

Windows 10 IoT – The Big Reboot

By Sean D. Liming and John R. Malin
Annabooks – www.annabooks.com

July 28, 2015

For those developing with Windows Embedded, there has been some frustration on the lack of future direction. Slowly over the past year, Microsoft has been making changes. With the release of Windows 10, there is finally a roadmap, but it wasn't easy getting here. In this paper, we will introduce the new direction.

Two Big Changes make all the Difference

For 7 years, 2001-2008, there was Windows XP and all the service packs. For 7 years, 2006-2013, there has been Windows Vista, Windows 7, Windows 8, and Windows 8.1. The flurry of new versions was the result of a couple of misfires, market inflections, processor advancements, and new application programming paradigms. Some of these releases were fantastic, but others fell really flat. Many in the media expressed concerns on Microsoft's industry relevance. Those in the IoT/Embedded space were equally concerned about the lack of updates or a roadmap. Add the fact that Linux domination in the embedded, now IoT, space is well known, the ideas of a royalty free OS and free professional development tools were appealing to many companies. Windows 8.x required activation, which for embedded systems was not welcome. Lack of support for Windows CE (a.k.a. Embedded Compact) made customers look to other operating systems for future products. Having supported Windows in the IoT/Embedded market for almost 20 years, even we had some serious doubts on the future direction. Now, Windows 10 comes along with a new company-wide approach that will provide better focus on the future.

A few years ago, change started coming to Microsoft. Companywide reorganizations, which included the Windows Embedded team moving into the Windows and Devices Group, the slow dropping of some projects and products, and a new CEO have resulted in new thinking about tools and operating systems going forward.

Zero-Cost Tools

The first change is with the tools. Visual Studio 2015 has a new community edition that is available for free download. It is not just a bare bones Visual Studio version, but a professional grade tool that is not limited to Windows development. Developers can now target iOS, Android, Linux, web application and cloud services. No longer limited to one OS, Visual Studio developers can now target the billions of devices using one cross-platform development suite. Here is a breakdown of the free tools:

- Visual Studio 2015 Community – Free. Pro and Enterprise versions are available for purchase.
- SDK (Software Development Kit) – Free download and included with Visual Studio.
- ADK (Assessment and Development Kit) – Contains SIM (Windows System Image Manager) and ICD (Windows Imaging and Configuration Designer) tools which will be discussed in a little later. Free download from Microsoft.
- WDK (Windows Driver Kit) – Used to develop device drivers. Free download from Microsoft.
- GitHub contains application and device driver samples. All Free.

Accounting for tool costs and the availability of Windows developers, the new Windows 10 and Visual Studio 2015 make a compelling case for your next IoT/Embedded system project.

One Operating System for All Devices

The second change is the operating system offerings. Rather than having two base operating systems for mobile (Windows CE) and desktop (Windows), there is now one operating system, Windows 10, that runs on a wide range of devices. From the very low end (IoT/Embedded) to high end devices (HoloLens), all of the devices are built on the Universal Windows Platform (UWP). Visual Studio can be used to write one universal application that runs on all devices, and address different screen sizes and orientations.



Universal Windows Platform

If small to large devices sounds familiar, those in the Linux community have had this solution for many years. It took some technology advancement and experimentation, but Microsoft is now positioned to reach an equivalent range of devices.

Windows 10 and Visual Studio 2015 create a break from the past to push forward a strategy that allows developers to create universal applications and allows OEMs to focus on one operating system for a variety of devices. The big reset button has been pushed at Microsoft.

IoT/Embedded has not been forgotten

As for IoT/Embedded developers, the Universal Windows Platform is a welcome sign. Where the former Microsoft CEO couldn't even spell embedded or acknowledge embedded products were available, the new CEO includes IoT/Embedded in the same breath as Xbox and Surface. With IoT under the group that produces Windows, the concerns of Microsoft completely abandoning the IoT/Embedded space begins to ease a bit. The IoT operating system offerings come from the same code base as kernel Windows desktop, and here are the Windows 10 IoT solutions

- Windows 10 IoT Enterprise - Windows 10 desktop that includes the lockdown features such as Unified Write Filter, Shell Launcher, Gesture Filter, etc.
- Windows 10 IoT Mobile - Windows 10 Mobile for enterprise devices that need flexibility in chassis implementation, long term servicing lifecycle, and enables the enterprise to manage platform updates.
- Windows 10 IoT Core - An OS built for ARM, x86 and x64 that enables building low cost, small footprint devices that support Universal and console applications only.

