Setting up Windows 8 Developer Environment

Here are the steps required to get you started on adding Bluetooth connectivity to Windows 8 Metro applications.

- Download and install Visual Studio 2012 Professional
- Obtain developer license for windows store apps. This will be used for creating store association for your app and creating meta data.
- Compile, Build and Install custom profile drivers
- Compile, Build and Run Metro applications that talk to the custom profile drivers

Pre- Requisites

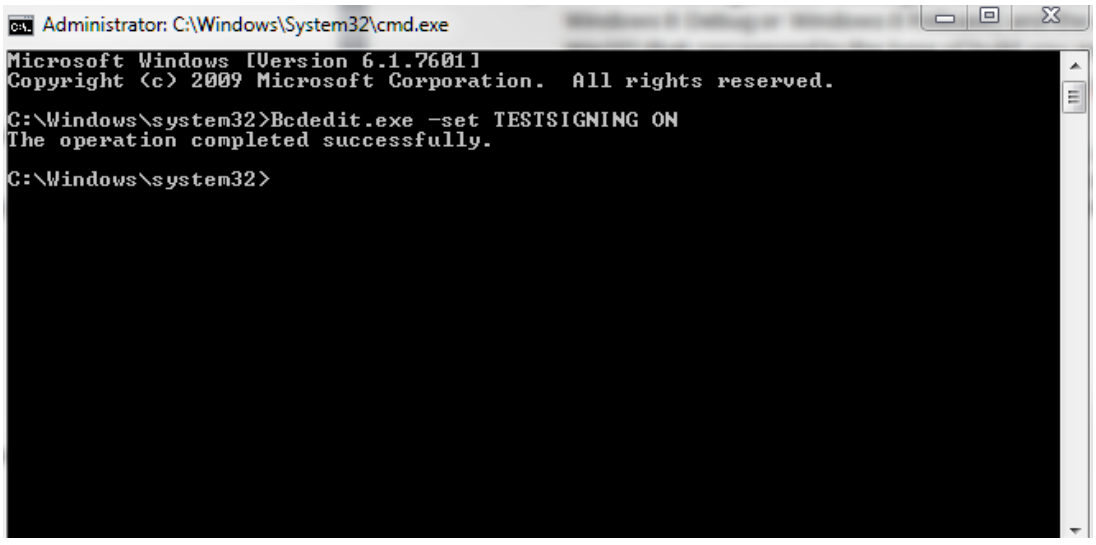
Installations & hardware

- Make sure you have download the quick start kit and created a folder called 'Bluetooth' where you can store and maintain your downloads.
- Windows 8 machine with a Bluetooth 4.0 controller
- Dual mode Bluetooth 4.0 controller (usually comes integrated on all latest Win 8 devices. It can also be purchased from Bluetooth SIG store)
- Install Visual Studio 2012 Professional
- Install WDK and redistributable components
- Make sure you have a developer license

Test Signing

Make sure you have turned on Test signing. Go to Command Prompt with administrative privileges

```
Bcdedit.exe -set TESTSIGNING ON
```



Secure Boot Policy

1) Go to Settings->Change PC Settings -> General
2) Scroll down to Advanced Startup -> Restart Now
3) On Restart, Select Troubleshoot ->Startup Settings ->Restart
4) Select “Disable driver signature enforcement”

Profile Driver

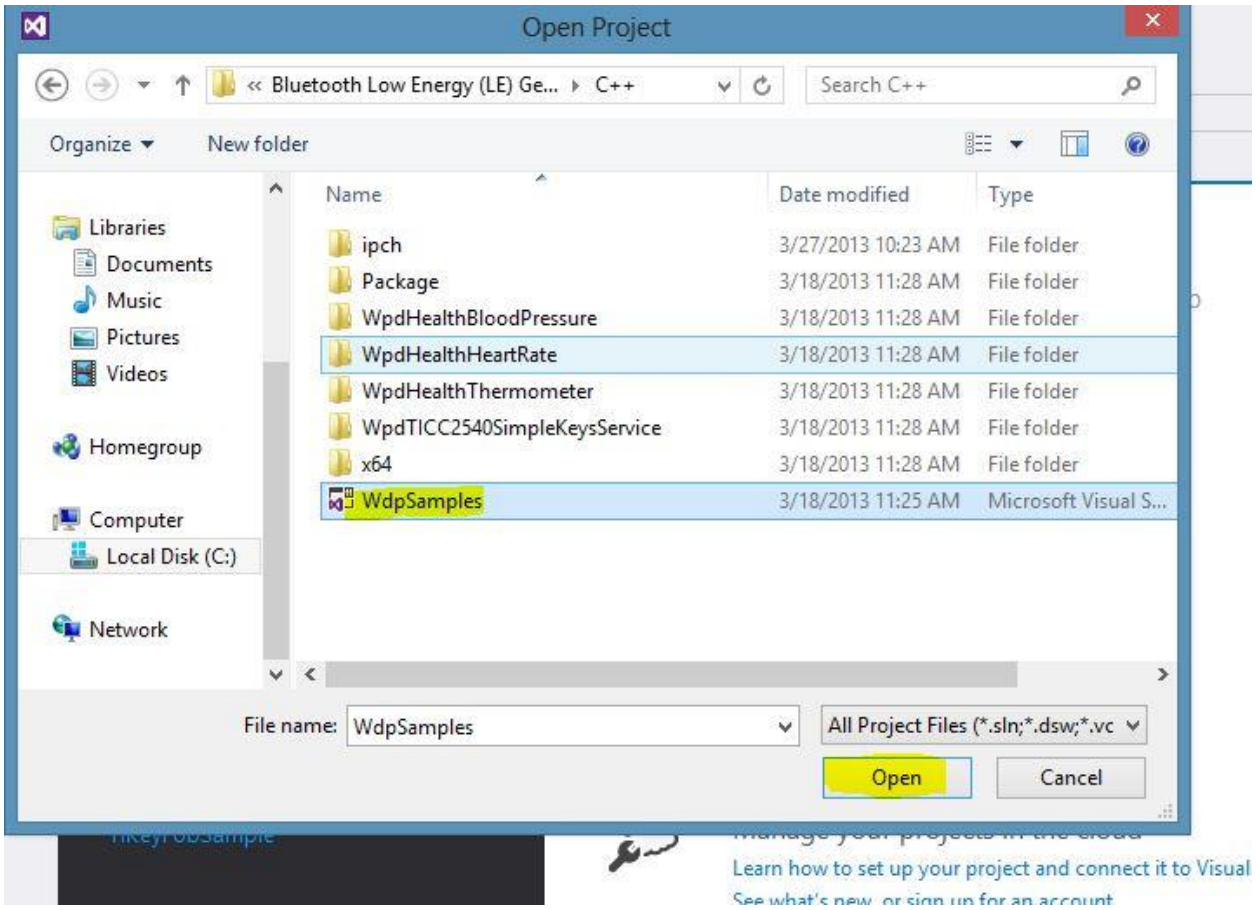
We will go over how to

- Compile and build your custom profile driver
- Discover devices and install custom profile drivers
- Create metadata package and associate Metro applications to use them

