOLS ARCHITECTURE TEST RESHARPER ANALYZE WINDOW HELP
New
Open
Close
Close Solution
Search Code Samples
Save Selected Items Ctrl+S
Save Selected Items As...
Save All Ctrl+Shift+S
Export Template...
Page Setup...
Print... Ctrl+P
Account Settings...
Recent Files
Recent Projects and Solutions
Exit Alt+F4
Project... Ctrl+Shift+N
Web Site... Shift+Alt+N
Team Project...
Repository...
File... Ctrl+N
Project From Existing Code...
Import
Spell Checker Configuration File for Selected Item

The New Project dialog will appear. From here, you can **select Installed/Templates/Visual C#/Windows/Universal**. Once you have Universal selected, you will see the full list of templates for Universal application development. Since I want to include a menu that will take me to various screens in my application, I will **choose** the **Hamburger (Template 10)** template.

NOTE: You could have selected Installed/Templates/Visual C#/Windows/Universal/Template10 and only viewed the Template 10 templates.

Now you can give your application a **name** and **select OK**.


New Project

Installed > Templates > Visual C# > Windows > Universal

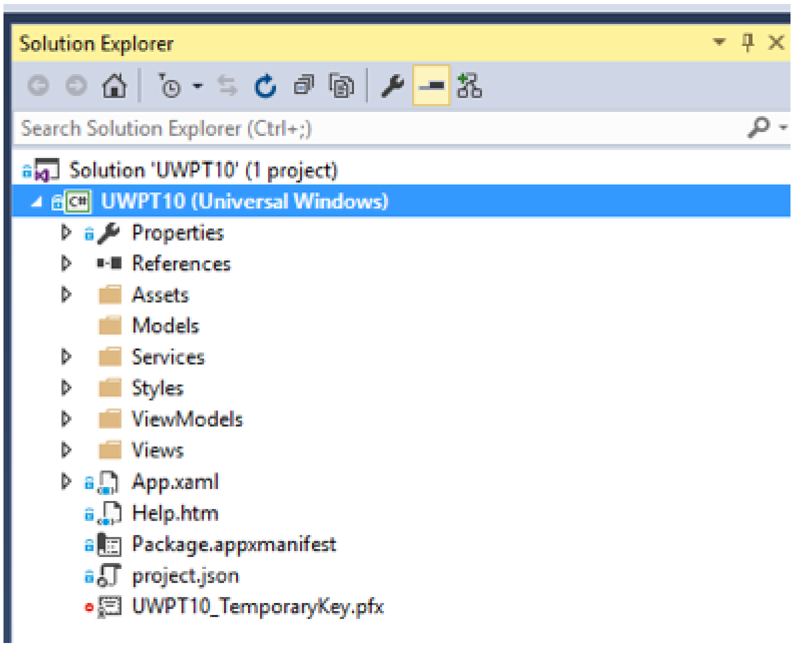
Blank (Template 10)	Visual C#
Hamburger (Template 10)	Visual C#
Minimal (Template 10)	Visual C#
Blank App (Universal Windows)	Visual C#
Class Library (Universal Windows)	Visual C#
Windows Runtime Component (Universal Windows)	Visual C#
Unit Test App (Universal Windows)	Visual C#
Coded UI Test Project (Universal Windows - Phone)	Visual C#
Coded UI Test Project (Universal Windows)	Visual C#
Holographic DirectX 11 App (Universal Windows)	Visual C#

Type: Visual C#

A rich library of helpers, services, and base classes for new and existing Windows UWP apps to maximize awesomeness and minimize boilerplate garbage.

Name: UWPBlogApp
Location: C:\OneDrive\Documents\Visual Studio 2015\Projects
Solution name: UWPBlogApp
☑ Create directory for solution
☐ Create new Git repository

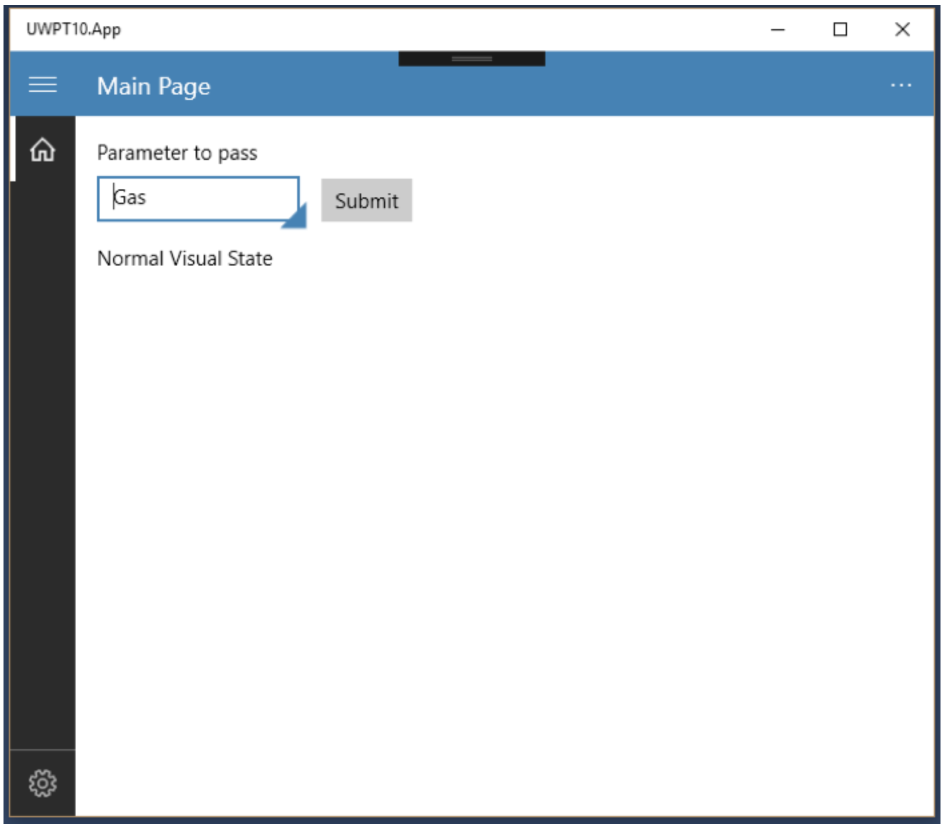
Visual Studio will work its magic and create a new solution and a Universal Windows Platform application based on the Template10 template.


