GetVersion API and Windows 8.1

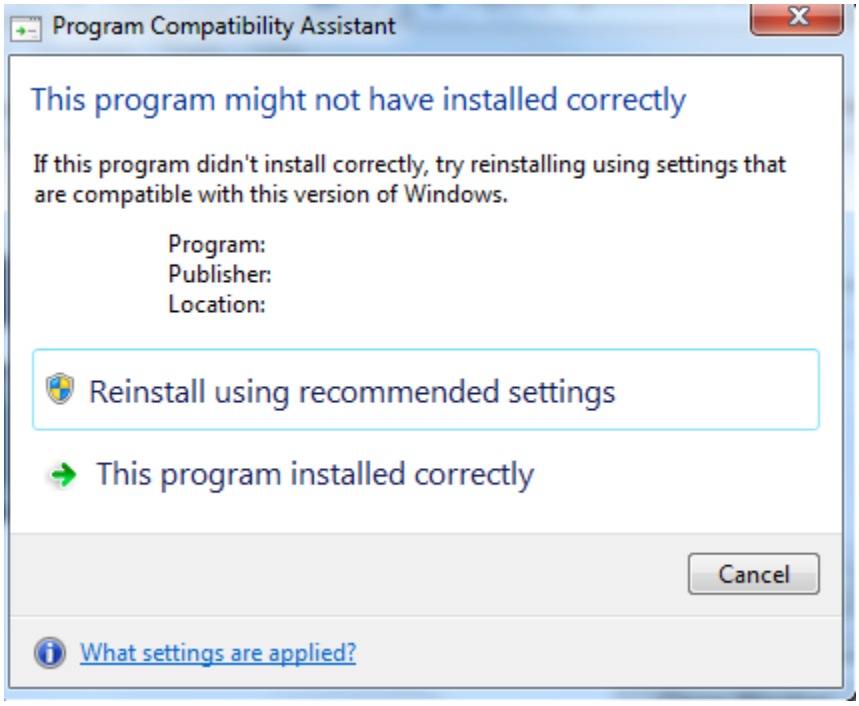
This document addresses two problems that can be resolved by updating the Application Manifest file for programs that are distributed to post-Vista Windows platforms:

1. The deprecated Windows version numbers returned by GetVersion API calls
2. The annoying Program Compatibility Assistant's "this program might not have installed correctly" warning messages

For the first problem- when GetVersion is called by an application on Windows 8.1, which has only provided a manifest addressing up to Windows 8, the GetVersion API will return the internal value for Windows 8 (6.2 for the major version**.**minor version), rather than that of Windows 8.1 (6.3). This practice of returning the lowest value, will continue into Windows 10.

Regarding the second problem- beginning with Windows Vista, Microsoft introduced the famous Program Compatibility Assistant warnings. *The best list of all situations addressed by the PCA can be found at Microsoft - https://msdn.microsoft.com/en-us/library/bb756937.aspx*

The following warning appears when a installer or application ends, and Windows did not receive parameters indicating from the application being run, that it was compatible with your version of Windows:



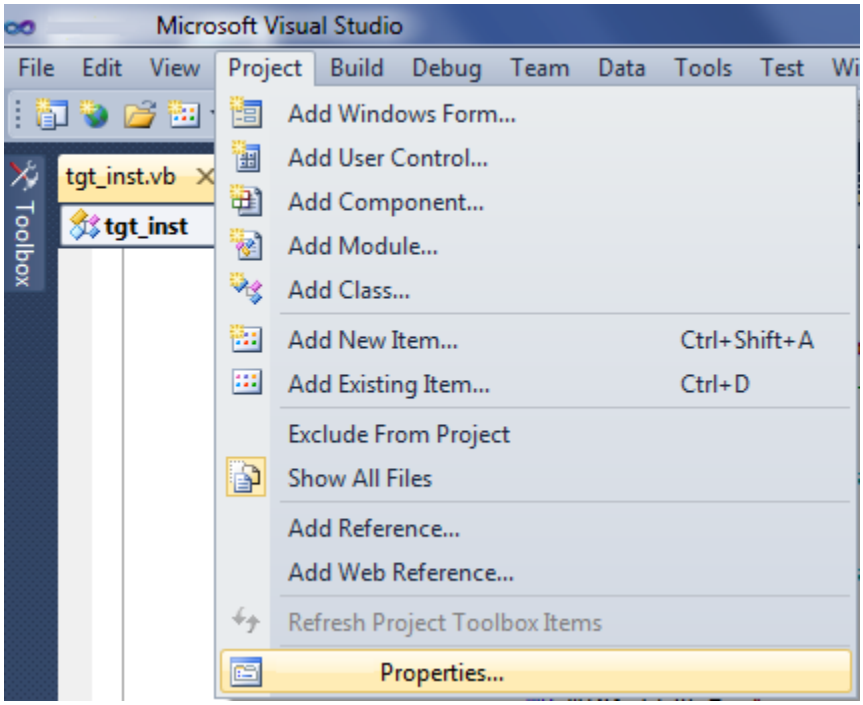
In order to assure Windows that an application is compatible with a specific version of the post-Vista operating system, Visual Studio 2008 began offering the compatibility section for the app.manifest file of a My Projects folder. The most recent supportedOS Id value provided by Microsoft, is in bolded blue below:

```
<compatibility xmlns="urn:schemas-microsoft-com:compatibility.v1">
    <application>
        <!-- Windows 8.1 -->
        <supportedOS Id="{1f676c76-80e1-4239-95bb-83d0f6d0da78}"/>
        <!-- Windows Vista -->
        <supportedOS Id="{e2011457-1546-43c5-a5fe-008deee3d3f0}"/>
        <!-- Windows 7 -->
        <supportedOS Id="{35138b9a-5d96-4fbd-8e2d-a2440225f93a}"/>
        <!-- Windows 8 -->
        <supportedOS Id="{4a2f28e3-53b9-4441-ba9c-d69d4a4a6e38}"/>
    </application>
</compatibility>
```