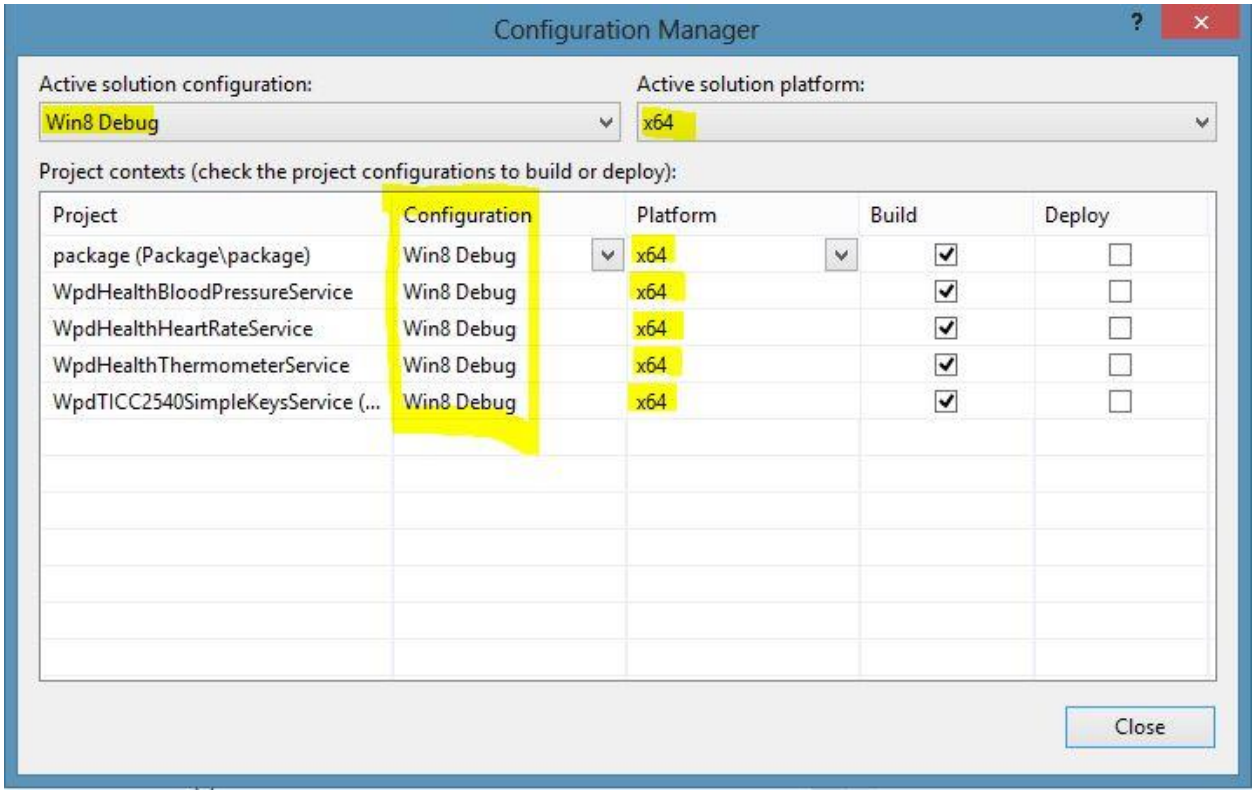
Compiling and Building custom profile driver

Building the sample using Visual Studio

1. Open Visual Studio. From the **File** menu, select **Open Project/Solution**. Within your WDK installation, navigate to src\bluetooth\GATT\Wpd and open the WpdSamples.sln project file.



2. Right-click the solution in the **Solution Explorer** and select **Configuration Manager**.
3. From the **Configuration Manager**, select the **Active Solution Configuration** (for example, Windows 8 Debug or Windows 8 Release) and the **Active Solution Platform** (for example, Win32) that correspond to the type of build you are interested in.



4. From the **Build** menu, click **Build Solution** (Ctrl+Shift+B).

Previous versions of the WDK used the Windows Build utility (Build.exe) and provided separate build environment windows for each of the supported build configurations. Starting in the Visual Studio Professional 2012 WDK, you can use the Visual Studio Command Prompt window for all build configurations.

Installing custom driver

The following steps are required to install the respective sample drivers:

1. Perform regular pairing operations with the device by going to the Device and Printers folder, Add a Device Wizard, and add the device.

2. Go to Device Manager (Right click on Computer and select Manage).

3. Expand the Bluetooth node.

4. You should see a number of Bluetooth LE Generic Attribute Service nodes there. Select the one that has the following UUID as part of the Hardware ID:

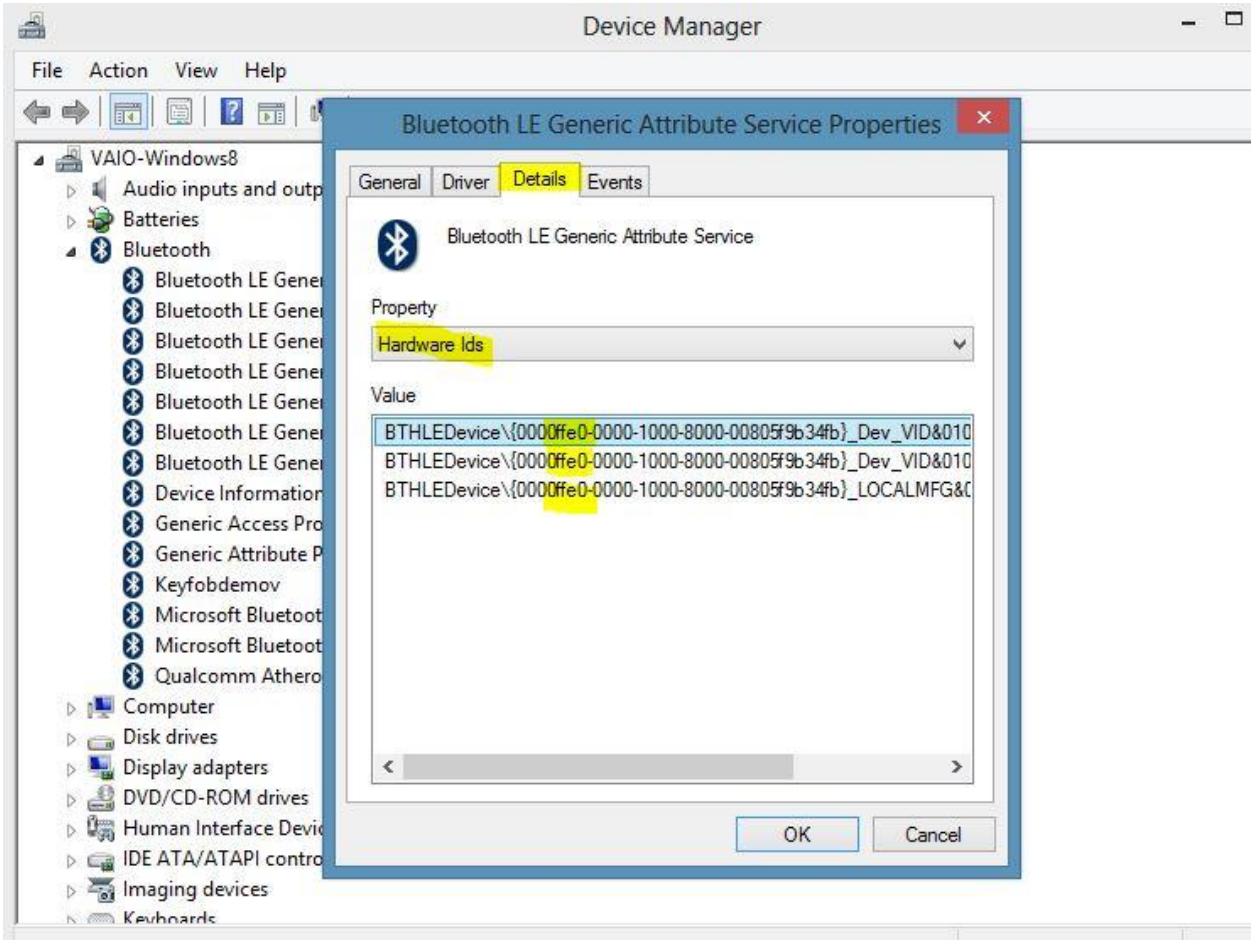
For blood pressure devices use {0000**1810**-0000-1000-8000-00805f9b34fb}