Solution Explorer

Search Solution Explorer (Ctrl+;)

```
Solution 'UWPT10' (1 project)
  UWPT10 (Universal Windows)
    Properties
    References
    Assets
    Models
    Services
    Styles
    ViewModels
    Views
    App.xaml
    Help.htm
    Package.appxmanifest
    project.json
    UWPT10_TemporaryKey.pfx
```

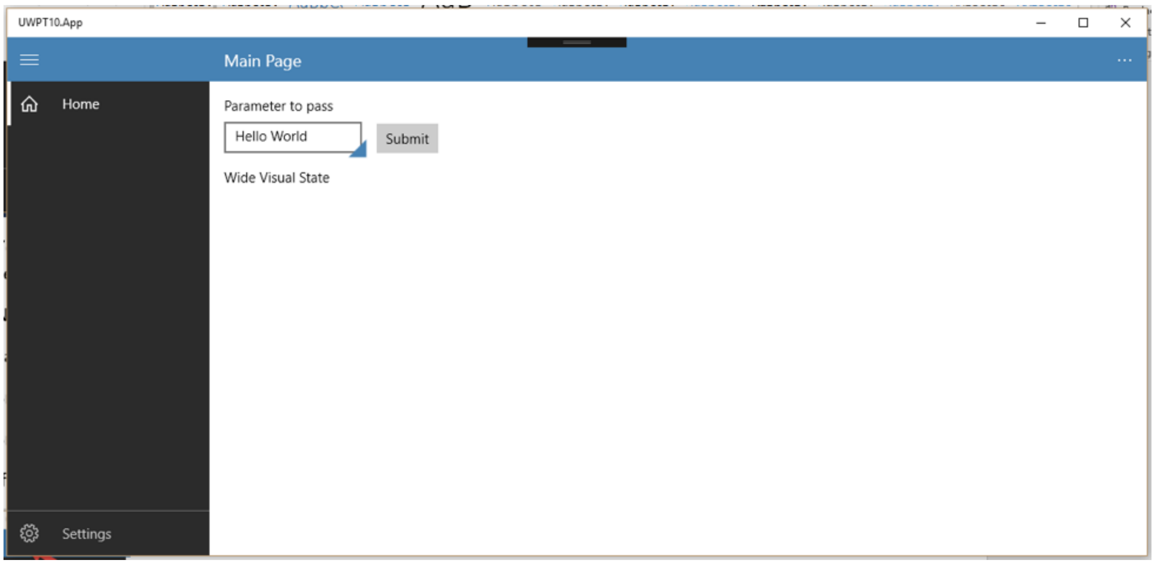
Go ahead and **press F5** and see the new application in action.



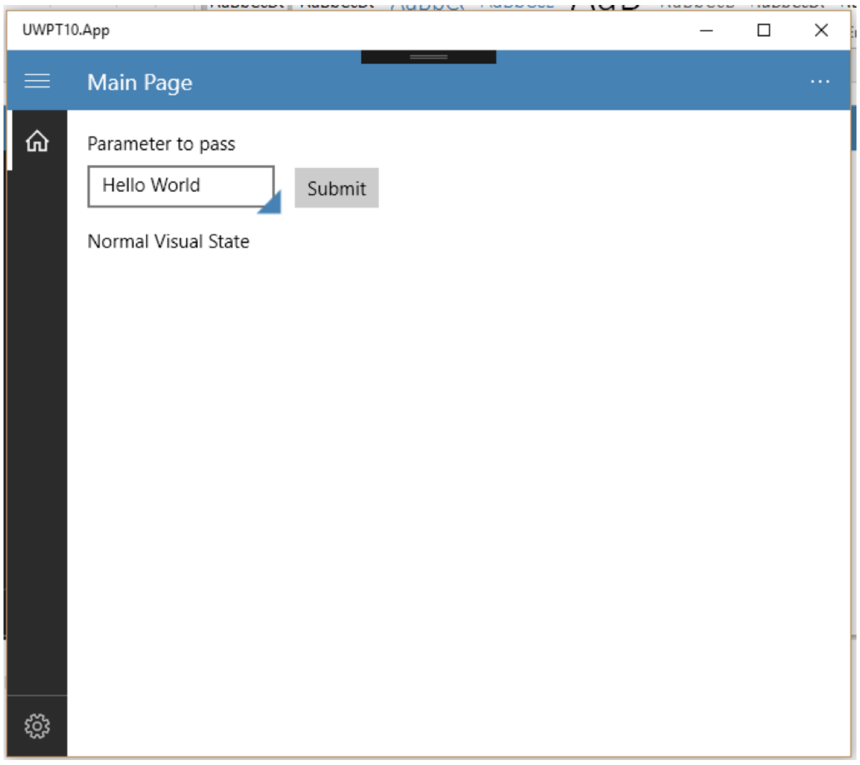
You will notice that it has a main menu which include a hamburger menu item, a home item and a settings item.

NOTE: Depending on the size of windows for the application, the menu will adjust:

