

Thinking in XAML for the Uno Platform – Part 1

Introduction

Copyright © Billy Hollis 2016-2019. All rights reserved. Do not copy, republish, reproduce, or distribute. Posting of this content to the web by anyone except the author is strictly forbidden by the copyright.

The Uno Platform is fairly new, but XAML isn't. This book is aimed at developers who:

- Would like to work with the Uno Platform, at least to try it out, but have not learned the fundamentals of XAML in an earlier platform, or
- Gained a shallow understanding of XAML – just enough to spoof older UI designs – but would like to leverage XAML in the Uno Platform to create far more compelling user interfaces

Basics of the Uno Platform

Before getting hip deep in XAML, let's cover some of the basic things you need to know about the Uno Platform.

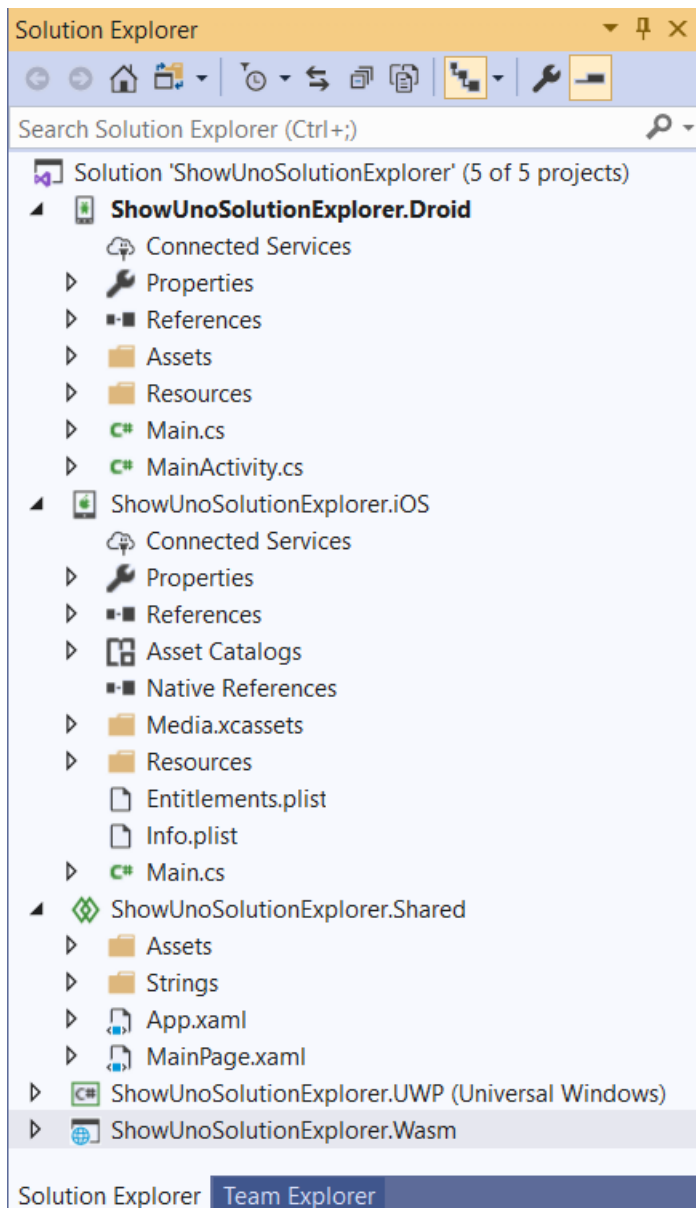
The first XAML platform, Windows Presentation Foundation, was released in 2007. Subsequent platforms include Silverlight, and the Universal Windows Platform. A more restricted form of XAML is also used in Xamarin Forms¹, which is another multi-platform alternative.