Windows 10 IoT Version	Processor	Universal Applications	Tradition PC (Form) and Console Applications
Windows 10 IoT Enterprise	x86, x64	Yes	Yes
Windows 10 IoT Mobile	ARM, x86-SOC	Yes	Console Only
Windows 10 IoT Core	ARM, x86, x64	Yes	Console Only

Those using x86 and x64 processors can write one application and one driver that fits all devices. OEMs can take advantage of device drivers already created for Windows to quickly build a system.

What is missing from the list is Windows CE. When Windows desktop was ported to ARM, Windows Phone 8 and Windows Mobile quickly adopted Windows on ARM, and Windows CE no longer fit in the overall strategy. Considering industry support was waning, economically, it didn't make sense to keep developing Windows CE tools and new kernel releases. The promise of a ROM-able 32-bit Windows OS introduced in 1996, Windows CE had some early success, but over time, it was not as successful as Windows Desktop embedded versions. The last Windows CE, Windows Embedded Compact 2013, release will still be support for some time, but there will not be any new releases. It is advised that new projects should not use Windows CE.

With Windows CE gone and various Windows Embedded release currently available, here is an unofficial road map for when you want to migrate from your current OS to a Windows 10 solution:

Previous Windows Embedded OS	Windows 10	Description
Windows 7, 8, or 8.1 for Embedded System	Windows 10 IoT Enterprise	Windows 10 desktop that includes the lockdown features such as Unified Write Filter, Shell Launcher, etc. Windows 10 IoT Enterprise is a special license version of Windows 10 Enterprise
Windows Embedded Standard 7, POSReady 7, Windows Embedded Standard 8, Windows Embedded 8.1 Industry	Windows 10 IoT Enterprise	
Windows 8.1 Phone	Windows 10 Mobile	An OS that is designed for mobility and phones.
Windows 8.1 Mobile	Windows 10 IoT Mobile	Windows 10 Mobile for enterprise devices that need flexibility in chassis implementation, long term servicing lifecycle and enables the enterprise to manage platform updates
Windows CE (Embedded Compact) there are two possibilities	Windows 10 IoT Mobile	Same as above, but must meet the requirements of a mobile device
	Windows 10 IoT Core	An OS built for ARM, x86 and x64 that enables building low cost, small footprint devices that support Universal Apps

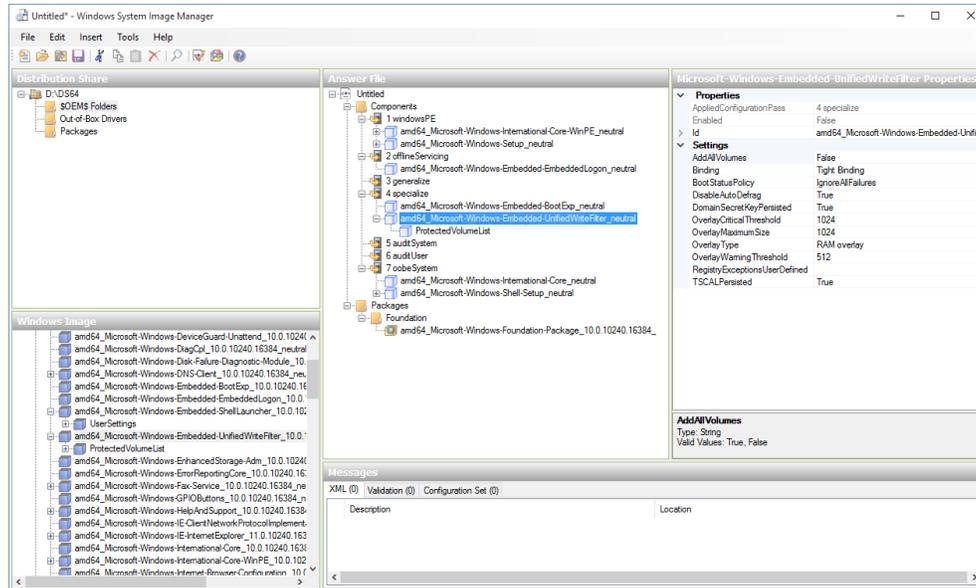
The above table may change as more information becomes available. Most IoT/Embedded OEMs are going to choose Windows 10 IoT Enterprise or Windows 10 IoT Core