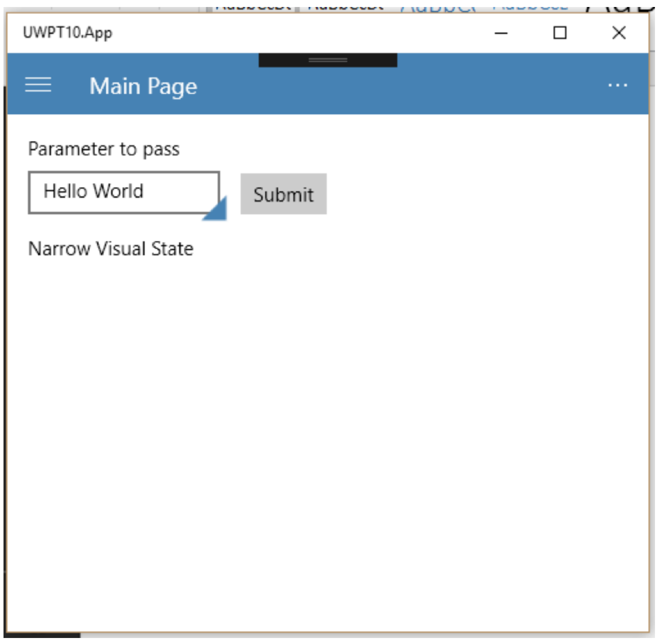
Large Screen



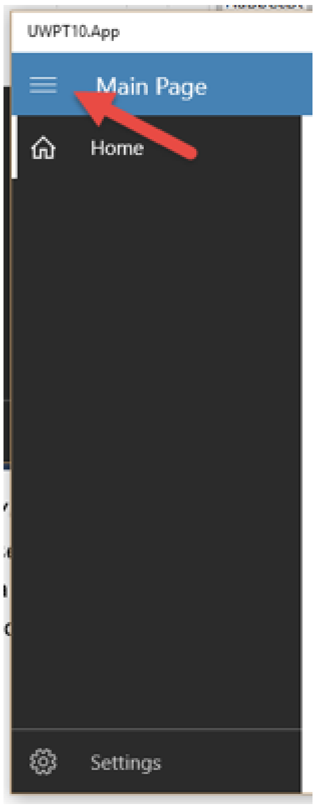
Smaller Screen



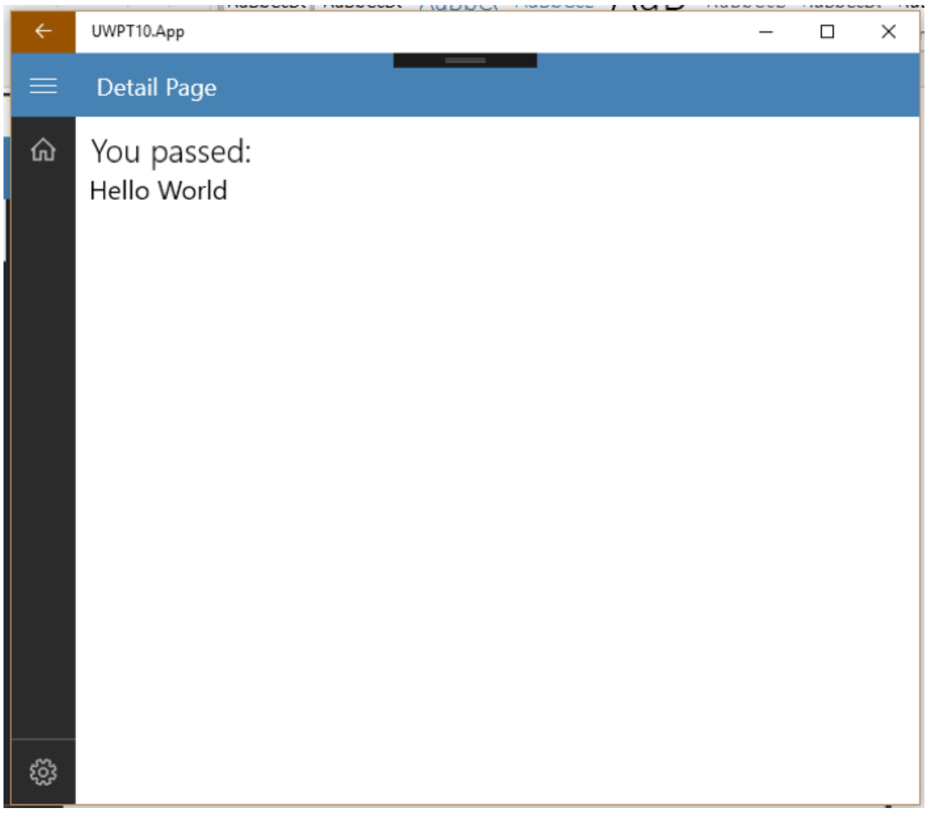
Smallest Screen



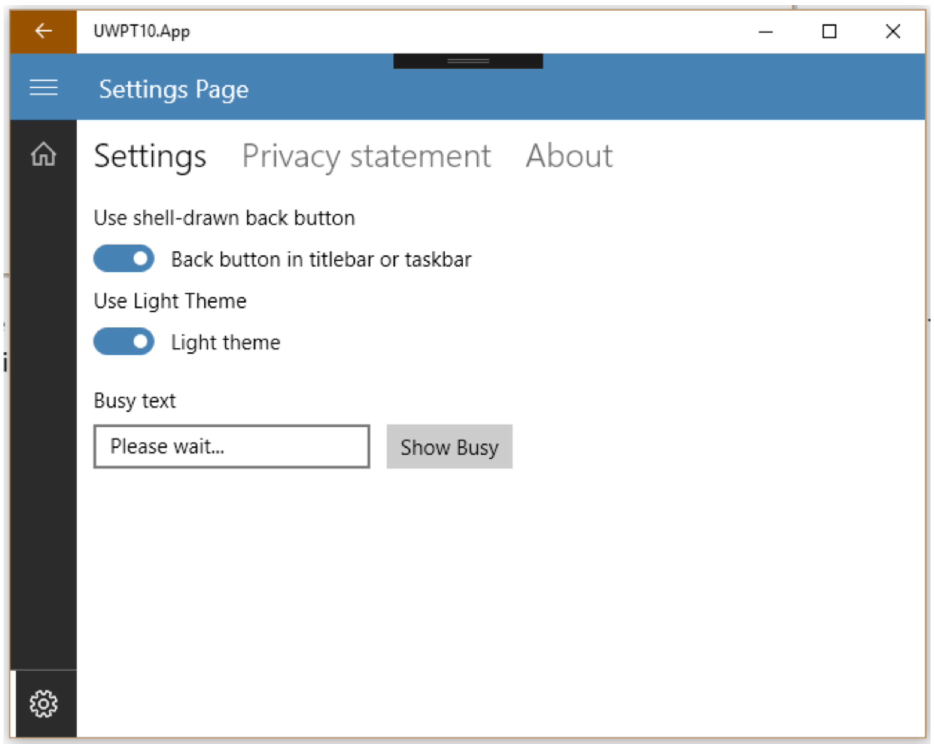
If you select the hamburger menu item, the menu will expand or collapse.



The main page has a string parameter example to demonstrate sending information to another screen. For example, if you were to **type** in "**Hello World**" and **select Submit**, you would see the following:



If you were to select the settings menu item, it would take you to the settings page with a couple of pre-built in options and information for you to explore.


UWPT10.App

Settings Page

Settings Privacy statement About

Use shell-drawn back button

Back button in titlebar or taskbar

Use Light Theme

Light theme

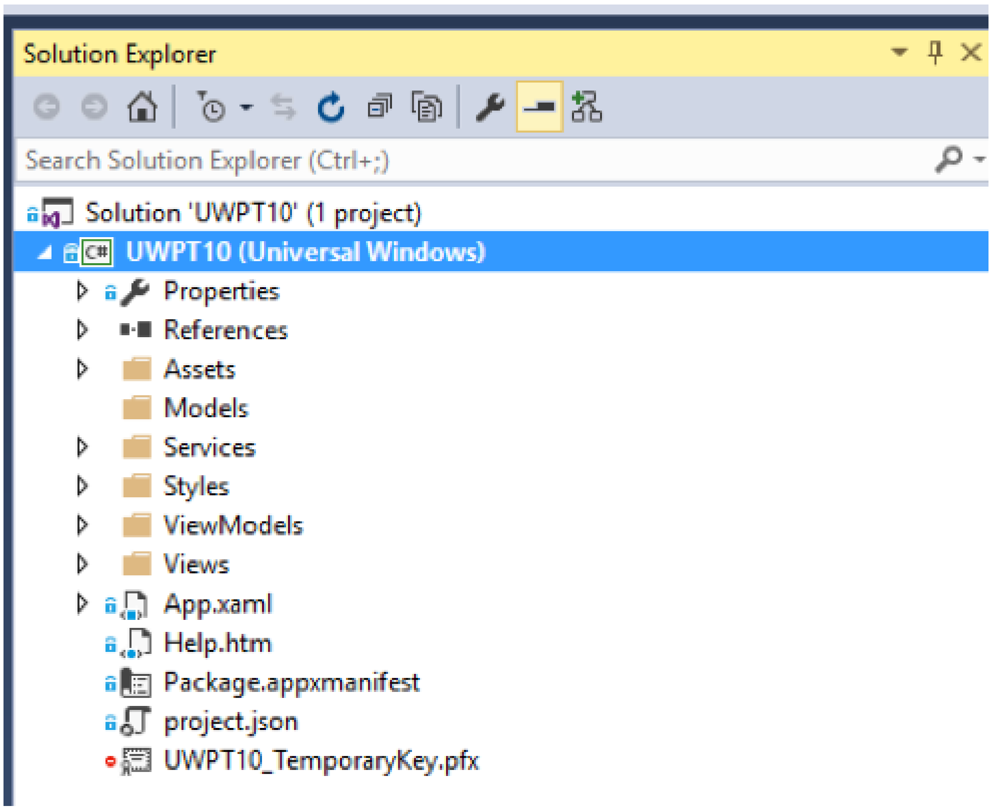
Busy text

Please wait...

Show Busy

Application Structure

Let's take a look at the UWP application structure. Stop the application and take a look at how it is structured.