For heart rate devices use {0000**180D**-0000-1000-8000-00805f9b34fb}

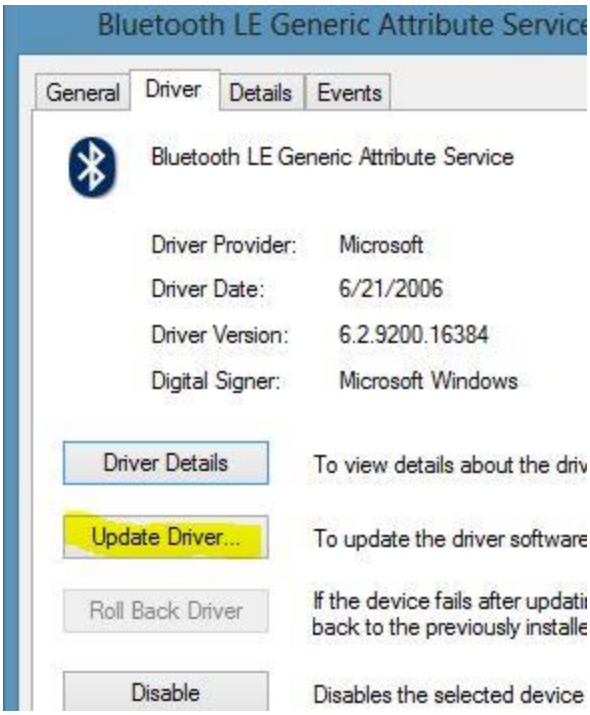
For thermometer devices use {0000**1809**-0000-1000-8000-00805f9b34fb}

For the Texas Instrument's CC2540 Mini Development Kit use {0000**ffe0**-0000-1000-8000-00805f9b34fb}

You can look up the Hardware ID of the node by right clicking on the node and select Properties. Go to the Details tab, and select Hardware Ids in the Property drop down.



5. In the Driver tab of the Properties window, select Update Driver.

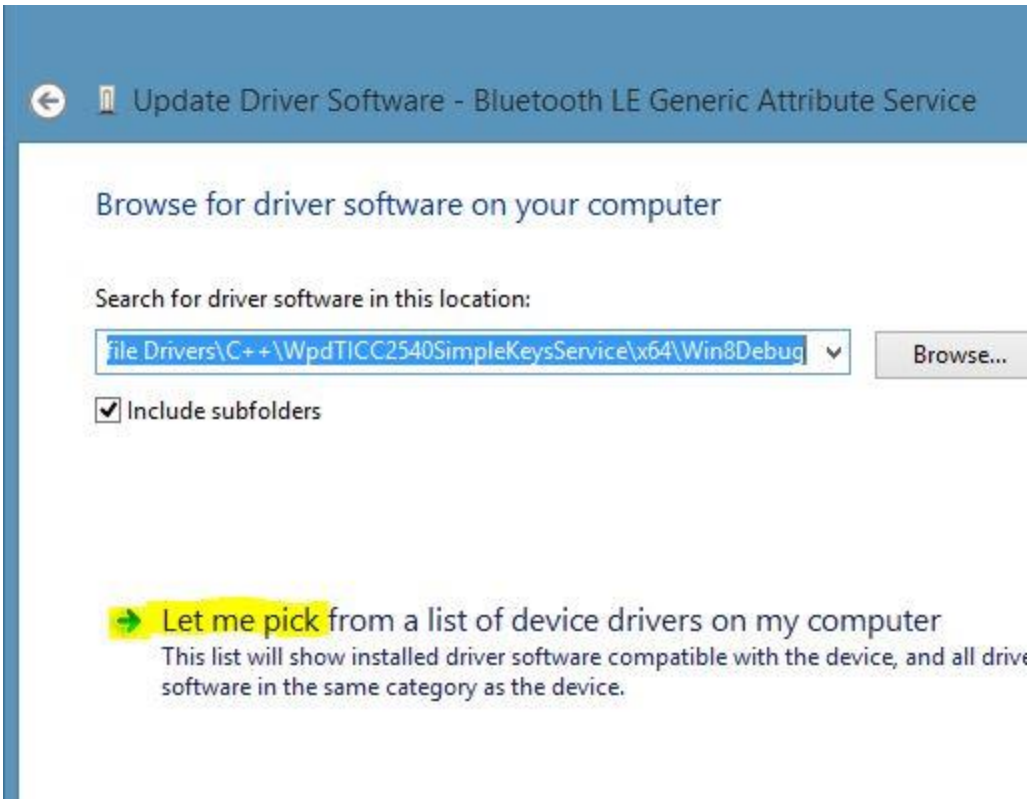


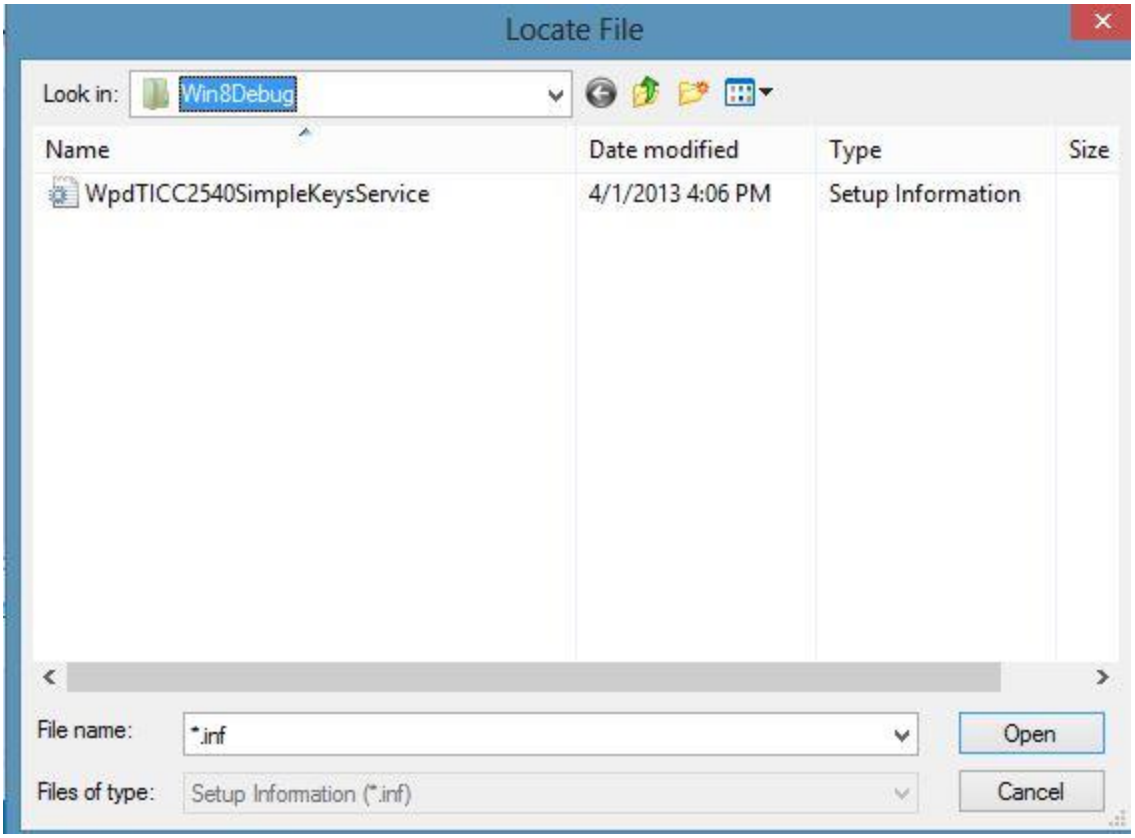
6. Select Browse my computer for driver software.

