Application Authenticity Protection in Native Windows Universal applications

This is a continuation of the Application Authenticity Protection (../) tutorial.

The application-descriptor.xml file

Modify the application-descriptor.xml file on your application by adding a security test configured for application authentication.

Adding the security test

Add the `securityTest` attribute to the Windows / Windows Phone environment element. For example:

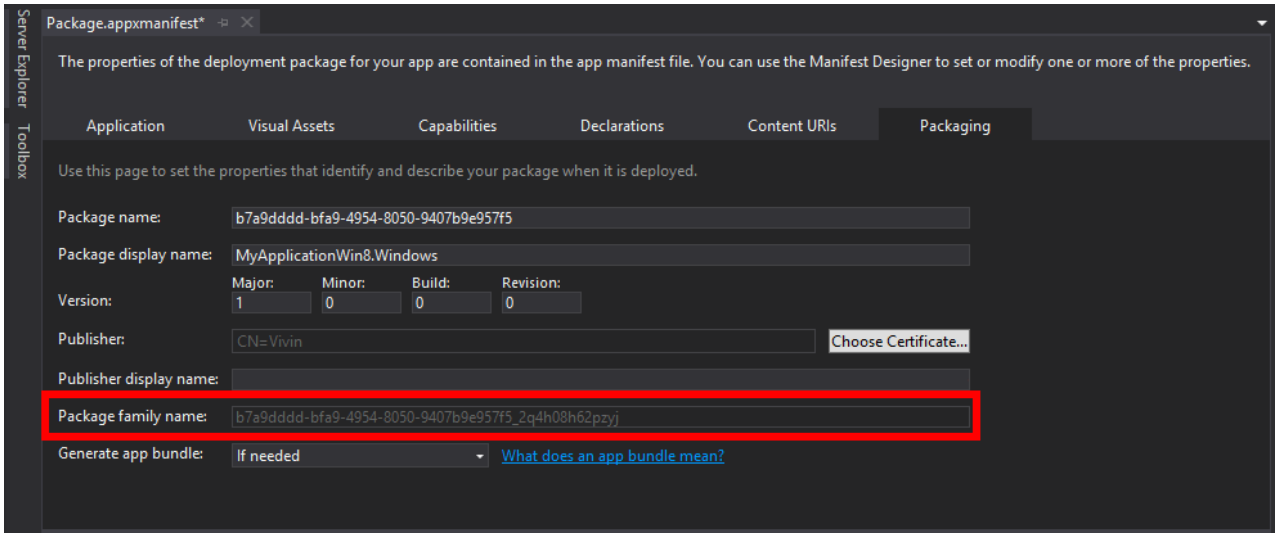
```
<nativeWindows8App id="MyApplication" platformVersion="7.1.0.00.20150703-0630" version="1.0" s
ecurityTest="MyCustomAuthenticityTest">
```

Adding the package family name

Add the `packageName` attribute to the Windows / Windows Phone environment element. For example:

```
<packageName>b7a9dddd-bfa9-4954-8050-9407b9e957f5_2q4h08h62pzyj</packageName>
```