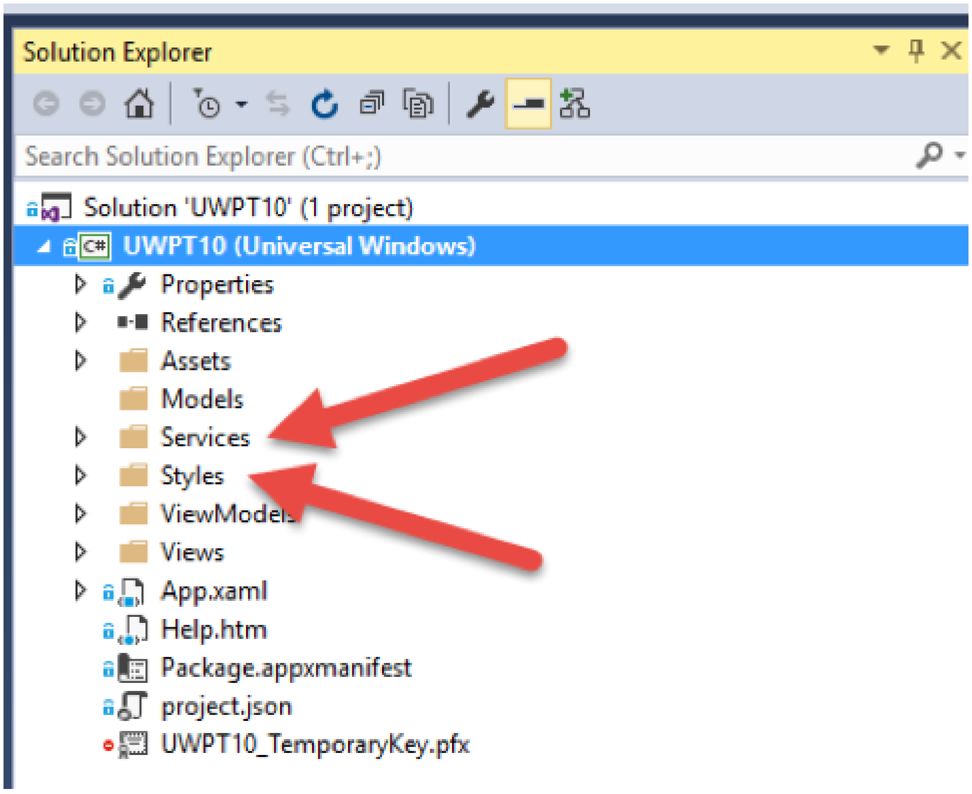

Solution Explorer

Search Solution Explorer (Ctrl+;)

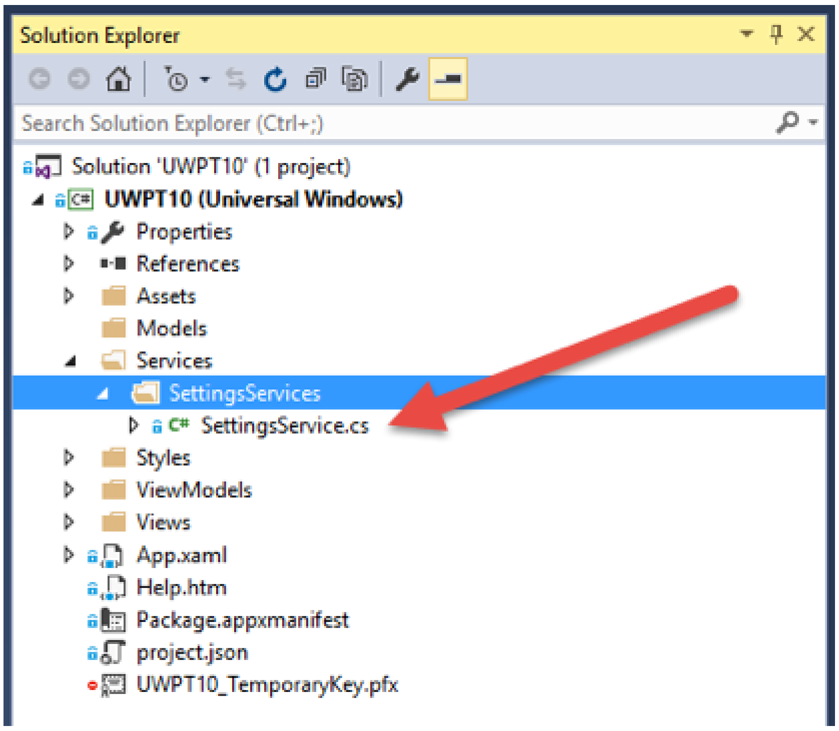
```
Solution 'UWPT10' (1 project)
  UWPT10 (Universal Windows)
    Properties
    References
    Assets
    Models
    Services
    Styles
    ViewModels
    Views
    App.xaml
    Help.htm
    Package.appxmanifest
    project.json
    UWPT10_TemporaryKey.pfx
```

Services and Styles

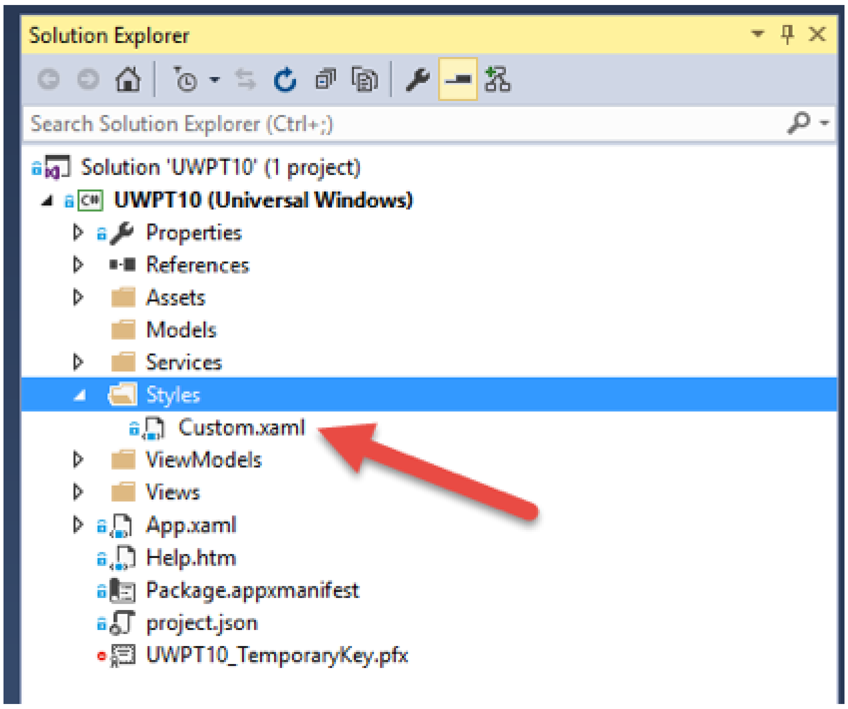
The Template10 template also added a Services and Styles directory for us.



