**XAML Conditional Compilation**

Conditional compilation feature is primarily used to include or exclude a portion of source code file based on certain condition. In a typical C# program, you can use preprocessor directives to perform conditional compilation as shown below –

public double MyMethod(int input)

{

double result;

#if CONDITION

result = input \* 0.5;

#else

result = input \* 0.10;

#endif

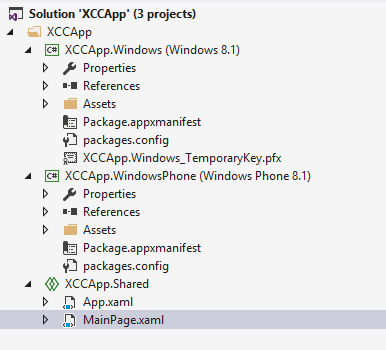
return result;

}

If the CONDITION is set, result = input \* 0.5 code will get compiled otherwise result = input \* 0.10 will be compiled. Such type of conditional compilation is commonly used if you want to isolate platform specific [Windows Phone, Windows Store] code, or if you want to execute certain code based on deployment configuration [Debug / Release] etc. Conditional compilation is baked-in in C# language, however that’s not the case in when it comes to XAML. But before dive into implementing conditional compilation in XAML, let’s understand why do we need it in XAML specifically?

During //Build 2014 event Microsoft introduced Universal App which allow you to build both Windows Phone and Windows Store app at the same time by sharing as much code as possible. The shared code can be C# class files, user controls, resources etc. Universal App project template by default creates a *Shared Project* within Visual Studio which stores all the common code across multiple projects. You can even add a XAML user control into the shared project, which then can be used by both Windows Phone and Windows Store apps. However, if you need to display platform specific layout [controls, color etc.] using the *same* XAML file, you need a way to perform conditional compilation. Unfortunately XAML conditional compilation is not baked-in in the platform yet, but that’s where open source projects like [‘XAML Conditional Compilation’ [XCC]](https://xcc.codeplex.com/) shines. As the name suggests, XCC allows you to conditionally compile XAML files. It’s available via NuGet package, however keep it mind that it’s a proof of concept and you need to use it at your own risk.

So now we have the context set, it’s time to get our hands dirty and build a very basic Universal App and use XCC for conditional compilation of XAML file. I assume you already know how to create a ‘Blank Universal App’ in Visual Studio, so that we can just focus on XCC. Below diagram shows the solution structure of XCCApp Universal App. The XCCApp.Shared project contains the shared code files between XCCApp.Windows and XCCApp.WindowsPhone projects. Please note, by default Visual Studio creates MainPage.xaml file in both Windows Store and Windows Phone projects. Since we are sharing XAML in this code example, I have moved it to Shared project.



First thing you need to do is install XCC using NuGet in both Windows Phone and Windows Store App.



Unlike most of the other NuGet packages, installing XCC won’t add any dll reference to the project, instead it adds MSBuild target file [xcc.targets] and Xcc.dll file in the packages folder. MSBuild refers to these files during project compilation and includes / excludes appropriate code depending on the condition set. Now let’s understand how to set these conditions in XAML.

In the code snippet below, we have defined two additional namespaces xmlns:store="condition:WINDOWS\_APP" and xmlns:wp="condition:WINDOWS\_PHONE\_APP". The condition attribute refers to the conditional compilation symbol defined in both Windows Store and Windows Phone projects.

Next we have defined two instances of StackPanel using store and wp prefix. As you might have gussed, store:StackPanel will be shown only in Windows Store App and wp:StackPanel will be shown only in Windows Phone App.

<Page

x:Class="XCCApp.MainPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="using:XCC"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:store="condition:WINDOWS\_APP"

xmlns:wp="condition:WINDOWS\_PHONE\_APP"

mc:Ignorable="d store wp">

<StackPanel>

<store:StackPanel Background="Yellow" Height="200">

</store:StackPanel>

<wp:StackPanel Background="Green" Height="200">

</wp:StackPanel>

</StackPanel>

</Page>

It’s that easy to conditionally compile XAML using XCC. It surely helps to isolate platform specific code without adding any complexity to the solution. Hope you liked it.