The Uno Platform extends XAML to more devices than any previous XAML platform. You can create one app, and expose it to any combination of the following environments:

- Android
- iOS
- Windows 10
- Common browsers (using WebAssembly)

Uno Platform projects are developed in Visual Studio 2017 or 2019, running on Windows 10. A typical Solution Explorer for an Uno project shows the versatility:

¹ Xamarin has been the most common choice for doing multi-platform development the last few years, and it does get the job done. However, I find Uno Platform to have a more comprehensive capability for developing cross-platform UI, with fewer obscure details I need to know to use it. That results in faster and more maintainable development, plus the option of running an app in the browser. For iOS and Android, Uno actually provides a layer on top of Xamarin to cover up some of the complexity of Xamarin.



You can see that there are separate projects for the platforms I listed earlier, plus one project for views and code that will be shared among the platforms. For many simple apps, you'll do most of your work there, but if it becomes necessary to vary your app for the different platforms, each one has a project in which to do that. (To save space, the projects for UWP and WebAssembly/Browser are collapsed in the above screen shot.)

To start developing in Uno Platform, follow the steps at:

<https://platform.uno/docs/articles/get-started.html>

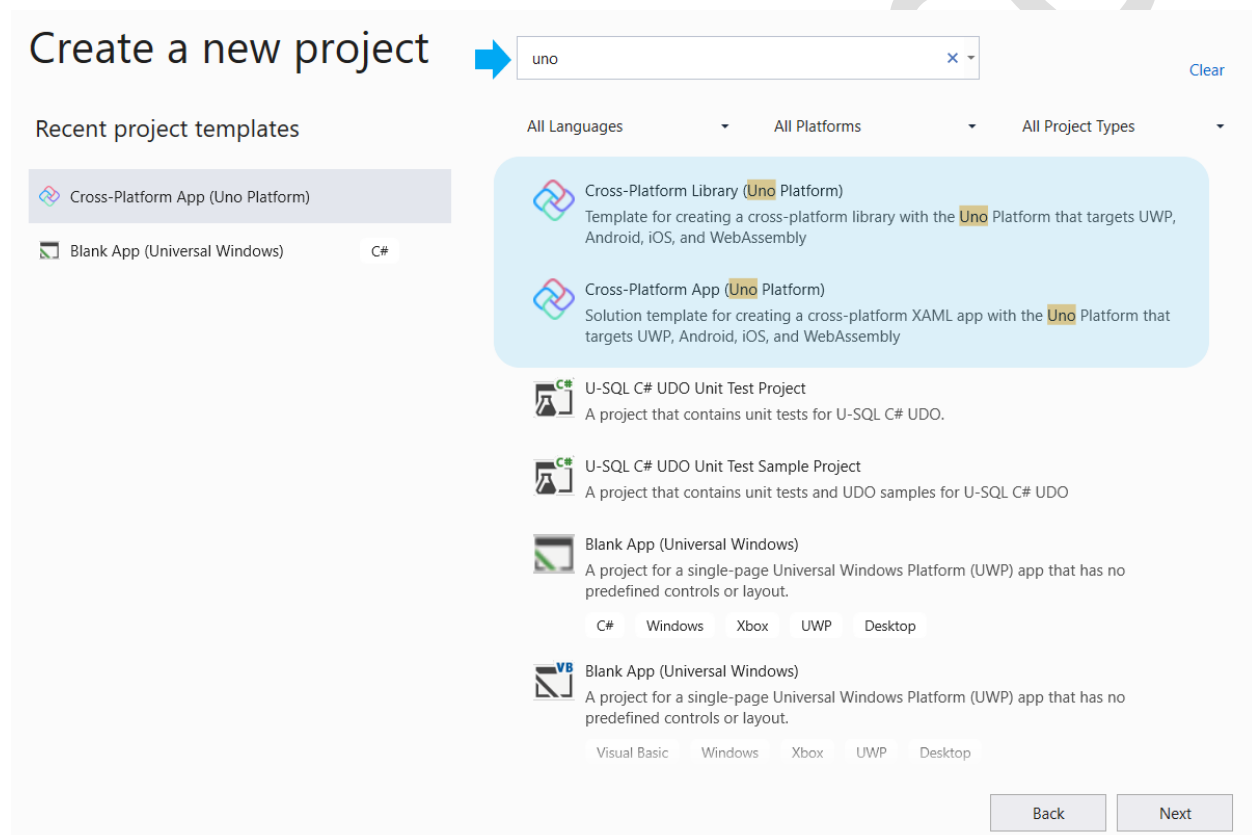
Here is a quick overview of the major steps:

- Get Visual Studio 2017 (15.5 or later) or Visual Studio 2019 if you don't already have one of those

- Install the Uno templates for Visual Studio
- Take care of any other necessities, such as installing the right Window SDK

It's pretty straightforward. I've installed several other platforms such as Google Flutter, and installing Uno was child's play compared to that.

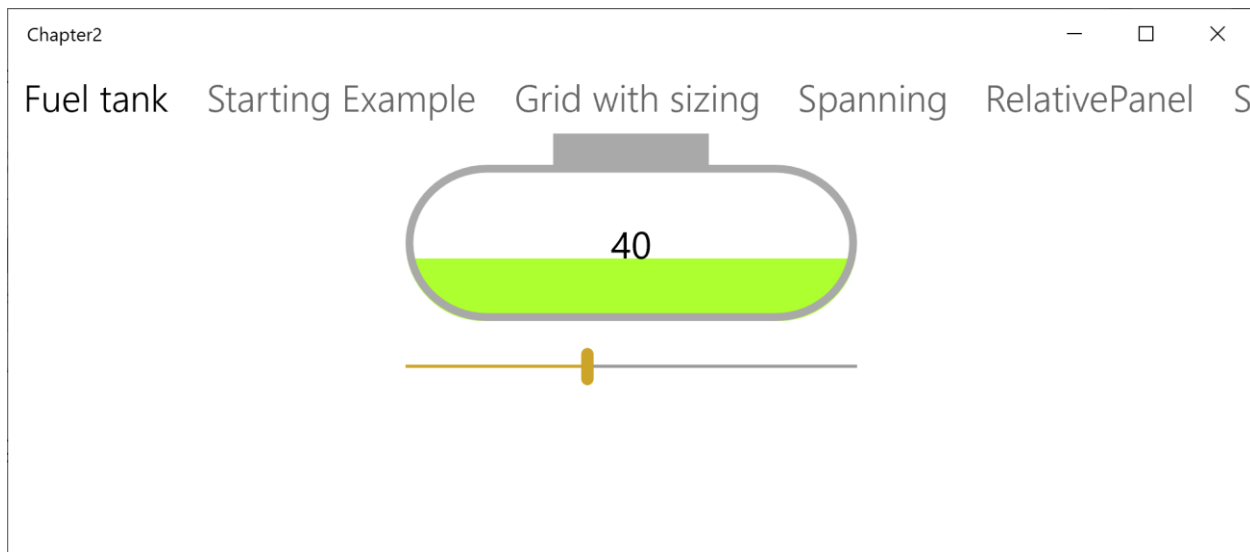
Once you've finished installation, you'll see some new options for project templates. In the Create a New Project dialog for Visual Studio, you can search for "uno" in the search box, and you'll be able to see those two new templates. One template is for creating an app, and the other is for creating a library that can be shared among apps. Here's what the dialog looks like at that point, with the two Uno Platform templates highlighted in blue:



If you select the one for "Cross-Platform App (Uno Platform)", then give it a name, let Visual Studio grind a bit, and you're off to the races.

In this book, I'll do all my work in the shared project. I'll put views and classes that I need to illustrate concepts there.

All of my examples are included in the code downloads for the book. I gather together the examples for each chapter in one solution, using a Pivot control to allow access to the chapter's examples one at a time. Here is a screen shot of the app with examples for Chapter 2, with one of the examples selected at the top:



To see an example in action, you just choose it in the Pivot control at the top. Each example is typically in its own UserControl in the shared project. You can open and examine that UserControl to inspect the XAML for the example, and of course play around with it all you like.