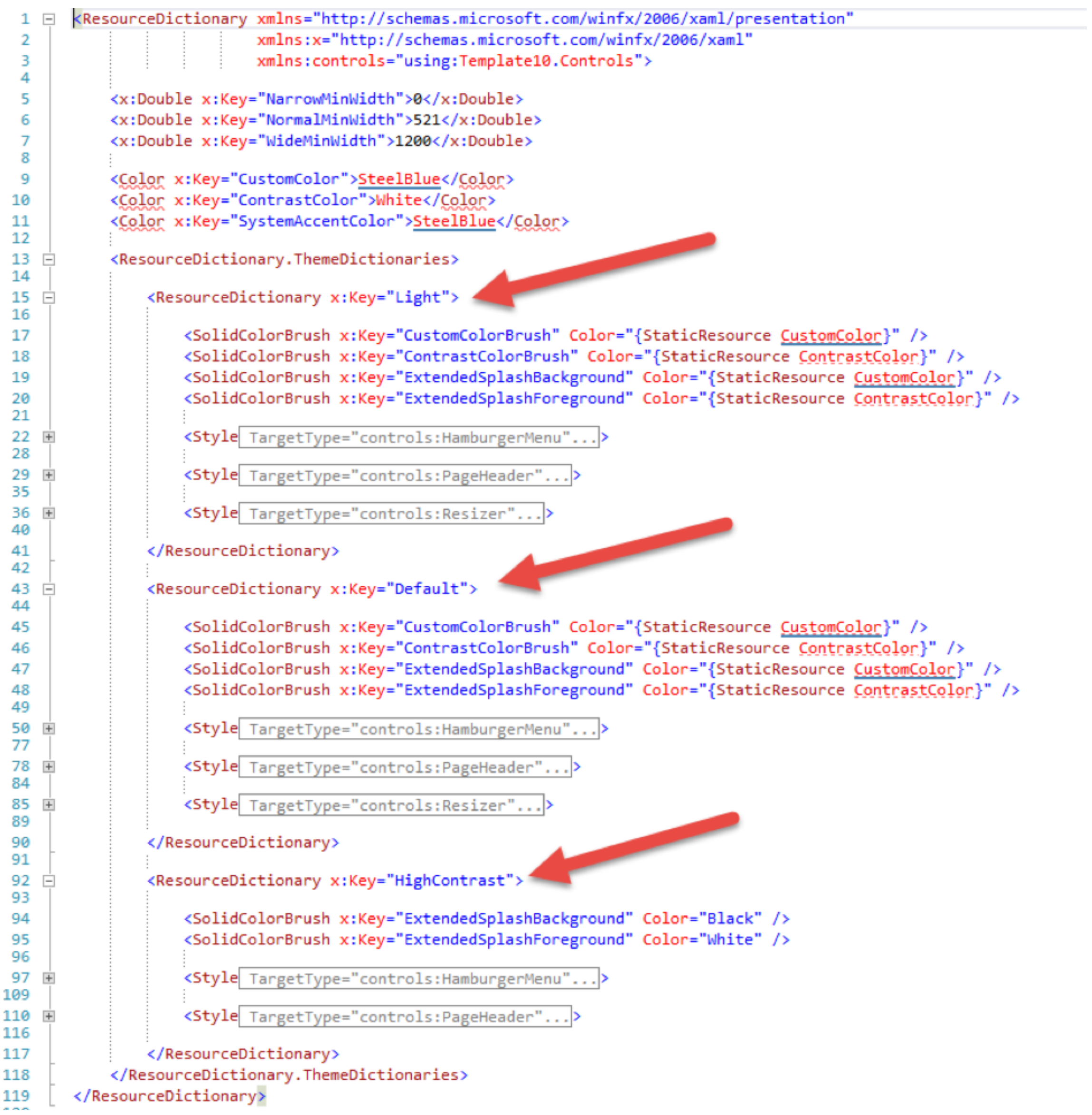
The Services directory also contains a **SettingService** for us. This settings service provides an easy way for you to store application specific settings that will roam across devices. Any other services that we create will reside in this directory.



If you open up the Styles folder, you will find the file, **Custom.xaml**.

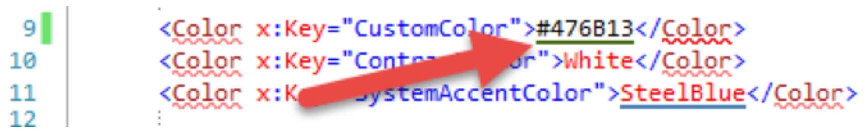


Custom.xaml defines the styles for your application. If you open it up, you will notice there are 3 main themes defined: Light, Default (Dark) and HighContrast.



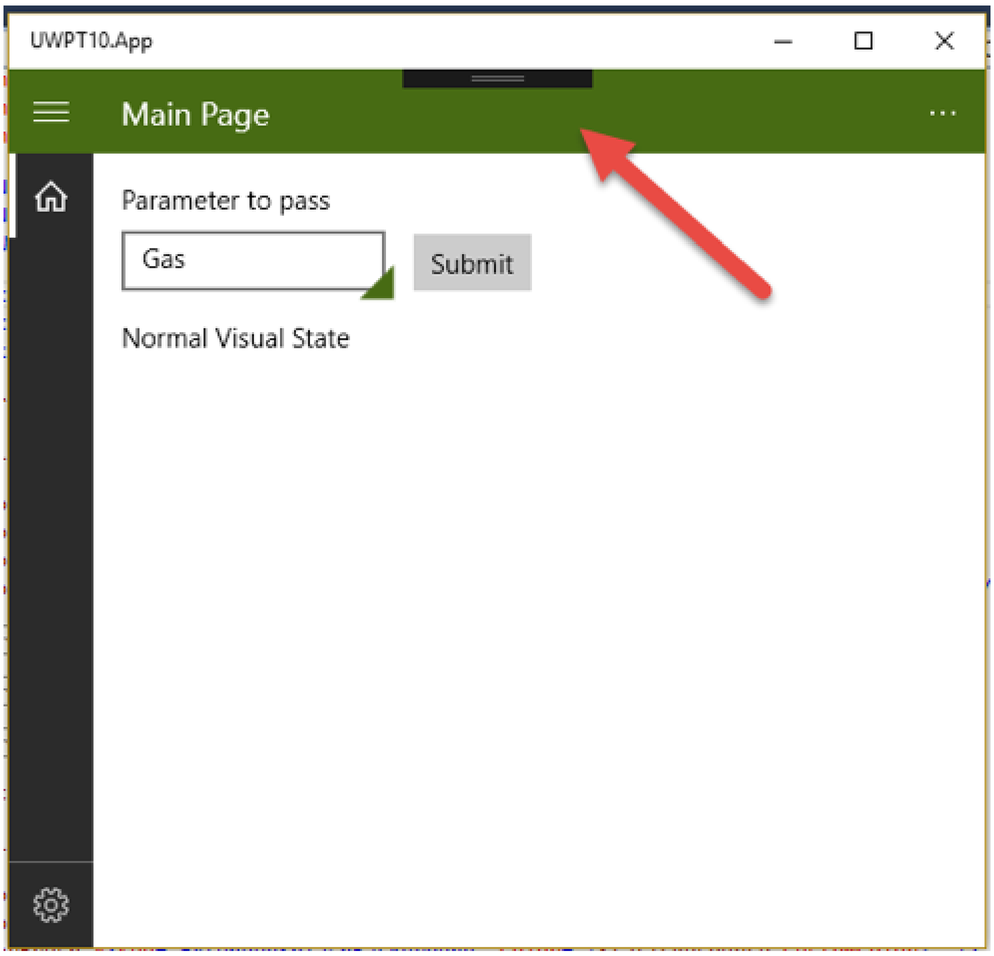
```
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
                    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
                    xmlns:controls="using:Template10.Controls">

    <x:Double x:Key="NarrowMinWidth">0</x:Double>
    <x:Double x:Key="NormalMinWidth">521</x:Double>
    <x:Double x:Key="WideMinWidth">1200</x:Double>

    <Color x:Key="CustomColor">SteelBlue</Color>
    <Color x:Key="ContrastColor">White</Color>
    <Color x:Key="SystemAccentColor">SteelBlue</Color>

    <ResourceDictionary.ThemeDictionaries>

        <ResourceDictionary x:Key="Light">

            <SolidColorBrush x:Key="CustomColorBrush" Color="{StaticResource CustomColor}" />
            <SolidColorBrush x:Key="ContrastColorBrush" Color="{StaticResource ContrastColor}" />
            <SolidColorBrush x:Key="ExtendedSplashBackground" Color="{StaticResource CustomColor}" />
            <SolidColorBrush x:Key="ExtendedSplashForeground" Color="{StaticResource ContrastColor}" />

            <Style TargetType="controls:HamburgerMenu"...>

            <Style TargetType="controls:PageHeader"...>

            <Style TargetType="controls:Resizer"...>

        </ResourceDictionary>

        <ResourceDictionary x:Key="Default">

            <SolidColorBrush x:Key="CustomColorBrush" Color="{StaticResource CustomColor}" />
            <SolidColorBrush x:Key="ContrastColorBrush" Color="{StaticResource ContrastColor}" />
            <SolidColorBrush x:Key="ExtendedSplashBackground" Color="{StaticResource CustomColor}" />
            <SolidColorBrush x:Key="ExtendedSplashForeground" Color="{StaticResource ContrastColor}" />

            <Style TargetType="controls:HamburgerMenu"...>

            <Style TargetType="controls:PageHeader"...>

            <Style TargetType="controls:Resizer"...>

        </ResourceDictionary>

        <ResourceDictionary x:Key="HighContrast">

            <SolidColorBrush x:Key="ExtendedSplashBackground" Color="Black" />
            <SolidColorBrush x:Key="ExtendedSplashForeground" Color="White" />

            <Style TargetType="controls:HamburgerMenu"...>

            <Style TargetType="controls:PageHeader"...>
        </ResourceDictionary>
    </ResourceDictionary.ThemeDictionaries>
</ResourceDictionary>
```