7. Select Let me pick from a list of device drivers on my computer.

8. Select Have Disk.

9. Provide the path to the directory that contains the *.inf files for the sample driver that you have just built.


Update Driver Software - Bluetooth LE Generic Attribute Service
Browse for driver software on your computer
Search for driver software in this location:
file Drivers\C++\WpdTICC2540SimpleKeysService\x64\Win8Debug
Browse...
Include subfolders
Let me pick from a list of device drivers on my computer
This list will show installed driver software compatible with the device, and all drive
software in the same category as the device.


Locate File
Look in:
Win8Debug
Name
Date modified
Type
Size
WpdTICC2540SimpleKeysService
4/1/2013 4:06 PM
Setup Information
File name:
*.inf
Open
Files of type:
Setup Information (*.inf)
Cancel

To view event information or invoke driver service methods on the installed driver, you can use the wpdinfo.exe tool that ships as part of the WDK.

If you plan to access driver functionality from a Windows Store application, you must also install device metadata for the device you're planning to use.

More on authoring Device Metadata Packages.

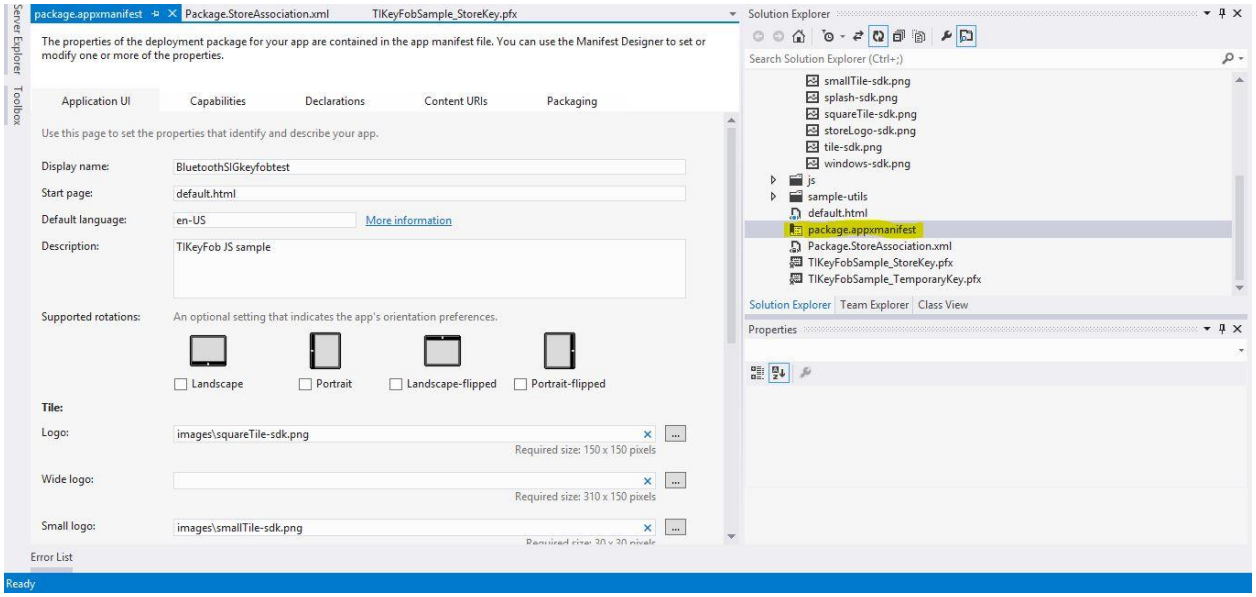
The metadata needs to be deployed locally and, for testing purposes, testsigning needs to be enabled.