This book also includes the XAML for the examples in the text. In many examples, the XAML is in separate snippets as the example is built up and explained, step-by-step.

That means the XAML in the text is scattered around a bit for instructional purposes. If you want to get the final working XAML, you'll probably prefer to open the solution for the chapter for that example, and navigate to the appropriate UserControl. I hope I've given them all descriptive and evocative names so you can easily find the one you want.

[Deploying Uno Platform apps](#)

For all platforms except the browser, the most common form of wide deployment of an app developed on Uno is the appropriate app store. That means you'll need an account for each device store you intend to target.²

In addition, to compile and deploy on iOS, you'll need a Mac. That isn't an Uno thing, it's an Apple thing – other cross-platform tools have the same requirement.

From testing Uno apps in Visual Studio, you select the target in the top toolbar, then select a device if necessary, and fire it off with F5 or by clicking the device.

For a browser target, just make sure you are targeting a browser that supports WebAssembly. That includes all the ones you're likely to have lying around. Keep in mind that WebAssembly is still a young technology, so you're more likely to see glitches of various kinds when running your Uno apps in the browser.

² You don't need accounts at any store to develop and test. They're only needed when you are ready to do wide deployment.

When targeting UWP, you don't have to worry about configuring anything. Because of that, the solutions in the code downloads have UWP as the default target when you load them.

For targeting Android, you'll need to have either an Android emulator configured, or an Android device attached. Configuring an Android emulator is an arcane art, and I don't have the space to go into it here. It's the same as doing it for Xamarin and other cross-platform tools, though, so there are plenty of web resources on how to do it. The most likely problem you'll run into is the need to configure your emulator with more memory, because default Android emulators are notoriously chintzy with memory.

That's about all you need to know to start learning more about XAML on the Uno Platform. There's plenty more to learn about Uno Platform of course, but I'm going to focus in this book on creating compelling, intuitive user interfaces in your Uno apps, by giving you the necessary understanding of XAML.

Thinking in XAML

Learning to use XAML to create really compelling user experiences means learning how to think about it the right way. That's where the title of the book comes from.

Why is that necessary? Because XAML has a major difference from most other UI technologies. Effective use of XAML requires putting together fine-grained pieces with narrow responsibilities. That gives enormous flexibility, but you must learn to think in a *compositional* fashion to get that flexibility. You must first understand what the pieces do, then understand the various ways they can hook together, and finally get some practice and experience at building user experiences with XAML. This book is intended to help you through the first parts.

For example, Chapter 2 is called Putting Pieces Together, and is about the composition model in XAML. Chapter 3 is called Useful Things, and highlights many of the most useful pieces you can compose.

For those who don't know me, I've been around the XAML world since the earliest bits leaked out of Microsoft. My consulting company has focused on XAML applications for about twelve years now. I've worked on so many XAML apps I can't remember them all. I've done dozens of sessions on XAML at major conferences, and trained teams all over the world in using it.

In this book, I'd like to give you my most important insights about developing user experiences with XAML, by presenting it in the context of the Uno Platform.

Part 1 of the book covers the essentials for using XAML. It leans towards being conceptual instead of trying to describe a lot of features. It covers basic visual elements, layout, composition, and how to work with data via binding.

Part 2 of the book fills in some of the most important areas you need to write compelling apps. That includes graphical elements such as color, translucency, and shapes. It also includes styling options and visual transformations. It concludes with another list of useful things for your personal toolbox.

The API for the XAML in Uno Platform is a large subset of the API for XAML on the Universal Windows Platform. That means the API is big – really big. It's impossible to cover it all, and I'm still learning details after over a dozen years of work with XAML. So this book doesn't try to cover the API in detail. Instead

we'll focus on how to *think* about XAML. If you learn that, filling in details you need for a particular need is dramatically easier.

Let's get started. The first chapter presents a high-level overview of XAML, and assumes you don't know anything about it.

Pre-release