For example, if you want to change the overall accent color for your application, go ahead and **change** the **CustomColor** color to **#476B13**.



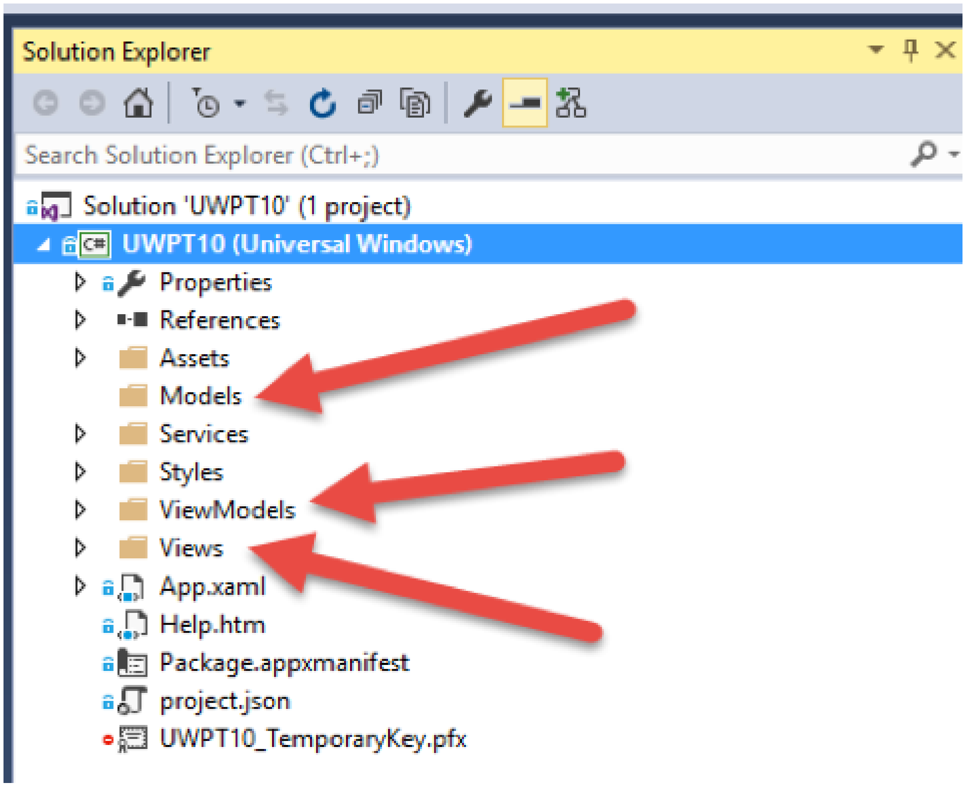
```
<Color x:Key="CustomColor">#476B13</Color>
<Color x:Key="Conte        r">White</Color>
<Color x:K      ystemAccentColor">SteelBlue</Color>
```

Now, **press F5**. Notice the color of the main page header.



MVVM Pattern

If you haven't noticed, the Universal Windows Platform application uses an MVVM pattern. You will notice that there are directories for **Models**, **Views** and **ViewModels**.



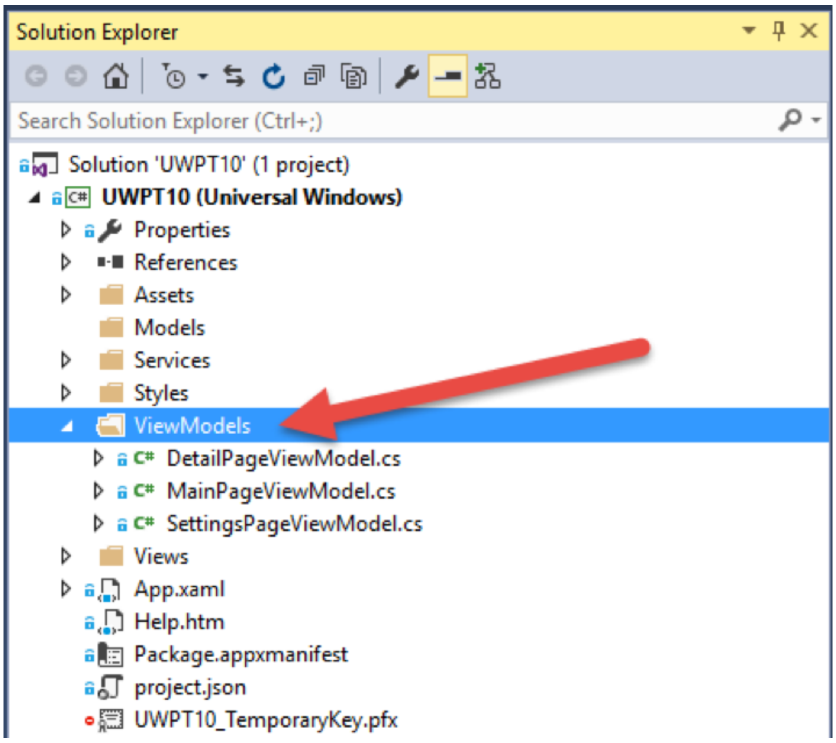
Learn more about the MVVM pattern.