Continue the long trend with Windows 10 IoT Enterprise (IoT-Enterprise)

If you have been on the Windows NT Embedded, Windows XP Embedded, Windows Embedded Standard 7, POSReady, and Window Embedded 8.x track, the Windows 10 IoT Enterprise (IoT-Enterprise) is your Windows 10 follow-on release. IoT-Enterprise is the same Windows 10 desktop operating system with a special license for IoT OEMs. IoT-Enterprise includes the lockdown features that have evolved from previous Windows Embedded releases, such as Unified Write Filter and Shell Launcher. IoT-Enterprise also has long term support, which is needed for IoT/Embedded systems.

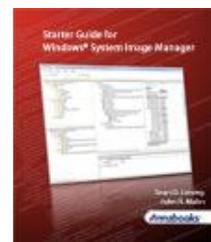
The IoT-Enterprise development process is more like Windows Embedded 8.1 Industry.
Copyright © 2015 Annabooks, LLC. All rights reserved

As far as development goes, Microsoft was able to save money by not componentizing or creating custom build tools for an embedded release, thus no WES10 (Windows Embedded Standard 10). Rather than Target Designer or component Designer, System Image Manager (SIM) is used to create custom installations of the operating system. By using the same deployment tools as Windows Server and Desktop, IoT-Enterprise can be built, tested, and released at the same time as Windows Desktop.



The departure from componentization might be an issue for some. Componentizing the operating system was originally performed so you can pick and choose features of the OS to put into the image and reduce the foot print by not including unwanted features. Componentization was a great idea and worked for the most part, but there were some subtle differences between desktop and embedded releases that caused some issues. For example, installing device drivers on Windows XP Embedded resulted in a hunt and find search for internal Windows support drivers that were already there. Also, forum posters would always ask about module or component that contains a certain file or feature. Recently, some forum posters have noticed that WES 7 has some performance differences from Windows 7 desktop that center on power management. With storage sizes always getting bigger and cheap, the need to componentize the OS for image size reduction is not as big of a factor as it was. Hopefully, we will not see any subtle issues between IoT-Enterprise and Windows 10 desktop.

If you are familiar with Windows Embedded 8.1 Industry, nothing has changed as far as developing an answer file using SIM to select features, present items, and include device drivers and applications. Best of all, we already have a book that covers SIM: [Starter Guide for Windows® System Image Manager](#). Although it calls out Windows Embedded 8.1 Industry, the same process and steps can be used for IoT-Enterprise. If you have been using WES 7, ICE was a derivative of SIM, so many things will look familiar. You will be able to carry the distribution share forward, but updating drivers to the latest available should be considered. For those still using Windows XP Embedded, you will have to start over, but you will be pleasantly surprised at how easier it is to develop a custom image.



