

LAB 3



Get Familiar with the Windows Store App Development Tools

Overview

In this lab you will look at Visual Studio, the main tool for creating Windows Store Apps.

Requirements

To complete this lab you will need:

- Windows 8.1 Pro
- Visual Studio 2013 (any version, including *Express*)
- Internet connection

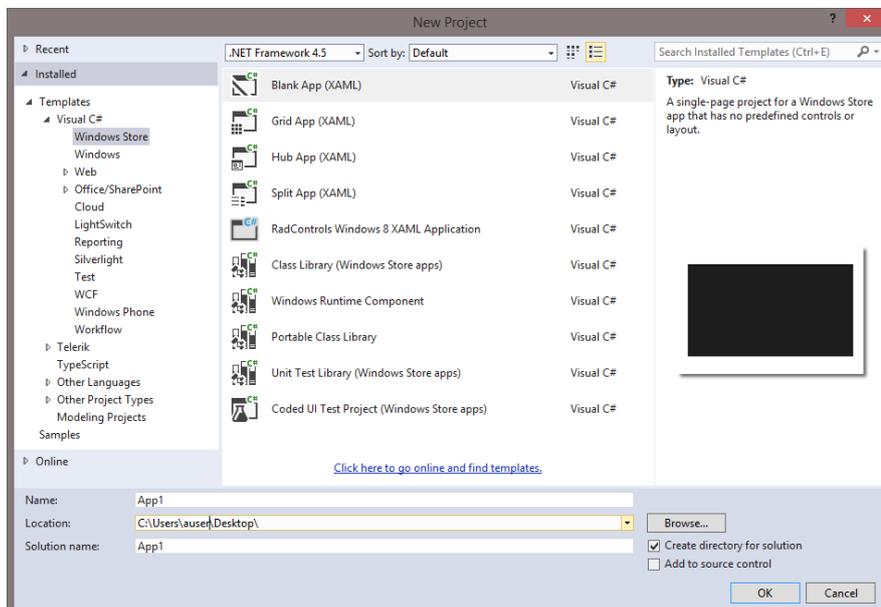
Duration

30 minutes or less

Step 1: Creating a Windows Store Project

Launch Visual Studio. From the menu on the right, select 'New Project'.

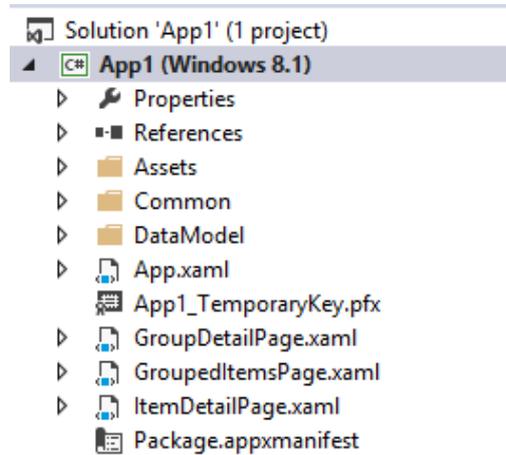
The list of project types included web-based projects (i.e. ASP.NET), desktop projects (e.g. WPF), and more. Select the 'Windows Store' node. You should see a list of templates for Windows Store apps.



Select each on in turn to see an explanation of what it provides, then select the 'Grid App' to create a new 'grid-based app'.

Step 2: Explore the Project

Notice that the project includes several files:

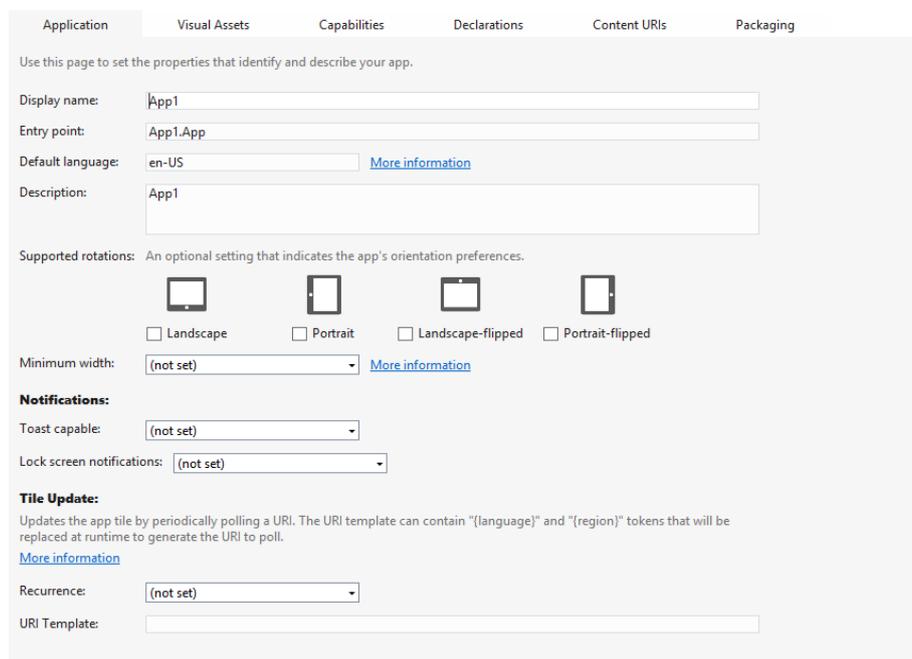


The XAML files define the layout and look of the app pages. Each is accompanied by a 'code behind' file such as 'MyPage.xaml.cs', which you can use to add logic to the app. It is common in advanced apps to use a development pattern called MVVM to store the logic in separate files and not in the code behind. This makes it easier to test the app, but is typically not needed for very simple apps (such as the ones we will create in these labs).

The Assets folder holds graphics for the apps Start Screen tile, etc.

Step 3: Examining the Package Manifest

The manifest defines some important aspect of the app that are used when publishing to the Windows Store:



Application Visual Assets Capabilities Declarations Content URIs Packaging

Use this page to set the properties that identify and describe your app.

Display name:

Entry point:

Default language: [More information](#)

Description:

Supported rotations: An optional setting that indicates the app's orientation preferences.

Landscape Portrait Landscape-flipped Portrait-flipped

Minimum width: [More information](#)

Notifications:

Toast capable:

Lock screen notifications:

Tile Update:

Updates the app tile by periodically polling a URI. The URI template can contain "[language]" and "[region]" tokens that will be replaced at runtime to generate the URI to poll.

[More information](#)

Recurrence:

URI Template:

The Application tab defines such things as which orientations the app supports, its display name, etc.

The Visual Assets tab defines the icons, splashscreen, etc, used by the app. Note that some images are required and some are optional—you can define different sizes of each image and windows will choose the most appropriate one based on the screen resolution.

The Capabilities tab allows the app to declare if it needs to use such things as the Internet, proximity sensors, etc.

The Declarations tab allows the app to declare if it uses features such as File Pickers and background tasks.

The Content URIs tab allows you to define external sites that can notify the app of events.

The Packages tab allows you to set the app version, etc.

Step 4: Build and Deploy the App

From the 'Build' menu, select 'Build Solution'. Once built, select 'Deploy Solution' from the same menu.

This should deploy the app to the system and it should appear on the Start Screen—you may need to look for it in the full list of apps.

Step 5: Test the App

Check that the app works ok once deployed.

To test it properly, use the testing kit supplied with the Visual Studio. Switch to the Start Screen and start typing 'cert'. This should show the 'Windows App Cert Kit' in the list of apps. Select it.

Choose 'Validate Windows Store App' from the options, then choose your app from the list of those available. Select all of the available tests.

Once the testing is in progress do not interact with the computer. Moving the mouse, launching other apps, etc can interfere with the test.

Once complete, examine the report to see if there are any problems with the app.

You should run this test against any apps you intend to submit to the Windows Store to ensure there are no problems that would result in it being rejected.

Step 6: Explore More.

Feel free to try out the other templates in Visual Studio and to poke around at the available settings.

Congratulations! You have used Visual Studio to build, deploy and test a Windows Store app.