Models

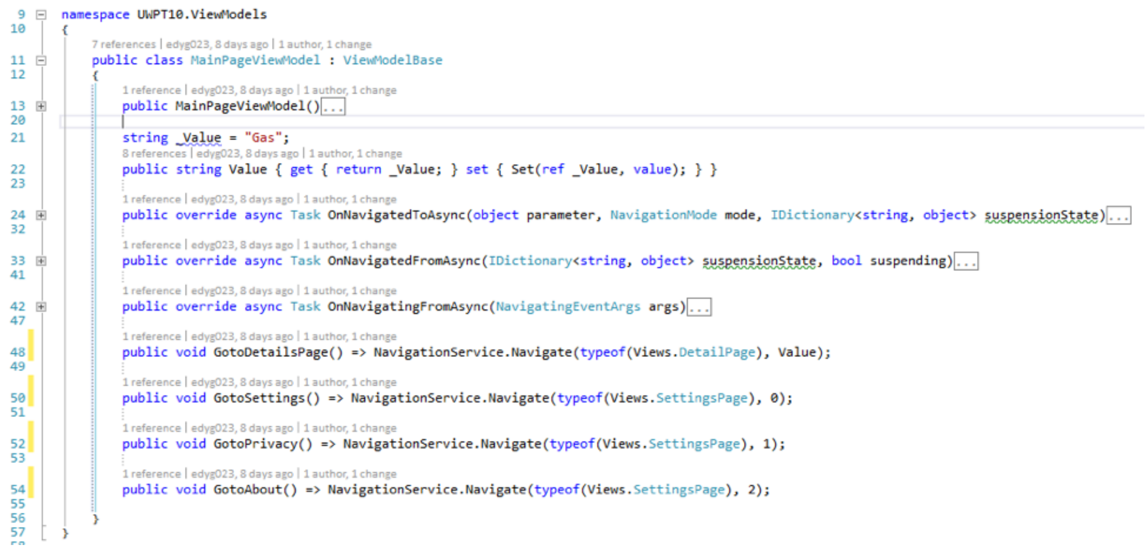
When building a Universal Windows Platform application, models refer to a domain object that represents information for some kind of state. This data generally comes from a database or data access layer. This directory doesn't have anything in it on initial creation since the template has no idea what type of application or business you are creating.

ViewModels

View models, on the other hand are represent an abstraction of view data which is represented by properties and methods. This directory contains all the view models for all the views (screens/pages).

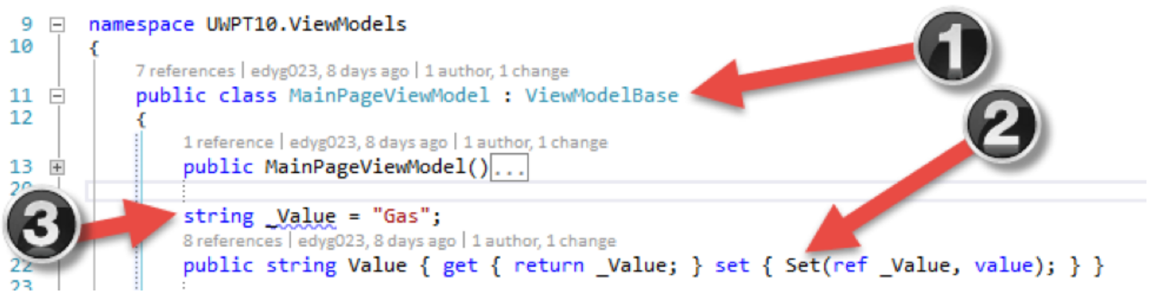


There is one thing to note. If you **open MainPageViewModel.cs**, you will see the following:



```
namespace UWPT10.ViewModels
{
    7 references | edyg023, 8 days ago | 1 author, 1 change
    public class MainPageViewModel : ViewModelBase
    {
        1 reference | edyg023, 8 days ago | 1 author, 1 change
        public MainPageViewModel()...

        string _Value = "Gas";
        8 references | edyg023, 8 days ago | 1 author, 1 change
        public string Value { get { return _Value; } set { Set(ref _Value, value); } }

        1 reference | edyg023, 8 days ago | 1 author, 1 change
        public override async Task OnNavigatedToAsync(object parameter, NavigationMode mode, IDictionary<string, object> suspensionState)...

        1 reference | edyg023, 8 days ago | 1 author, 1 change
        public override async Task OnNavigatedFromAsync(IDictionary<string, object> suspensionState, bool suspending)...

        1 reference | edyg023, 8 days ago | 1 author, 1 change
        public override async Task OnNavigatingFromAsync(NavigatingEventArgs args)...

        1 reference | edyg023, 8 days ago | 1 author, 1 change
        public void GotoDetailsPage() => NavigationService.Navigate(typeof(Views.DetailPage), Value);

        1 reference | edyg023, 8 days ago | 1 author, 1 change
        public void GotoSettings() => NavigationService.Navigate(typeof(Views.SettingsPage), 0);

        1 reference | edyg023, 8 days ago | 1 author, 1 change
        public void GotoPrivacy() => NavigationService.Navigate(typeof(Views.SettingsPage), 1);

        1 reference | edyg023, 8 days ago | 1 author, 1 change
        public void GotoAbout() => NavigationService.Navigate(typeof(Views.SettingsPage), 2);

    }
}
```

However, what I really want you to focus on is a couple of things around the Value property for the view model.



```
namespace UWPT10.ViewModels
{
    7 references | edyg023, 8 days ago | 1 author, 1 change
    public class MainPageViewModel : ViewModelBase
    {
        1 reference | edyg023, 8 days ago | 1 author, 1 change
        public MainPageViewModel()...

        string _Value = "Gas";
        8 references | edyg023, 8 days ago | 1 author, 1 change
        public string Value { get { return _Value; } set { Set(ref _Value, value); } }
```

First, notice that the view model inherits **ViewModelBase**. This is part of the Template10 library.

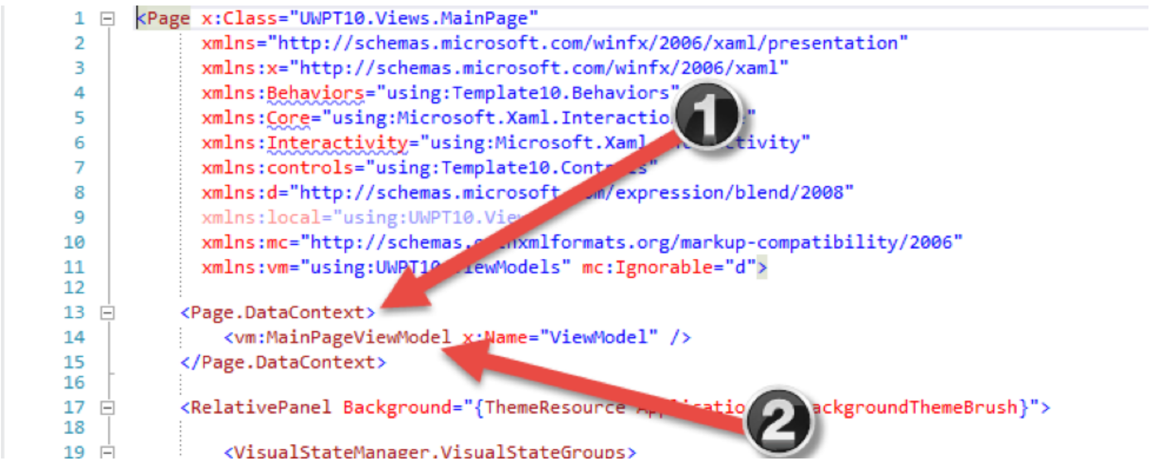
Second, in the setter for Value, you will notice a call to the **Set()** function inherited by ViewModelBase. The Set() function takes care of all your property changed events.

