Summary: Windows Embedded Compact 2013 is a componentized device operating system that gives developers all the tools they need to create the next generation of intelligent systems solutions. With support for x86 and ARM architectures, Windows Embedded Compact 2013 provides the flexibility and real-time support to help device manufacturers reduce time to market while creating easy-to-use, multitouch, small footprint devices that can connect to the cloud.

Windows Embedded Compact also offers the full Win32 API set along with the .NET Compact Framework, allowing device manufacturers to utilize the skills of Windows developers on their devices.
# Contents

Sensors to Cloud with Windows Embedded Compact 2013 ........................................................................................................................... 1

- Contents ............................................................................................................................................................................................................ 2
- Copyright ........................................................................................................................................................................................................... 2

**Introduction: Windows Embedded Compact 2013 and the Cloud** .................................................................................................................. 3

**Solution Overview** ........................................................................................................................................................................................................ 4

- The Device ........................................................................................................................................................................................................ 4
- The Monitor ........................................................................................................................................................................................................ 4
- The Cloud ........................................................................................................................................................................................................ 4

**How We Paint the picture?** .......................................................................................................................................................................................... 5

- Step 1: Storage in the Cloud ........................................................................................................................................................................ 5
  - ReST ........................................................................................................................................................................................................ 5
  - CRUD ........................................................................................................................................................................................................ 5
- Step 2: Information to and from the Device ................................................................................................................................................ 5
- Step 3: Monitor to Watch, React and Learn ................................................................................................................................................ 5

**Conclusion** .......................................................................................................................................................................................................................... 6

**Where do I go for more information?** ............................................................................................................................................................................. 6

© 2014 Microsoft. All rights reserved.

This document is provided “as is.” Information and views expressed in this document, including URL and other Internet website references, may change without notice. You bear the risk of using it.
The demand for connected devices is accelerating in day-to-day life across various consumer and business scenarios. Embedded devices generate large amounts of data—including machine-generated and user-driven data—but in the past, such data left some untapped opportunity due to the disparate nature of the systems. In the last few years, this trend has been changing rapidly. If these devices are Internet-enabled, this valuable data can be further harnessed in the cloud and the resulting analytics can be used for improving decision-making for enterprises and help redefine the customer experience. This is already a reality today in some cases. Though you may not always realize it, these intelligent devices exist, and they collect and send data to the cloud for processing, using the resulting analytics for either self-improvement based on the situation or to be tuned externally by other systems.

It should come as no surprise that intelligent devices powered using Windows Embedded Compact are already building cloud-ready solutions. For example, human-machine interface (HMI) panels built using Windows Embedded Compact can seamlessly collect data from various sensors as well as user generated data and then transmit the data to a backend system for analysis. This data can aggregate feedback gathered from similar devices in the backend system to provide broader insights on usage and workflow. Just think of all those small devices you interact with by pressing a button or touching a small screen to visualize how this data gets generated from these devices.

A great example of a device that generates data is a product catalog vending machine. This machine, based on its inventory level and demand, on its own, can change its display to advertise a specific product more than the other items it contains. However, if it were connected to the cloud, it could send signals to indicate its product demand status as well as inventory level to prompt a replenishment of optimal products with the correct volume. The vending machine company could use the analytics to make accurate investment decisions in terms of bringing in the right products, determining the best locations to place the vending machine, and so forth. The same data, when reviewed for multiple vending machines over a period of time, using a combination of product type, location, and volume of sales, can help the company adapt to changing trends and consumer preferences, much like web-based advertising because it’s tuned to a consumer’s location and prior navigation and purchase history.

From a developer point of view, it can overwhelming to think of all the different tools and dev environments you might need in order to create a solution that works end to end. From sensor to cloud. In this case we are discussing not just the service to monitor and collect the data but the device as well.
Solution Overview

“Where’s the data?” is a great question to ask at any point. The data-generation cycle is actually quite obvious. Data must be generated, then collected and stored somewhere that is accessible so that it can be monitored and reviewed to provide actionable insights to improve the customer experience or business operations.

**The Device**

A key component to an intelligent system is the device that generates the data. Be it a temperature sensor or a vending machine, it generates data that can be collected, processed, and analyzed. Start with the question “What do I want to know?” and look for relevant answers, then you will likely discover that this data is actually quite simple but, when this data grows in volume, it becomes extremely valuable.

Take the temperature sensor as a very simple example. Answers to the question of “Where is it located?” and “What is the temperature at any given time?” when used in a given context and over an extended period will present more useful answers and can provide solutions to complex problems.

The data can be useful for not just improving business operations but also improving customer experiences. Just think of location sensors in your car. It can provide the location information with respect to a GPS coordinate.

You as the driver of the car are interested in a comfortable interior temperature and also would like to ensure that the engine is not overheating. However, the automobile manufacturer could be getting this same data from cars all over the world to ensure all their components are designed and operating effectively everywhere.

The device is normally the closest point of interaction with the user, where valuable data from user interactions as well as from its surroundings—including the device itself—can be used to grow, expand, and streamline operations, and improve the user experience.