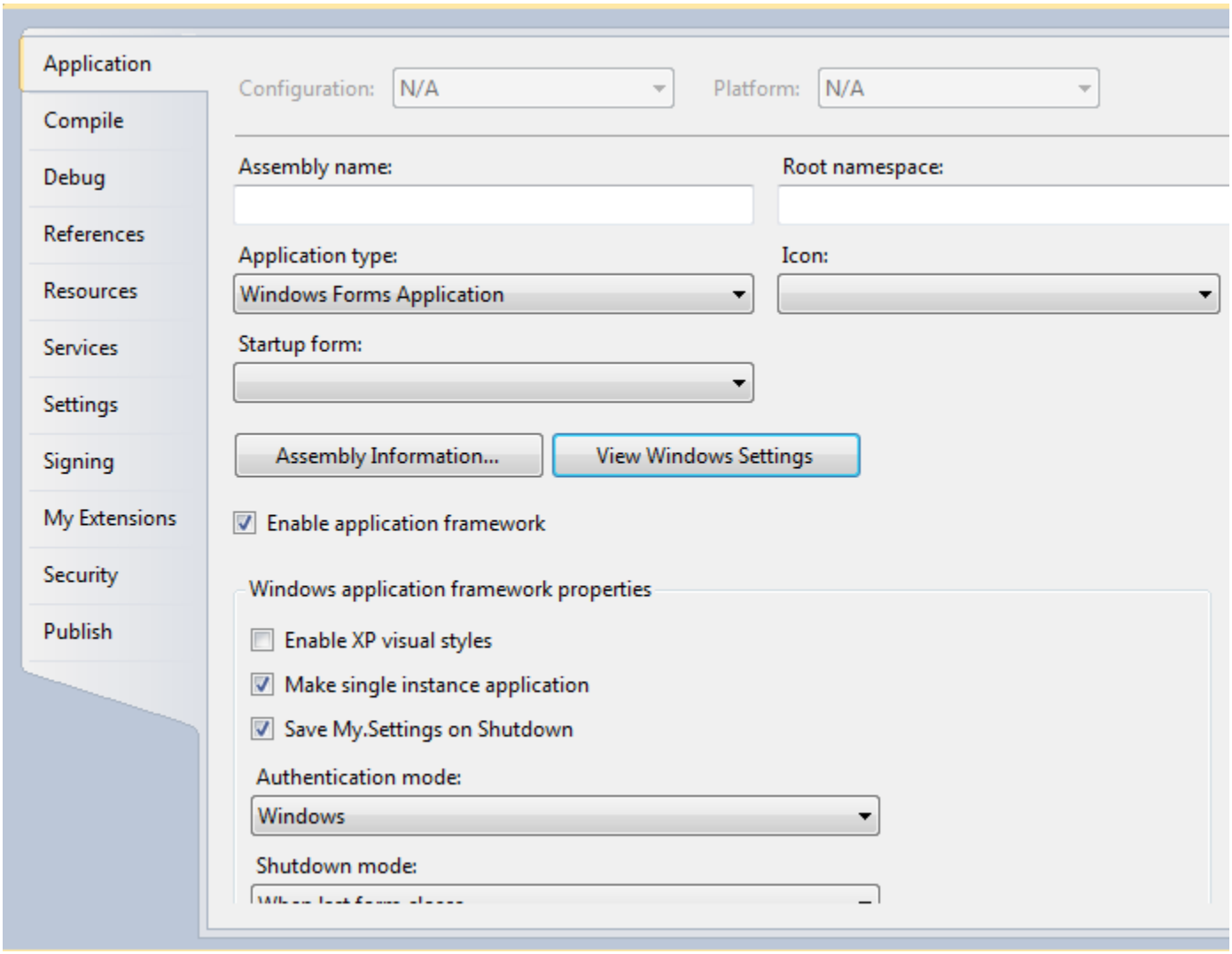
A simple example app.manifest file containing these values can be downloaded from http://www.david-e-young.com/wordpress/downloads/app.manifest

To edit the application manifest directly from Visual Studio 2010 and subsequent versions:

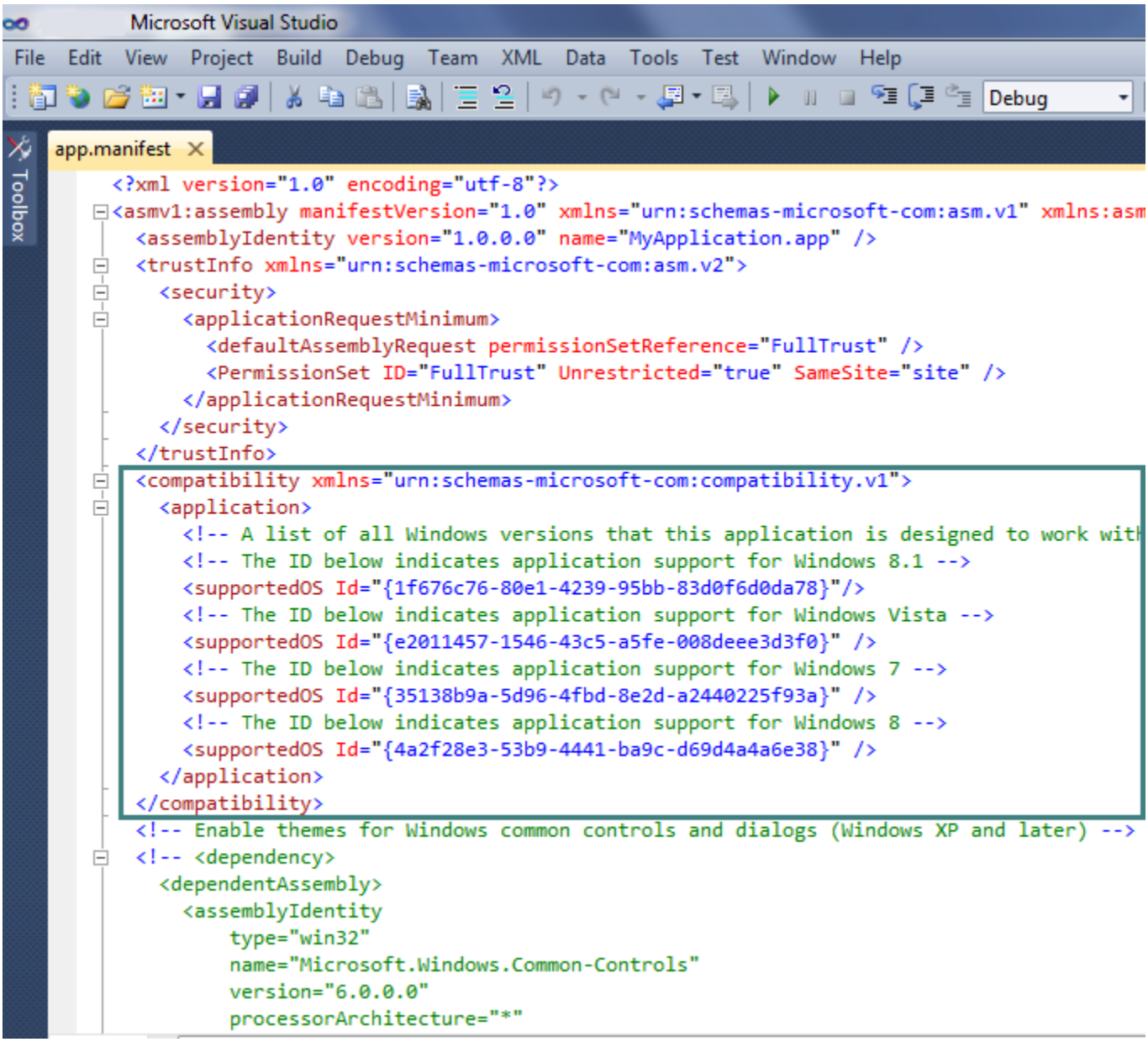
1. Click on the Project Properties item of the Properties dropdown menu.



2. When the Properties window appears, select the ‘Application’ tab on the left side. Click on the ‘View Windows Settings’ button, to open the app.manifest file for the current project.



3. Edit the app.manifest file as needed, to provide the compatibility section for all versions of Windows that your application will support. The one shown below will support Windows Vista, Windows 7, Windows 8, and Windows 8.1.



Helpful Links

For Windows 8.1-related Application Manifest Recommendations:
https://msdn.microsoft.com/en-us/library/windows/desktop/dn302074%28v=vs.85%29.aspx

For the structure of the entire app.manifest file:
https://msdn.microsoft.com/en-us/library/aa375635%28v=vs.85%29.aspx

For an explanation of significant parts of the app.manifest file:
https://msdn.microsoft.com/en-us/library/windows/desktop/hh848036%28v=vs.85%29.aspx

For a good explanation of the registry keys that are used by the Program Compatibility Assistant:
http://journeyintoir.blogspot.com/2013/12/revealing-program-compatibility.html

A simplistic FAQ regarding Program Compatibility Assistant:
http://windows.microsoft.com/en-us/windows/program-compatibility-assistant-faq#1TC=windows-7