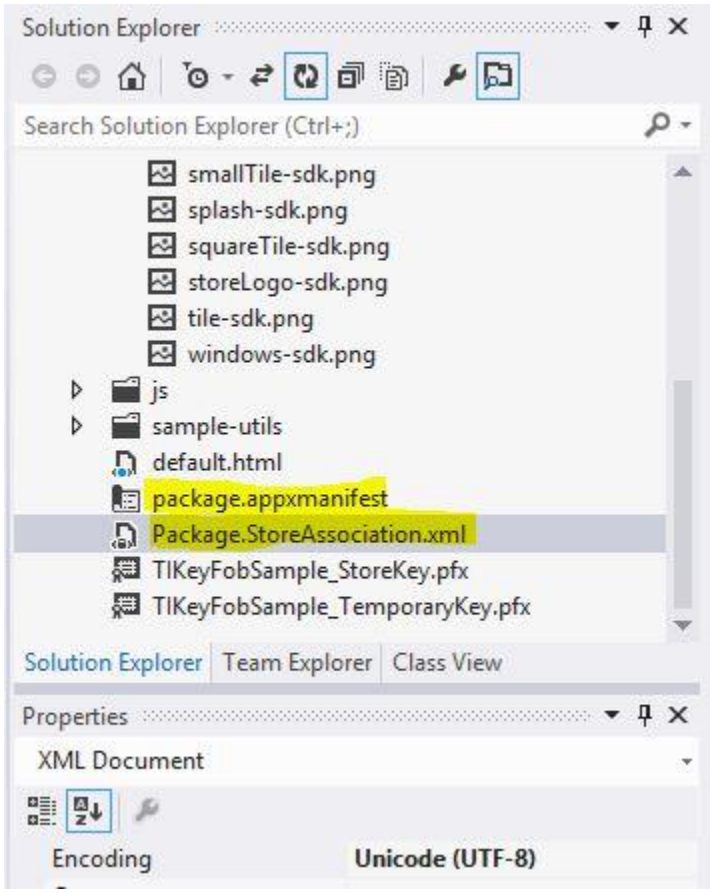
Creating Metadata package

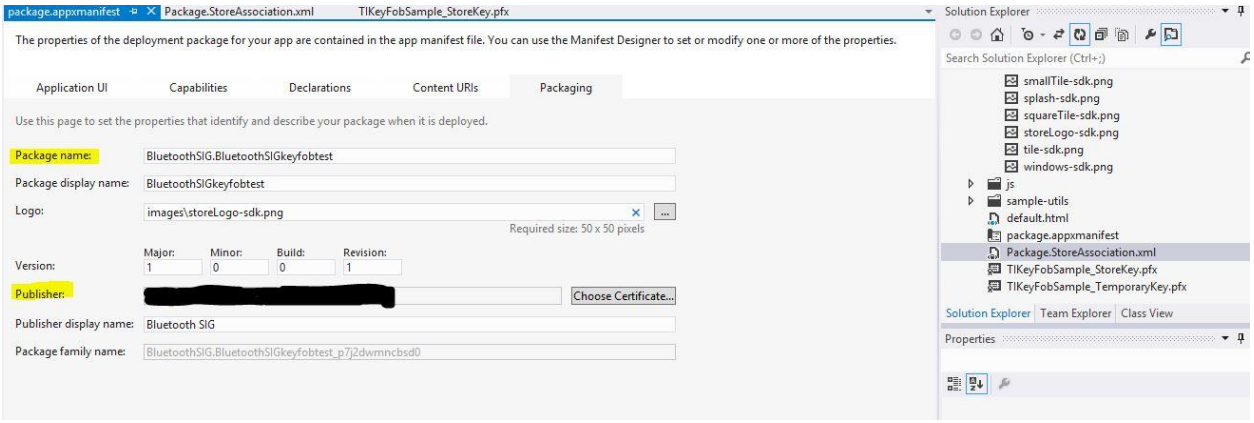
1. First of all you have to have a Live Id that's registered as a developer account (you need one in Visual Studio to develop the app in the first place, so you're probably already set with that).
2. You need to get your app ready for submission, from Visual Studio 2012 select Project->Store->Create App Package, log in with your developer account, and create the app package.
3. One of the side-effects of performing the previous step is that your application's "Package Name" and "Publisher" will change based on your original selection for those fields and your store account.

```
<?xml version="1.0" encoding="utf-8"?>
<StoreAssociation xmlns="http://schemas.microsoft.com/appx/2010/storeassociation">
  <Publisher>[redacted]</Publisher>
  <NamespacePrefix>BluetoothSIG</NamespacePrefix>
  <PublisherDisplayName>Bluetooth SIG</PublisherDisplayName>
  <GeneratePackageHash>http://www.w3.org/2001/04/xmlenc#sha256</GeneratePackageHash>
  <SupportedLocales>
    <Language Code="af" InMinimumRequirementSet="false" />
    <Language Code="af-za" InMinimumRequirementSet="false" />
    <Language Code="am" InMinimumRequirementSet="false" />
    <Language Code="am-et" InMinimumRequirementSet="false" />
    <Language Code="ar" InMinimumRequirementSet="true" />
    <Language Code="ar-ae" InMinimumRequirementSet="true" />
```

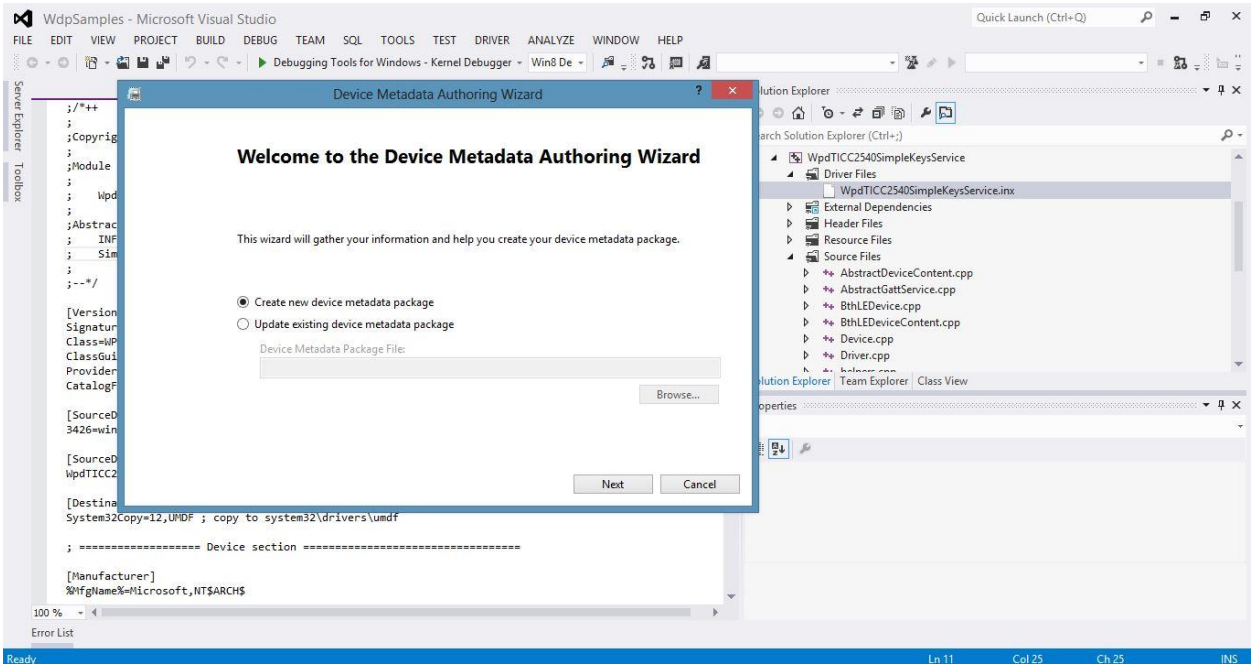
4. Copy the store enabled "Package Name" and "Publisher" to your device metadata "Associations" section (you will find these as well as your App ID in the "package.appxmanifest" file of your app)

