Release notes: Visual Studio Tools for Windows 10

Windows 10 Technical Preview | March 2015 (Updated April 10, 2015)

Contents

[About this document 1](#_Toc416433794)

[Diagnostic tools 1](#_Toc416433795)

[Debugging 2](#_Toc416433796)

[.NET Native 3](#_Toc416433797)

[C++ 4](#_Toc416433798)

[Build 4](#_Toc416433799)

[Maps 4](#_Toc416433800)

[Speech recognition 5](#_Toc416433801)

[Other 5](#_Toc416433802)

[Copyright information 5](#_Toc416433803)

# About this document

These release notes describe significant known issues with the SDK and Visual Studio development tools released with the Windows Technical Preview. Although this preview gives you early access to the next version of the development tools and the opportunity to provide feedback that could influence future releases, its features aren’t yet complete and so it does include the following issues.

# Diagnostic tools

The following issues exist when you run the diagnostic tools without the debugger for Windows universal app projects (**Debug** > **Start Diagnostic Tools without Debugging**):

* When the startup project is a C++ universal app set to deploy to a physical Windows Phone device, the profiling tools fail to launch.

**Workaround:** Using the **Start debugging** drop-down menu, change the target to Local Machine, Simulator, or Emulator.

* When running the CPU Usage or Memory Usage tool on a debug build of C++ universal apps, function names may not be displayed and clicking the **Create Detailed Report** link in the CPU Usage tool will cause Visual Studio to crash.

**Workaround**: Switch the build type of the current startup project to "Release" before running the tools, or do this:

1. Right-click the project.
2. Select Properties.
3. Open Linker.
4. Select Debugging.
5. Set Generate Debug Info to "Optimize for Debugging (/DEBUG)".

* When the startup project is a universal app with the Memory Usage Tool selected and it is set to deploy to a physical Windows Phone device, the profiling tools fail to launch.

**Workaround:** Use the Emulator, Local machine, or Simulator as targets.

# Debugging

* Deployment of a Windows app to the Windows 10 for Mobile emulator fails in Visual Studio if you have the Visual Studio Emulator for Android installed.

**Workaround:** Avoid having Visual Studio Emulator for Android and Windows 10 for Mobile Emulator installed on your machine at the same time.

* Visual Studio hangs when you start debugging a solution that contains a Windows Store app and an ASP.NET app.

**Workaround:** Disable diagnostic tools while debugging. To do this, click **Debug**, click **Options**, open **Debugging**, select **Diagnostic Tools**, and set **Enable while debugging** to false.

* Your Windows 10 app does not appear in list of available apps when you’re using **Debug Installed App package**.

**Workaround:** Launch the project from Visual Studio, or manually launch the app and use **Debug** **>** **Attach to Process**.

* Cannot debug scripts in Windows Store apps using JavaScript.

**Workaround:**

1. Run a command prompt as an administrator.
2. Copy:  
    C:\Program Files\Common Files\Microsoft Shared\VS7Debug\pdm.dll  
   to:  
    C:\Windows\System32
3. On a 64-bit OS, also copy:  
    C:\Program Files (x86)\Common Files\Microsoft Shared\VS7DEBUG\pdm.dll  
   to:  
    C:\Windows\syswow64

* Evaluating a C# variable results in Error CS1704:

"An assembly with the same simple name 'Windows' has already been imported. Try removing one of the references (e.g. 'Windows.winmd') or sign them to enable side-by-side."

**Workaround:**

1. On the **Tools** menu, select **Options** and then **Debugging**.
2. Select the **Use the legacy C# and VB expression evaluators** check box. This setting will take effect after you restart the debug session.

* While debugging a 32-bit managed app, the app randomly exits or execution does not continue as expected after stopping in the debugger (that is, after stepping or hitting a breakpoint), or Visual Studio crashes or hangs.

**Workaround:** Launch the app as x64.

# .NET Native

.NET Native is used by Windows universal apps that include .NET code. By default, new Windows projects containing .NET code will use .NET Native for "Release" builds. In this release of Visual Studio, .NET Native is used only for C# projects. .NET Native is not yet supported for Visual Basic, JavaScript, or C++ projects that contain .NET code.