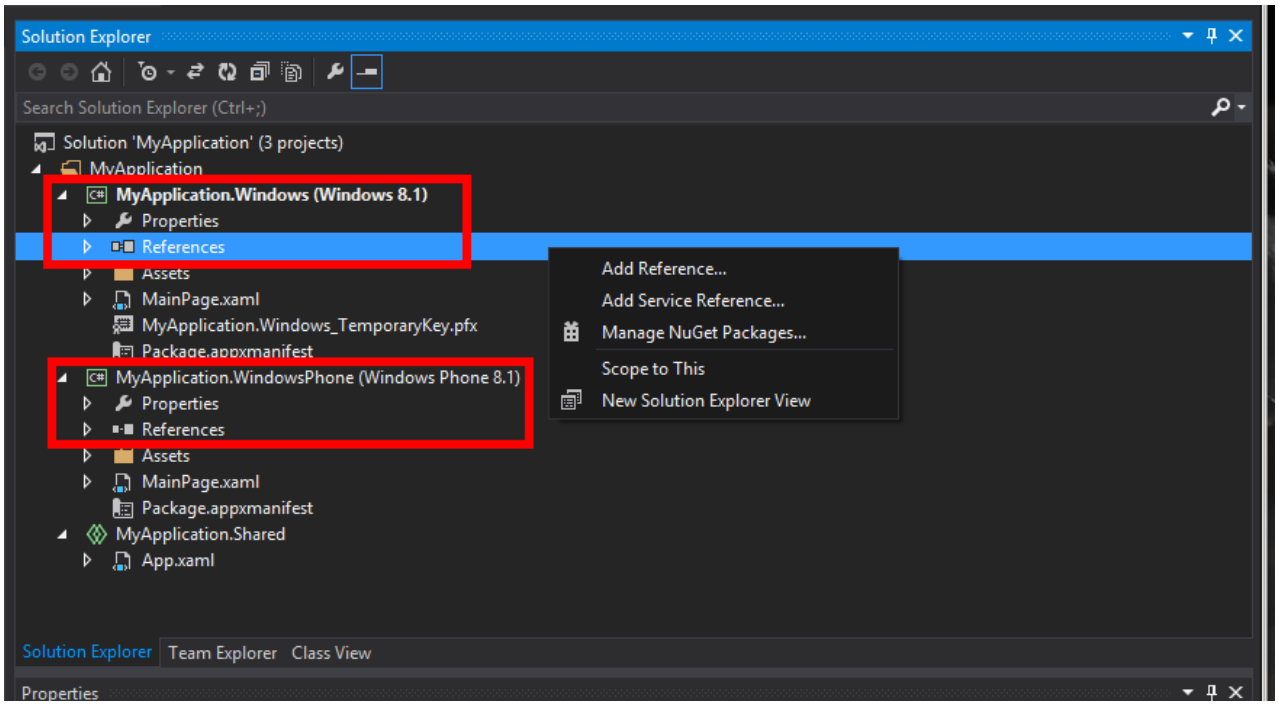
The package name can be extracted from Visual Studio. Open the Project in Visual Studio and search for `Package.appxmanifest` file. Navigate to the Packaging Tab and pick up the package name as shown below.