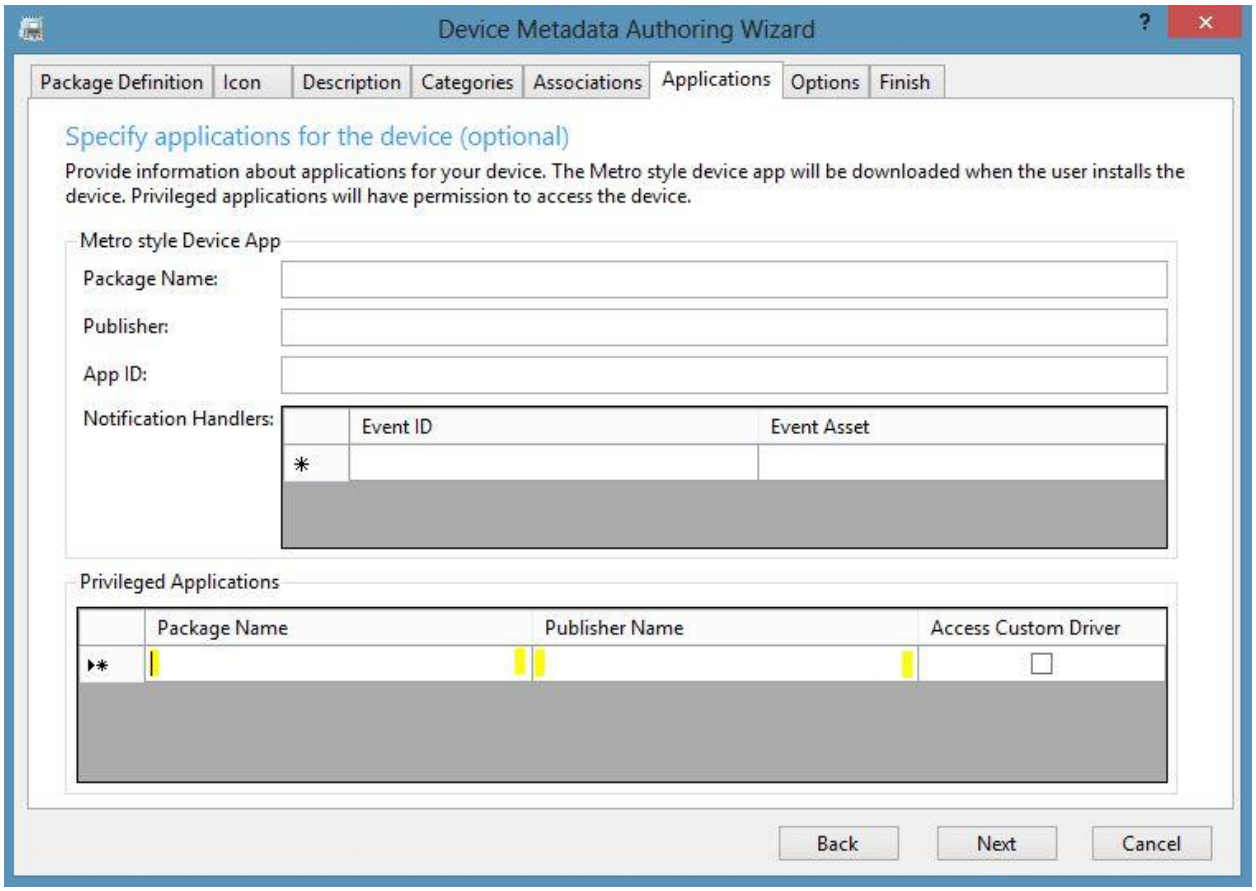

package.appxmanifest
Package.StoreAssociation.xml
TIKeyFobSample_StoreKey.pfx
The properties of the deployment package for your app are contained in the app manifest file. You can use the Manifest Designer to set or modify one or more of the properties.
Application UI
Capabilities
Declarations
Content URIs
Packaging
Use this page to set the properties that identify and describe your app.
Display name:
BluetoothSIGkeyfobtest
Start page:
default.html
Default language:
en-US
More information
Description:
TIKeyFob JS sample
Supported rotations:
An optional setting that indicates the app's orientation preferences.
Landscape
Portrait
Landscape-flipped
Portrait-flipped
Tile:
Logo:
images\squareTile-sdk.png
Required size: 150 x 150 pixels
Wide logo:
Required size: 310 x 150 pixels
Small logo:
images\smallTile-sdk.png
Error List
Ready
Solution Explorer
Search Solution Explorer (Ctrl+;)
smallTile-sdk.png
splash-sdk.png
squareTile-sdk.png
storeLogo-sdk.png
tile-sdk.png
windows-sdk.png
js
sample-utils
default.html
package.appxmanifest
Package.StoreAssociation.xml
TIKeyFobSample_StoreKey.pfx
TIKeyFobSample_TemporaryKey.pfx
Solution Explorer
Team Explorer
Class View
Properties

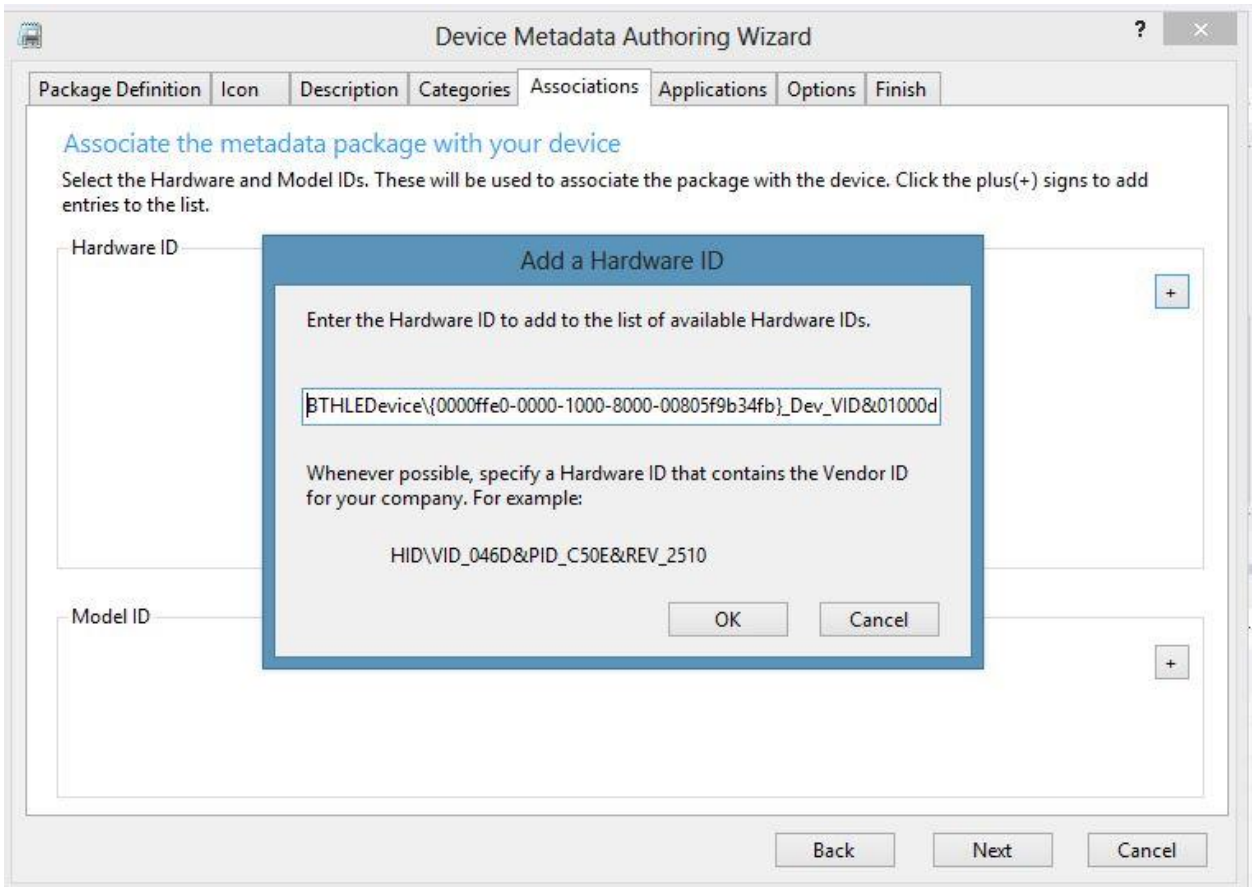

Solution Explorer
Search Solution Explorer (Ctrl+;)
smallTile-sdk.png
splash-sdk.png
squareTile-sdk.png
storeLogo-sdk.png
tile-sdk.png
windows-sdk.png
js
sample-utils
default.html
package.appxmanifest
Package.StoreAssociation.xml
TIKeyFobSample_StoreKey.pfx
TIKeyFobSample_TemporaryKey.pfx
Solution Explorer
Team Explorer
Class View
Properties
XML Document
Encoding
Unicode (UTF-8)