.NET Native applications created with a “Release” build cannot be debugged on Windows Phone 10. An error message ("Unable to debug .NET native compilation code...") appears if you attempt to do so. Debugging a non-.NET Native application, which is produced when you perform a "Debug" build, will continue to work as expected.

# C++

* When C# or JavaScript apps use a C++ component, the C++ runtime framework package (VCLibs.14) is not available.

**Workaround:** Add a reference to the C++ 14 AppLocal Runtime Package instead.

* Rebuild fails for a C++ XAML app after you edit a file. You receive Error MSB6003: “The specified task executable ’CL.exe‘ could not be run. The process cannot access the file 'FILE' because it is being used by another process.”

**Workaround:** Terminate the process ScriptedSandbox.exe.

# Build

* After you install the Windows SDK, Windows 10 apps do not build successfully with [Visual Studio 2015 Community Technology Preview (CTP) 6](https://www.visualstudio.com/en-us/news/vs2015-vs.aspx) because of a metadata issue.

**Workaround:** Although you should not install the Windows SDK if you are doing app development, you can do this to enable Windows 10 apps to build successfully:

1. Exit Visual Studio.
2. Open a command prompt as administrator.
3. Run this script: \Program Files (x86)\Windows Kits\10\bin\x86\GenerateUnionWinMD.cmd

* MIDL fails to compile IDL files in Windows universal app projects, and gives this error:

“midlrt : error MIDL4034: Failed to load a dependency file. Windows.winmd (HRESULT:0x80070002 - The system cannot find the file specified.)”

**Workaround:** Set the **Additional Metadata Directories** property of the IDL file (in Visual Studio) to "C:\Program Files (x86)\Windows Kits\10\References\  
Windows.Foundation.FoundationContract\1.0.0.0"

# Maps

* When you include the Windows 10 **MapControl** (Windows.UI.Xaml.Controls.Maps) in your XAML-based app, it does not render maps on Windows 10 *phone* emulators. A blank, blue map appears without any map tiles. The map renders fine in on a physical phone device.

**Workaround:** Use a physical phone device to test the map, or continue to use it in a desktop environment. This issue will be fixed in the next public release of the SDK.

# Speech recognition

* The **SpeechRecognizer.RecognizeWithUIAsync** method displays an error dialog, stating that the specified module could not be found.

**Workaround:** Don’t invoke **SpeechRecognizer.RecognizeWithUIAsync** in this release.

* The **SpeechRecognizer** class fails to initialize in the Windows Phone emulator. The constructor throws an exception.

**Workaround:** Deploy to a physical device instead of to the emulator.

# Other

* Visual Studio Simulator does not refresh correctly.

**Workaround:** Rotate the simulator to force a refresh.

* Windows 8.1 extension SDKs do not work for Windows universal app projects.

**Workaround:**

1. Copy the extension SDKs from the Windows 8.1 Extension SDK location to the universal apps location—that is:

From:  
 C:\Program Files (x86)\Microsoft SDKs\**Windows\v8.1\**ExtensionSDKs  
to:   
 C:\Program Files (x86)\Microsoft SDKs\**UAP\v0.8.0.0\**ExtensionSDKs

1. Reference the third-party extension SDK in your project; for example:

<ItemGroup>

<!-- A reference to the entire .NET Framework and

Windows SDK is automatically included -->

<SDKReference Include="BehaviorsXamlSDKManaged, Version=12.0">

<Name>Behaviors SDK %28XAML%29</Name>

</SDKReference>

</ItemGroup>

# Copyright information

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes. Some information relates to pre-released product, which may be substantially modified before it's commercially released. Microsoft makes no warranties, express or implied.

© 2015 Microsoft. All rights reserved.

Please refer to [Microsoft Trademarks](http://www.microsoft.com/en-us/legal/intellectualproperty/Trademarks/EN-US.aspx) for a list of trademarked products.

All other trademarks are property of their respective owners.