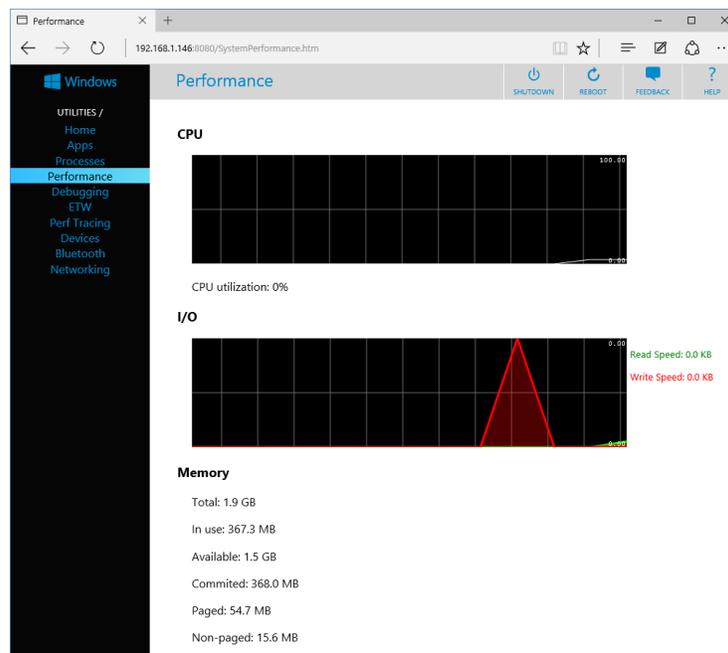
The output from SIM is a configuration set that you manually merge with the IoT-Enterprise installation DVD to create a custom installer. The custom installer can be placed on a USB flash disk for deployment on target systems, and the deployment process is the same as Windows Desktop.

There is a new tool in the ADK called Windows Image Configuration Designer (ICD), which can also create custom OS installers. The tool has some nice provisioning features like presetting security settings. ICD supports all versions of Windows. We will cover the ICD usage in a future paper.

Going small with Windows 10 IoT Core

For those using Windows CE, you have two choices: Windows 10 IoT Mobile or Windows 10 IoT Core. There is little information on Windows 10 IoT Mobile since Microsoft is focused on a few OEMs to produce devices. Hopefully, in the future more access will be granted.

This leaves the smallest UWP that has gained the biggest interest: Windows 10 IoT Core. The Core version is a cut down version of Windows that supports universal and console applications. Traditional Windows Form applications are not supported as the old graphics interface support has been removed. Applications like Control Panel, Notepad, GPedit, and MS Paint will not run. PowerShell and various command line utilities make up for the lack of a Control Panel. There is also a web management tool to remotely maintain and service a device. The removal of the old graphics support reduces the OS footprint to about 1GB.

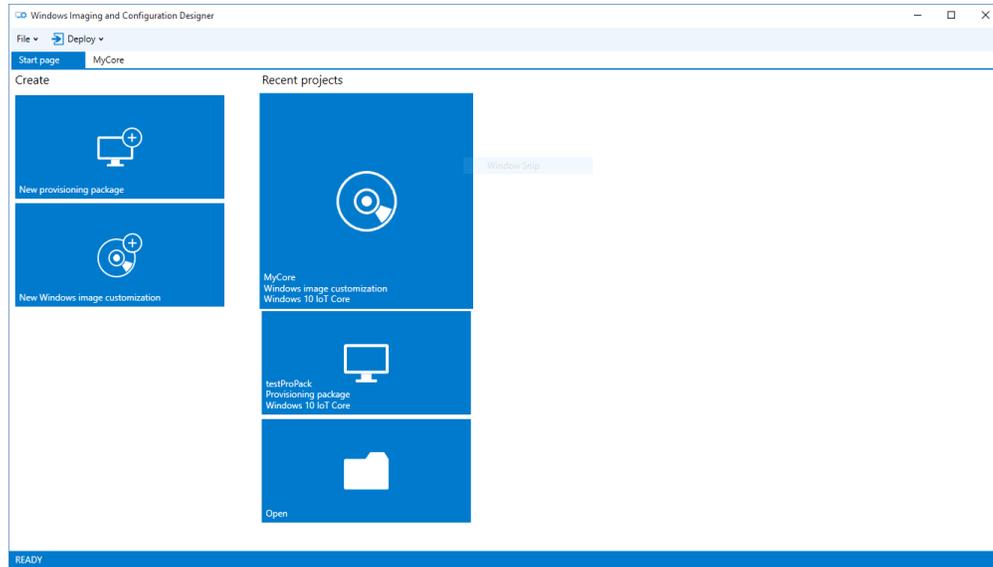


The most important feature of Windows 10 IoT Core is the new API's and drivers for micro controller I/O such as GPIO, SPI, PWM, and I²C. ARM SOCs have had these I/O since the beginning, but Intel has recently added these I/O through a Low Power Sub-System (LPSS). The latest Intel Atom processors include LPSS support. Maker and IoT developers can choose between either architecture.