**The Monitor**

Data is not worth collecting unless there is a real-world scenario that needs improvements and there are a number of situations where that data has value.

- To respond to problems in or around an industrial device in a timely manner, that temperature sensor needs to be monitored for fault occurrences so that someone could be dispatched to fix the problems quickly.
- If the vending machine needs inventory replenishment or has been vandalized, a technician can be sent to restock or repair the machine.
- By reporting trends in temperature or sales over a time period can help with business review and long-term planning.

Note that there is no difference in the data itself, only in how it is being used. One little caveat here is that the monitoring systems may also provide data back to the device to control or activate a device. In the case of the temperature sensor, it may be attached to a thermostat and the monitoring device could send data back indicating that a cooling or heating system needs to be activated.

**The Cloud**

To support the device and the monitor, we need an accessible place to store and retrieve the data. A critical component of an intelligent system is the ability to store and manage all the data. In the past this might have been achieved with many hardware and software components that were needed to be installed, configured, and managed. Now we have cloud resources like Windows Azure, which is a perfect solution for this because it offers World Wide Web accessibility and well-understood communication protocols for sending and receiving data. It also provides a great opportunity to quickly implement and scale a solution, reducing significantly the time required to re-create or redesign a backend system to collect and monitor all the data.
How Do We Paint the Picture?

Data sits in the cloud, client devices/sensors post data to that cloud, monitoring applications read the data and potentially send updates. What needs to be created to make this all happen.

**Step 1: Storage in the Cloud**

A database is a well-known tool for storing and retrieving information so it is an obvious place to start. Unlike the data entry clerks of old, cloud databases can collect their information directly from the source using the following two mechanisms. Plus, they require just a few clicks to set up and configure in Windows Azure.

**ReST**

Representational State Transfer (ReST) uses the well understood web technology and protocols, including HTTP and XML, to offer a uniform interface across different data producers. It offers unambiguous client/server communication with a limited instruction set including only GET, POST, PUT, and DELETE. To avoid issues of synchronization, ReST is both stateless and cacheable, allowing device access anytime without worry if their data has been delayed. This stateless service also makes it easily scalable, which means adding compact devices to your system to meet growing business demands becomes that much simpler.

**CRUD**

Create, read/retrieve, update, and delete are basic database actions available to both the device and the monitor.

**Step 2: Information to and from the Device**

If you have never created a device before, but you are a developer who is familiar with Microsoft Visual Studio, you are in luck! The Windows Embedded Compact 2013 Developer Kit is a free add-on to Visual Studio and there are a full set of instructions to get you started at http://msdn.microsoft.com/en-us/library/jj200349.aspx.

After you have a device, the next steps are using the ReST and CRUD interfaces available from **Step 1** to both send and receive HTTP messages containing XML data. Windows Embedded Compact offers not only the Win32 Microsoft XML library and HTTP support, but also the .NET Compact Framework that includes “HttpWebRequest” and “HttpWebResponse” APIs to directly post and receive your XML information in a Microsoft Visual C# or Microsoft Visual Basic application.

**Step 3: Monitor to Watch, React, and Learn**

Using the word “monitor” is the best definition of what we want to do with any data that is of value to meet our device scenario. Using the same architectures we use to get the data, we can monitor the device as well. So what does that nuance give us? This means that a Windows Embedded Compact device is not only capable of being a generator of data but also may also be constructed to be a monitor.

Take a wind farm, for example. These great fields of large turbines generate power when they get the optimum wind through their propellers. However, the wind direction near them may be affected by the windmills themselves so having a wind direction sensor too close may give incorrect data. The Windows Embedded Compact monitoring device can get wind direction and speed data from multiple sources, such as government stations and airports nearby, as well as from other sensors, and triangulate the data for better accuracy. The monitoring device then calculates the best direction and instructs the windmill to turn to that vector.
Conclusion

Windows Embedded Compact is a valuable member of the existing information age and beneficiary of the advent of Big Data. This flexible, componentized Windows operating system runs on ARM and x86 chipsets, making it a great choice for small footprint devices connected to the cloud. Because it supports both the Win32 API set and the .NET Compact Framework, any Windows developer familiar with Visual Studio can both create and program Windows Embedded Compact devices, connect them to the cloud, and transform business intelligence into actionable operational intelligence.

With Windows Embedded Compact, Microsoft provides 15-year product availability, which is well suited for industrial market needs that require longer product life. What's next? Look for new ways to use Windows Embedded Compact in connected devices to collect data that can be used for developing better analytics and can be a critical part of the next generation intelligent systems.

Where do I go for more information?

Windows Embedded website:
www.microsoft.com/windows/embedded/default.mspx

Windows Embedded Compact webpage:
http://windowsembedded.com/compact

Windows Azure website:
www.windowsazure.com

Webinar on how to connect Windows Embedded Compact devices to Windows Azure:
www.youtube.com/watch?v=KUaTnBO0z-A&feature=youtu.be

Codeplex sample on connecting a Windows Embedded Compact based sensor to a Windows Azure database:
weccloudsensorsample.codeplex.com