Windows Embedded

Building Connected Devices with Windows Embedded Compact 7

Microsoft Corporation
December 2010

Applies to: Windows Embedded Compact 7

Summary: Windows Embedded Compact 7 introduces a number of new networking and connectivity technologies and updates capabilities incorporated in Windows Embedded Compact CE 6.0. Windows Embedded Compact 7 leverages a simplified development model similar to the desktop versions of Windows, providing increased flexibility that allows greater OEM customization and options that provide the next level of rich, immersive user experiences.

This whitepaper introduces the new features of Windows Embedded Compact 7 that enable new device scenarios and identifies a number of the updated networking/connectivity technologies included. It also discusses technologies that increase platform flexibility that better enable OEM customization.

Introduction

We are now well into the age of pervasive network connectivity and associated data services. Powerful PC-like networking incorporated into embedded devices is no longer just a wish; it is an expectation.

As a developer, your choice of operating systems becomes more complicated as the number and complexity of services and protocols increases. The task of developing or integrating disparate, third-party products becomes even more difficult and costly.

Windows Embedded Compact 7 introduces a number of new networking and connectivity technologies and revises and updates a number of technologies in Windows Embedded Compact CE 6.0. Windows Embedded Compact 7 includes a simplified development model that is similar to the desktop versions of Windows. It is more flexible and allows more OEM customization and options to provide richer user experiences.

This whitepaper introduces the new features of Windows Embedded Compact 7 that enable a number of new device scenarios and identifies a number of the updated networking and connectivity technologies. It also discusses the technologies that enable the platform to be more flexible and support OEM customizations.

Consumer Devices Drive User Demand

The current connectivity requirements for embedded systems are driven by other devices, most notably consumer electronics. This is similar to the late 1990s when consumer electronics first became touch enabled and embedded devices of that time were still using two-line character displays and keypads.

Driven by user demand, falling component costs, and competitive pressures, touch enabled graphical user interfaces (GUIs) with drag-and-drop functionality soon became the norm for embedded devices as well.

Connectivity

Today, users expect the same connected experience across all their devices. Users want to access their media, online services, and information not only on their desktop computers, but also on embedded devices. They expect a "connected experience" that extends across multiple devices.

What is a Connected Experience?

A connected experience is not only a device that connects to services or to another device through the cloud or over a wired or wireless network. Through connected experiences you can begin an activity on one device and then complete it on another. For example, you can purchase an eBook on a desktop PC or kiosk, browse the table of contents, and then open it and read it on a portable device.

Another possibility is combining different connectivity scenarios to create one desirable experience. An example is to listen to streamed media and at the same time purchase content from an online store. Another example is getting restaurant recommendations from a social network, and then having them appear on a portable navigation device. Places of interest are provided based on the physical location of the device and a user's personal profile and preferences using location-based services. With the connectivity included in Windows Embedded Compact 7 you can create these scenarios.

Networking and Connectivity Updates

A number of connectivity and networking technologies have been updated in Windows Embedded Compact 7. Many of these technologies are functionally equivalent to the technology in Windows 7 and Windows Server 2008. In addition to improved technology, the programming and application interface is similar to, and in some cases identical to that of Windows 7 and Windows Server 2008, making the technology more familiar and improving ease-of-use.

Connecting Windows with Devices

When incorporating connectivity between embedded devices and Windows PC, developers have typically developed their own application set or device driver and protocol. This required Windows desktop knowledge and resulted in each OEM reinventing the same functionality each time. For the OEM, device drivers are often difficult to write, hard to maintain and update, and are a frequent source of reliability problems.

In Windows 7, Windows Device Stage (WDS) was introduced, providing a centralized location, uniform user interface, and connection protocol to all WDS-compatible devices.

Windows Device Stage

Using WDS, your device can provide the discovery and installation of devices that are connected to a single PC via USB or TCP/IP using Microsoft Media Transfer Protocol (MTP). WDS also delivers a seamless user experience from the device to the PC and is where users can see and manage all their devices, including viewing information about the device such as available capacity and battery status on a portable device.

Windows Embedded Compact 7 includes the ability to create WDS-compatible devices. A default WDS experience is provided by Windows 7 as well as the ability to provide a customized OEM experience, including branding and additional device specific functionality as shown in the following two figures:

Figure 1. Default device experience

Figure 2. Customized, branded experience

Windows Device Stage reduces OEM development costs, reduces peripheral failures and improves the device connection experience for the user. Windows Embedded Compact 7 includes the ability to extend the default device experience to include OEM-specific branding, content, and functionality.

OEM Customizations

By providing a customized experience, OEMs can have users interact with their devices in a device specific manner within Windows Device Stage.

This customized experience can include logos and background images along with large device images. The customized experience can also include links to additional software and online services associated with the device, such as links to the OEM website, installers for additional software and services, manuals, and other services. Device-specific customizations beyond this may require an OEM provided device driver.