Microsoft started the effort to develop support for the micro controller I/O using Windows 8.1 on the Galileo platform. A new website was created to initiate the beta program: Windowsondevices.com. The Intel Galileo platform served as an incubator to work out the details and get interest in Windows on a Maker device. With Windows 10 IoT Core, support for MinnowBoard Max (x86) and Raspberry Pi2 (ARM) has been added.

Applications are written in Visual Studio 2015 and remotely deployed and debug over Ethernet. There are several project examples on the Windowsondevices.com website that demonstrate SPI, I2C, Node.JS, and GPIO.

When your application is finally complete, you can integrate it into a custom Windows 10 IoT Core image using the new ICD tool. For Windows 10 IoT Core, ICD can be used to preset some information and integrate applications. ICD is then used to build an FFU file that can then be flashed to a microSD or USB flash disk. At this time, information on how to create BSPs and other ICD details are still being developed. More details about Windows 10 IoT Core and ICD will be coming over the next few months.



The table below shows a comparison between Windows CE and Windows 10 IoT Core.

	Windows Embedded Compact 2013	Windows 10 IoT Core
Processor Support	ARM, x86, SH4, MIPS	ARM, x86, x64
Footprint	4GB to 20 GB depending on features included	OS size is 1GB, fully installed on 8GB microSD
RAM	256 MB Minimum	256MB Minimum, more is recommended
Application development tools	Visual Studio 2012 and 2013	Visual Studio 2015
.NET Support	.NET Compact Framework	.NET Framework 4.5+
Programming Languages	C#, VB.NET, C/C++ MFC, XAML	C#, VB, Node.js (JavaScript), C++/CX
Application Types	Windows Forms, Silverlight, and Console Applications	Universal Applications and Native Win32 as Console Applications
Shell	Supports the creation of a custom GUI shell, and can run multiple applications	Only one application can run as the shell, multiple background tasks are supported
Device Driver Support	Custom drivers for the Windows CE/EC kernel	Windows 10 Universal Device Drivers
SPI, I²C, GPIO	Yes	Yes
Real-Time Support	Kernel is designed for Real-Time	Not Supported
BSP Support	Multiple BSPs available	BSP information coming soon
License	Per Unit Royalty	Free

A Positive Reboot with a Cost Effective Approach for Developers and OEMs

We already hear the jokes: “*The common way to fix most problems with Windows is to turn-it-off-and-turn-it-on-again*”. In this particular case, a full company reboot has created a longer term strategy that embraces IoT/Embedded development, which is a far cry from years past. We finally have a roadmap.

For those of you who have been on the NT Embedded to Windows Embedded 8.1 Industry track, nothing has really changed. The same popular desktop OS brought to the IoT/Embedded space with special feature for ruggedized applications continues to march on with Windows 10 IoT Enterprise.

For those looking for a smaller Windows operating system that can address micro controller I/O, Windows 10 IoT Core is a welcome addition. The fact that Windows 10 IoT Core is the same Windows desktop kernel rather than different kernel with different drivers and API set will make overall development a little easier. If you are developing on x86 architecture, you will have access to all the Windows universal device drivers, which is far better than having to create drivers from scratch or hunting for drivers available from a 3rd party source like it was under Windows CE.

The icing on the cake, or what completes the reboot, is having tools that are free and support all Windows versions. Time will tell if Microsoft will get market share back in the IoT/Embedded space, but the reboot is going in the right direction.