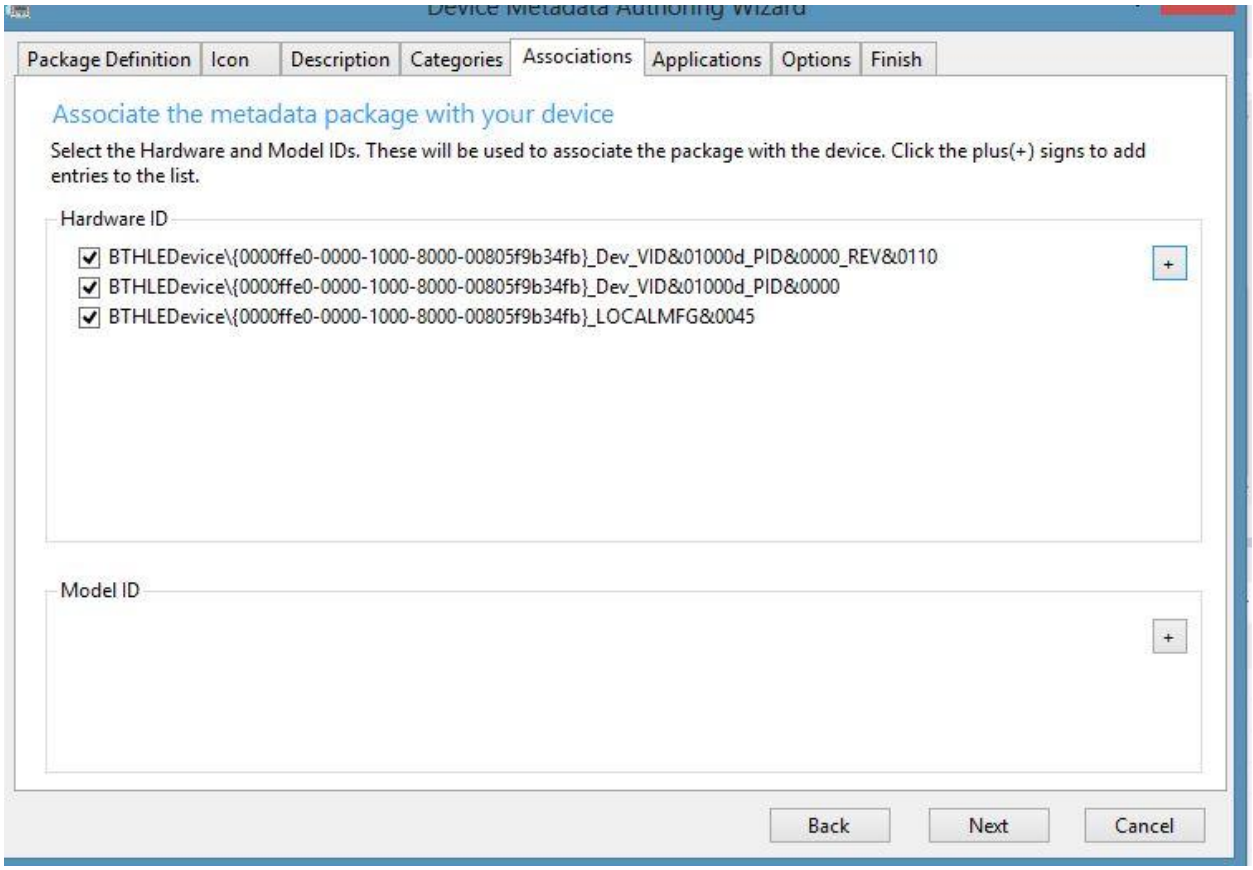


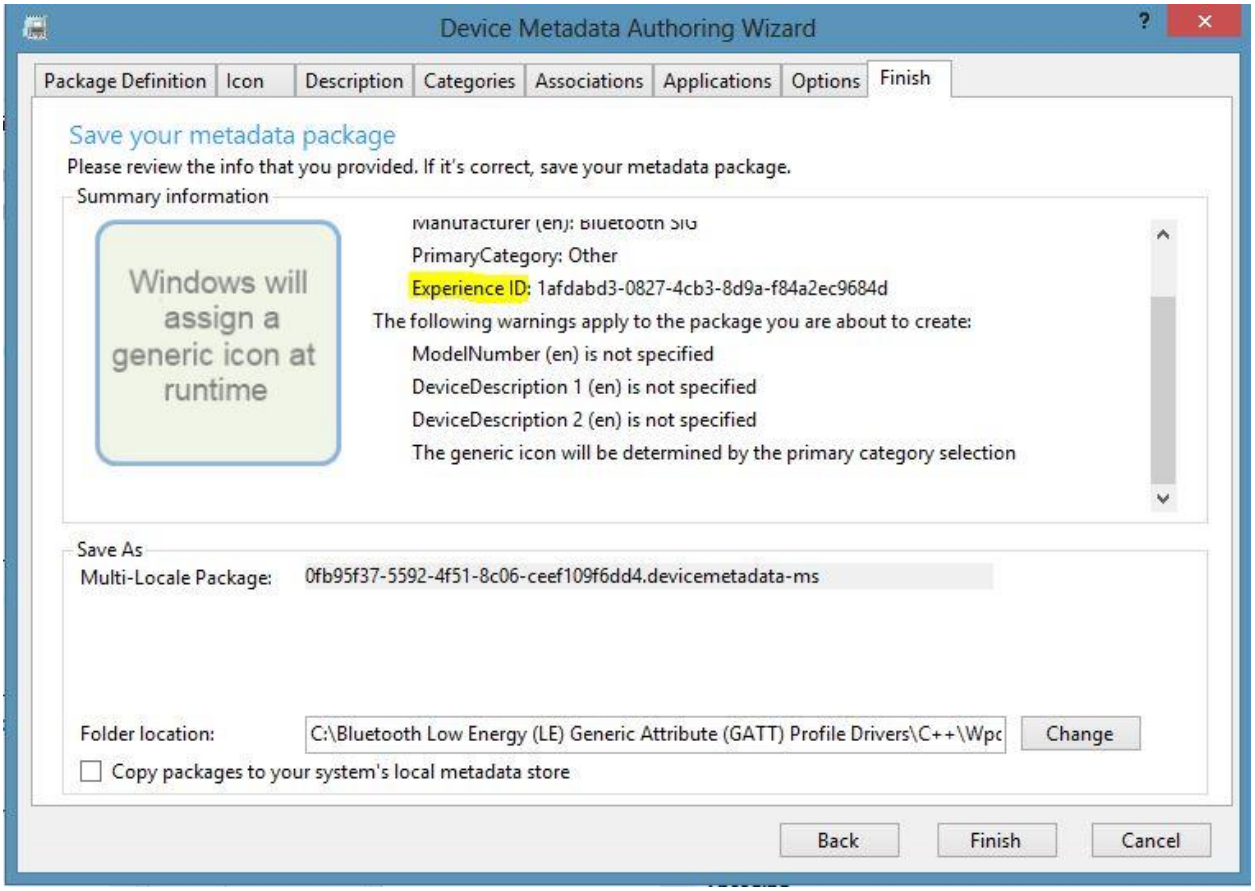
5. You don't need to register any Notification Handlers
6. Since your app will be accessing a custom driver you do need to declare it as privileged, so copy your Package Name and publisher Identity that you received from the store to the Privileged Applications section, by creating a new entry and be sure to check the "Access Custom Driver" option.
7. On the "Finish" page of the Device Metadata Authoring Wizard make a note of your "Experience ID", you'll need it in the next step, be sure to check "Copy Packages to your system's local metadata store" and click "Finish"




Device Metadata Authoring Wizard
Package Definition
Icon
Description
Categories
Associations
Applications
Options
Finish
Specify applications for the device (optional)
Provide information about applications for your device. The Metro style device app will be downloaded when the user installs the device. Privileged applications will have permission to access the device.
Metro style Device App
Package Name:
Publisher:
App ID:
Notification Handlers:
Event ID
Event Asset
Privileged Applications
Package Name
Publisher Name
Access Custom Driver
Back
Next
Cancel


Device Metadata Authoring Wizard
Package Definition
Icon
Description
Categories
Associations
Applications
Options
Finish
Associate the metadata package with your device
Select the Hardware and Model IDs. These will be used to associate the package with the device. Click the plus(+) signs to add entries to the list.
Hardware ID
Model ID
Add a Hardware ID
Enter the Hardware ID to add to the list of available Hardware IDs.
BTHLEDevice\{0000ffe0-0000-1000-8000-00805f9b34fb}_Dev_VID&01000d
Whenever possible, specify a Hardware ID that contains the Vendor ID for your company. For example:
HID\VID_046D&PID_C50E&REV_2510
OK
Cancel
Back
Next
Cancel