MTP

MTP is a protocol created by Microsoft to enable the seamless connectivity of portable and mobile devices to Windows desktop operating systems. It also allows optional interconnectivity between various peripheral devices such as a portable device and car infotainment system. Typically MTP is used to synchronize media data (audio, video, or pictures) between the connected devices.

In this release of Windows Embedded Compact 7, the MTP driver has been extended to support the following:

- Bi-directional media sync with the Windows Media Player
- Windows Device Stage (WDS) PMP (Portable Media Player) Profile support
- Notification mechanism on the device to inform apps on WDS to MTP Initiator connections

With the source code provided in Windows Embedded Compact 7, you can modify and extend the MTP Responder to support additional MTP commands and properties. By using an MTP extension, you can define new MTP operations, properties, and object formats that are not part of the MTP specification, but still operate within the Windows Device Stage.

Using MTP Device Services mode, you can add a service to the device that the PC can query.

For more information, see *<reference to MTP Responder whitepaper>*.

Updated Connectivity Infrastructure

Windows Embedded Compact 7 includes updated connectivity infrastructure in the following areas:

- Updated Networking Stack
- Bluetooth 2.1
- Wi-Fi
- Cellcore/RIL
- Connection Manager
- IPSec
- Kerberos V5

Updated Networking Stack

Windows Embedded Compact 7 includes an updated networking stack with an improved TCP implementation and support for SMP (Symmetrical Multi Processor) implementations. Windows Embedded Compact 7 also includes updates in the following areas:

- Winsock is updated to version 2.2
- The Network Driver Interface Specification (NDIS) is now compatible with NDIS 6.1 and backward compatible with NDIS versions 6.0 and 5.x
- Internet Protocol Helper (IP Helper) is implemented
- The Windows Filtering Platform (WFP) is added to the networking stack

These technologies are now programmatically and functionally equivalent to Windows 7 and Windows Server 2008.

Windows Sockets 2

Windows Sockets 2 (Winsock) enables programmers to create advanced internet, intranet, and other network-capable applications to transmit application data across the network, independent of the network protocol being used.

Winsock programming previously centered around TCP/IP, however some programming practices that worked with TCP/IP do not work with every protocol. As a result, the Windows Sockets 2 API adds functions necessary to handle several protocols while still maintaining backward compatibility with Windows 1.1. The interface for networking applications remains the same, therefore no additional effort is required to use the new Winsock 2 networking functionality.

NDIS 6.1

The NDIS stack is now NDIS 6.1 compatible. NDIS 6.1 is included in the Windows Server 2008 and Windows 7 operating systems and is backwards compatible with NDIS version 5.x implementations. NDIS specifies a standard interface between kernel-mode network drivers and the operating system. NDIS also specifies a standard interface between layered network drivers, thereby abstracting lower-level drivers that manage hardware from upper-level drivers, such as network transports.

NDIS 6.x provides enhanced performance and scalability including "NDIS Scatter/Gather DMA" and "NDIS Filter Drivers".

NDIS 6.x simplifies the NDIS driver model, including easier initialization with updated driver registration interfaces. These NDIS 6.x interfaces provide simplified driver registration and the ability to register optional services separately from required services. NDIS improvements that simplify driver development include:

- Streamlined driver initialization
- Versioning support for NDIS interfaces
- Simplified reset handling
- A standard interface for obtaining management information
- A filter driver model to replace filter intermediate drivers

IP Helper

IP Helper (Internet Protocol Helper) enables applications to retrieve information about the network configuration of the device and modify the network configuration. IP Helper also provides notification mechanisms to ensure that an application is notified when certain aspects of the network configuration change. IP Helper is distinct from the Management Information Base (MIB) and Simple Network Management Protocol (SNMP).

IP Helper also provides capabilities in the following areas:

- Managing network adapters
- Managing Interfaces and IP Addresses
- Using the Address Resolution Protocol (ARP)
- Retrieving Information on the Internet Protocol and the Internet Control Message Protocol (ICMP)
- Managing routing
- Retrieving Information about the Transmission Control Protocol and the User Datagram Protocol (UDP)

Windows Filtering Platform

Windows Filtering Platform (WFP) is a set of APIs and system services that provide a platform for creating network filtering applications. The WFP API is not a firewall and is distinct from NDIS Filters. Using the WFP, you can write code that interacts with the packet processing that takes place at several layers in the networking stack of the operating system. Network data can be filtered and modified before it reaches its destination.

By providing a simpler development platform, WFP replaces previous packet filtering technologies such as Transport Driver Interface (TDI) filters, Network Driver Interface Specification (NDIS) filters, and Winsock Layered Service Providers (LSP).

With the WFP API, you can implement firewalls, intrusion detection systems, anti-virus programs, network monitoring tools, and parental controls. WFP integrates with, and provides support for, firewall features such as authenticated communication and dynamic firewall configuration based on an application's use of sockets API (application-based policy).