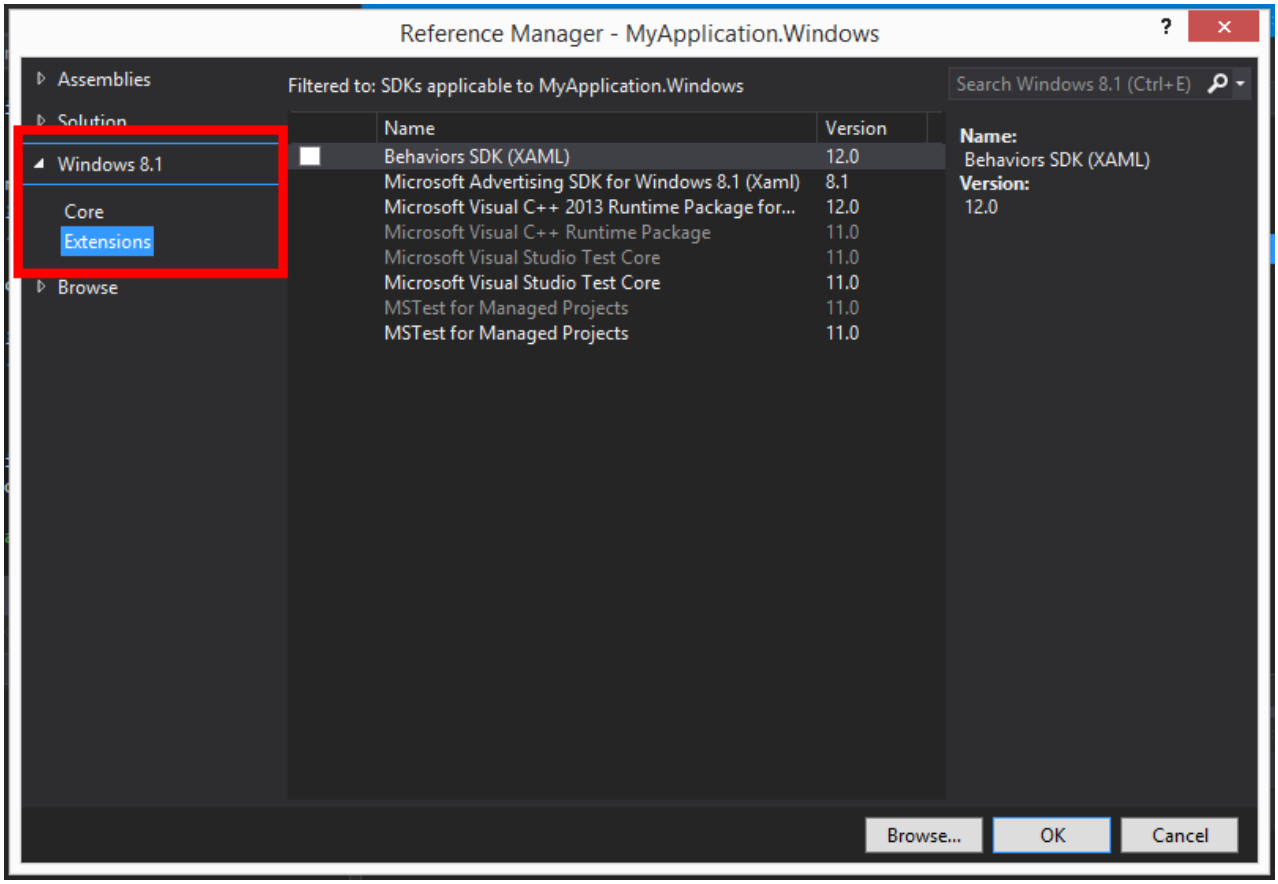
Adding the runtime package to enable application authenticity

1. Open your Native Mobile First Project in Visual Studio. Choose the environment, either Windows or

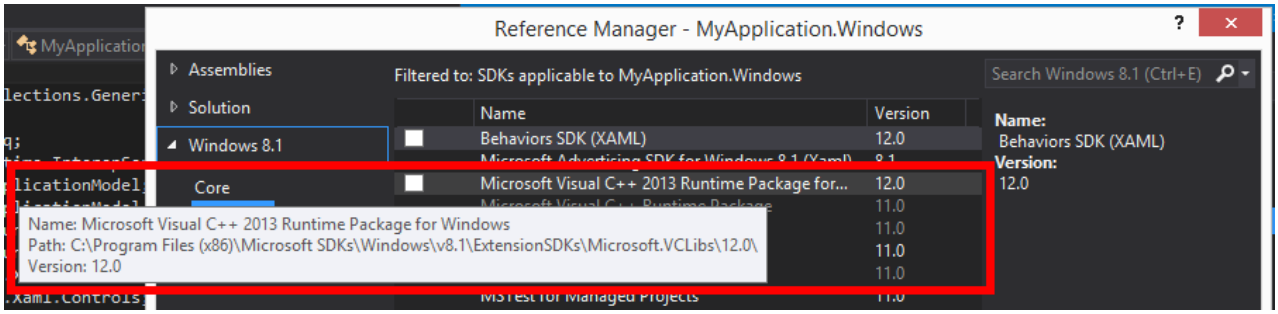
WindowsPhone in your solution. Choose References in the environment and select to Add Reference.



2. In the Reference Manager Window, select Windows 8.1 -> Extensions, as shown. This will list out the SDKs applicable to your project.



3. Add the `Microsoft Visual C++ 2013 Runtime Package for Windows` or the `Microsoft Visual C++ 2013 Runtime Package for Windows Phone` based on which environment your are adding it to.



4. Now build your project to generate the final executable (.appx) and pass it to the `wladm` command to enable extended application authenticity as explained earlier