However, if you want to manually raise the property changed event, you can do so by calling the **RaisePropertyChanged()** function, which accepts a property name parameter. For example, RaisePropertyChanged("_Value"); Also, if you are calling it from within the setter for the current property, you simply only need RaisePropertyChanged() without a property name.

Views

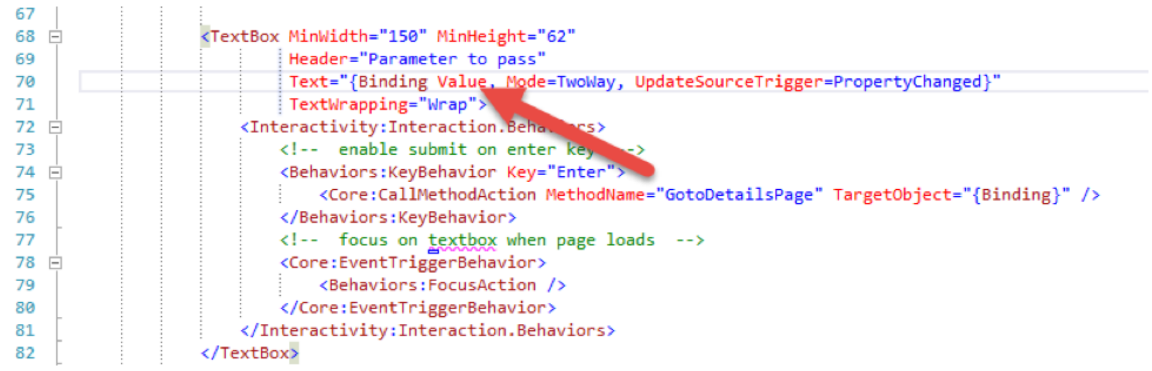
Views are basically a definition of the page/screen layout for your application. It is what the user of the application sees and uses.

One thing to note about the templated views is how the DataContext is set by default in view. Under the **Views** folder, **open MainPage.xaml**. You will notice that there is a **Page.DataContext** element that defines a view model called **ViewModel**. Notice that it's type is **MainPageViewModel**.



```
<Page x:Class="UWPT10.Views.MainPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:Behaviors="using:Template10.Behaviors"
      xmlns:Core="using:Microsoft.Xaml.Interactio[illegible]"
      xmlns:Interactivity="using:Microsoft.Xaml[illegible]tivity"
      xmlns:controls="using:Template10.Cont[illegible]s"
      xmlns:d="http://schemas.microsoft[illegible]m/expression/blend/2008"
      xmlns:local="using:UWPT10.Vie[illegible]
      xmlns:mc="http://schemas.[illegible]nxmlformats.org/markup-compatibility/2006"
      xmlns:vm="using:UWPT1[illegible]iewModels" mc:Ignorable="d">

    <Page.DataContext>
        <vm:MainPageViewModel x:Name="ViewModel" />
    </Page.DataContext>

    <RelativePanel Background="{ThemeResource [illegible]ackgroundThemeBrush}">

        <VisualStateManager.VisualStateGroups>
```

Now, in the code, we can bind our Value property in our view model, MainPageViewModel.



```
            <TextBox MinWidth="150" MinHeight="62"
                     Header="Parameter to pass"
                     Text="{Binding Value, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"
                     TextWrapping="Wrap">
                <Interactivity:Interaction.Beha[illegible]rs>
                    <!--  enable submit on enter key[illegible]-->
                    <Behaviors:KeyBehavior Key="Enter">
                        <Core:CallMethodAction MethodName="GotoDetailsPage" TargetObject="{Binding}" />
                    </Behaviors:KeyBehavior>
                    <!--  focus on textbox when page loads  -->
                    <Core:EventTriggerBehavior>
                        <Behaviors:FocusAction />
                    </Core:EventTriggerBehavior>
                </Interactivity:Interaction.Behaviors>
            </TextBox>
```

Pretty nice huh?

So, to summarize what we accomplished in this blog post:

- We added the Template10 NuGet package.
- Took a look at the structure of the code
- Discussed briefly about the MVVM pattern that is used in the Template10 library.
- Gave an example of View and View Model binding using Template10 MVVM pattern.

Well, the Template10 Library gives us a number of other nice features, but we will discuss those as we need in future posts. For now, you get a pretty good idea how quickly you can build the shell of a Universal Windows Platform application.

In part 2 of this series, before we go crazy with adding more new features, I find it advantageous to put a few things in place before we start slinging around too much code. I will talk a little more about adding a string service for the application. Using T4 Text Templates we will generate a string helper class that will give us intellisense and compile time support.

I hope you enjoyed this post. If so, please share it with others.

3 Comments **Intertech Blog** 1 **Login**

♡ **Recommend** 1 **Share** **Sort by Best**

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Jerry Nixon • 5 months ago

This is a thing of beauty

12 • Reply • Share ›

Matt Edwards • 10 months ago

This was extremely helpful for me today. Sometimes you just need a place to get started and this is a good start. thank you.

1 • Reply • Share ›

Colt Bauman • 3 months ago

Thanks for sharing. I was just searching for recommended folder structure and stumbled upon this gem.

• Reply • Share ›

Subscribe **Add Disqus to your siteAdd DisqusAdd** **Privacy**

Get Our Secret to Good Code Documentation Guide

Subscribe to our blog and gain access to our guide developed by our consulting teams.

Some ad blockers can block the form below.

Email

Subscribe

Recent Posts

Thousands of Passengers Stranded, Windows on Git, and Much More…

Building Full Stack Development Skills on Your Team

DDoS-ing GitHub, Google's ML Training, and More…

Top Trends in Javascript for 2018 and Beyond

SpaceX's Software, Email Overwhelm, and More…

Categories

Categories

Select Category ▼

Practical insights, tips, tutorials and examples from team of software development consultants and trainers.

Recent Posts

- Thousands of Passengers Stranded, Windows on Git, and Much More…
- Building Full Stack Development Skills on Your Team
- DDoS-ing GitHub, Google's ML Training, and More…
- Top Trends in Javascript for 2018 and Beyond
- SpaceX's Software, Email Overwhelm, and More…

Contact Details

1575 Thomas Center Drive
Eagan, MN 55122
General: 651.288.7000

Training: 651-288-7109
Consulting: 651-288-7001
Toll Free: 800-866-9884

About Intertech

Home
Consulting
IntertechU Course Selector