Device Metadata Authoring Wizard
Package Definition
Icon
Description
Categories
Associations
Applications
Options
Finish
Associate the metadata package with your device
Select the Hardware and Model IDs. These will be used to associate the package with the device. Click the plus(+) signs to add entries to the list.
Hardware ID
BTHLEDevice\{0000ffe0-0000-1000-8000-00805f9b34fb}_Dev_VID&01000d_PID&0000_REV&0110
BTHLEDevice\{0000ffe0-0000-1000-8000-00805f9b34fb}_Dev_VID&01000d_PID&0000
BTHLEDevice\{0000ffe0-0000-1000-8000-00805f9b34fb}_LOCALMFG&0045
+
Model ID
+
Back
Next
Cancel

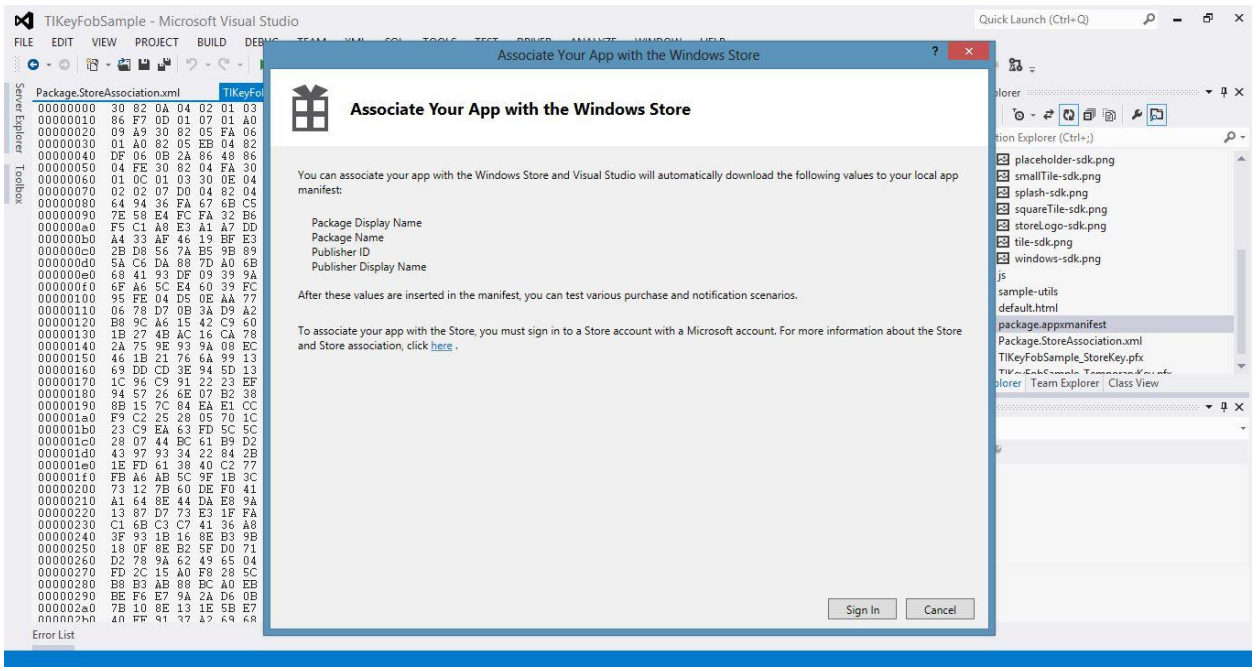


8. To configure the app, make sure that your “package.appxmanifest” file contains the required device capabilities. There is no UI for setting custom device capabilities, so you’ll have to open the package.appxmanifest file with your favorite XML editor (right-click -> “Open With” ->XML (Text) Editor, from Visual Studio), and add the capability for accessing the TI device in the Capabilities section. (You can find an example of this in our “Simple Key service sample” : http://code.msdn.microsoft.com/Bluetooth-Simple-Key-e29ea456)

package.appxmanifest | Package.StoreAssociation.xml | TIKeyFobSample_StoreKey.pfx

```
<?xml version="1.0" encoding="utf-8"?>
<Package xmlns="http://schemas.microsoft.com/appx/2010/manifest">
  <Identity Name="BluetoothSIG.BluetoothSIGkeyfobtest" Version="1.0.0.1"
  <Properties>
    <DisplayName>BluetoothSIGkeyfobtest</DisplayName>
    <PublisherDisplayName>Bluetooth SIG</PublisherDisplayName>
    <Logo>images\storeLogo-sdk.png</Logo>
  </Properties>
  <Prerequisites>
    <OSMinVersion>6.2.1</OSMinVersion>
    <OSMaxVersionTested>6.2.1</OSMaxVersionTested>
  </Prerequisites>
  <Resources>
    <Resource Language="en-US" />
  </Resources>
  <Applications>
    <Application Id="TIKeyFobSample.SimpleBLEPeripheralApp" StartPage="de
      <VisualElements DisplayName="BluetoothSIGkeyfobtest" Logo="images\s
        <DefaultTile ShortName="TIKeyFobJS" ShowName="allLogos" />
        <SplashScreen BackgroundColor="#00b2f0" Image="images\splash-sdk.
      </VisualElements>
    </Application>
  </Applications>
  <Capabilities>
    <!-- Simple Key Fob Device Service -->
    <DeviceCapability Name="0000ffe0-0000-1000-8000-00805f9b34fb" />
  </Capabilities>
</Package>
```

9. Create a StoreManifest.xml file, which must be added to your project, be sure to include the ExperienceId element with the experience Id that you got from the device metadata in step 7. (More information on the Experience ID: http://msdn.microsoft.com/en-us/library/windows/hardware/ff544913%28v=vs.85%29.aspx)


TIKeyFobSample - Microsoft Visual Studio
Associate Your App with the Windows Store
Associate Your App with the Windows Store
You can associate your app with the Windows Store and Visual Studio will automatically download the following values to your local app manifest:
Package Display Name
Package Name
Publisher ID
Publisher Display Name
After these values are inserted in the manifest, you can test various purchase and notification scenarios.
To associate your app with the Store, you must sign in to a Store account with a Microsoft account. For more information about the Store and Store association, click here .
Sign In
Cancel
Package.StoreAssociation.xml
placeholder-sdk.png
smallTile-sdk.png
splash-sdk.png
squareTile-sdk.png
storeLogo-sdk.png
tile-sdk.png
windows-sdk.png
sample-utils
default.html
package.appxmanifest
Package.StoreAssociation.xml
TIKeyFobSample_StoreKey.pfx
Team Explorer
Class View
Error List

Compiling and Building Metro App

Build the sample

1. Start Microsoft Visual Studio Express 2012 for Windows 8 and select **File** > **Open** >**Project/Solution**.
2. Go to the directory in which you unzipped the sample. Go to the directory named for the sample, and double-click the Visual Studio Express 2012 for Windows 8 Solution (.sln) file.
3. Press F7 or use **Build** > **Build Solution** to build the sample.

Run the sample

To debug the app and then run it, press F5 or use **Debug** > **Start Debugging**. To run the app without debugging, press Ctrl+F5 or use **Debug** > **Start Without Debugging**.

References

[1] Bluetooth simple key service sample

[2] Bluetooth Low Energy GATT profile drives

[3] Device Metadata Authoring Tool