Bluetooth 2.1

Windows Embedded Compact 7 has been updated to Bluetooth 2.1 and now supports Extended Inquiry Response (EIR), which includes the ability to filter devices that would appear in your device list. Secure Simple Pairing (SSP) is also included and makes it easier and more secure to pair two devices. With Bluetooth 2.1, encryption is mandatory for all connections aside from the Service Discovery Protocol. The ability to automatically refresh the encryption ensures that the connection is always secure.

Changes to how the device communicates allow the transmitter to be turned off much faster, resulting in power savings of up to 80% when compared to Bluetooth 2.0 implementations.

Windows Embedded Compact 7 also includes sample code for a Silverlight-based pairing and connecting user interface, to give you a starting point for developing a Bluetooth pairing and connecting interface.

Windows Embedded Compact 7 meets the latest Bluetooth SIG qualification requirements and has a new Bluetooth implementation and programming model that is consistent with the Bluetooth implementation on Windows 7.

Wi-Fi

Windows Embedded Compact 7 provides Wi-Fi support via the Native Wi-Fi automatic configuration component and replaces the previous Windows Zero Configuration (WZC) Wi-Fi stack in Windows Embedded CE 6.0 R3. Native Wi-Fi is the latest Wi-Fi networking implementation and is compatible with the same technology featured on Windows 7.

The Native Wi-Fi stack has the following improvements:

- Improved API
- Configurations based on XML profiles
- Improved extensibility of the networking stack. OEMs can develop their own security provider
- Less work in implementing an OEM specific Miniport Wi-Fi driver. OEMs no longer need to convert 802.11 frame packets to 802.3 frame packets in their Wi-Fi Miniport driver. The Wi-Fi filter driver provided by Microsoft in NDIS6.x can handle 802.11 frames directly.

  OEMs can port any desktop Wi-Fi Miniport driver to the new stack without large changes

The Native Wi-Fi automatic configuration component configures, connects to, and disconnects from wireless networks and can store profiles on the networks it interacts with in XML files.

As a Windows Embedded Compact 7 based device moves from one location to another, Native Wi-Fi searches for available wireless networks and notifies the device when there are wireless networks available which it can connect to.

The wireless network adapter is updated to match the settings of the selected wireless network automatically and attempts to connect in a secure manner. If the connection attempt fails, it will attempt to authenticate with any additional configured wireless networks. 802.11 open authentication is supported using a pre-configured Wired Equivalent Privacy (WEP) key.

Native Wi-Fi also supports Media Sense, which is a part of the Network Driver Interface Specification (NDIS) technology. Media Sense detects a move to a new wireless Access Point (AP). When a move occurs, Media Sense forces a re-authentication to ensure consistent network connectivity.

The Windows Embedded Compact 7 wireless client also performs a Dynamic Host Configuration Protocol (DHCP) renewal of the IP address configuration for the wireless network adapter. The IP address configuration does not change within the same Extended Service Set (ESS). When the wireless client crosses an ESS boundary into a new subnet, the DHCP renewal obtains a new IP address configuration for that subnet automatically.

Windows Sockets extensions and network-aware applications are notified of changes in network connectivity and can update their operation based on these changes. The reconfiguration and move detection minimizes the need for mobile IP when a wireless device roams to another subnet.

Although the Wi-Fi networking stack still supports the legacy NDIS5.x Wi-Fi Miniport driver, we recommend that you develop new, native 802.11 Miniport drivers for NDIS6.0. The support for legacy Miniport drivers will be removed in next release of Windows Embedded Compact.

CellCore/RIL

CellCore is a basic set of wireless connection-oriented services and has been in Windows Mobile devices since V1. CellCore provides Windows API support for digital cellular technologies and an extensible framework that supports a SIM interface, SMS and WAP communications, and GPRS support along with other cellular services.

CellCore abstracts the underlying networks and hardware, eliminating the need for you to directly interface to the cellular radio hardware and networks using your own interfaces. CellCore provides an interface to GSM and CDMA networks via a single API.

RIL

The Radio Interface Layer (RIL) driver is the interface between the various cellular services and the physical radio communications hardware. In most cases OEMs will need to provide the RIL device driver specific to the radio hardware.

The RIL driver services system requests for radio functionality (voice, data, SMS, etc.), and notifies the system of changes in the radio state (coverage, signal strength, incoming calls, etc.). The RIL contains all of the radio-specific code for voice and data networks including 3G, GSM, CDMA and EDGE.

In Windows Embedded Compact 7, the RIL driver model is based on the layered driver model and provides an additional level of abstraction. The MDD (Model Device Driver) layer is defined by Microsoft and includes the code that is common for simple 3GAPP AT based radios. The PDD (Platform-Dependent Driver) layer implements the radio-specific features and typically will be provided by the cellular radio manufacturer. This new model simplifies the task of implementing the RIL.

Another new feature is that the RIL APIs do not poll the RIL driver, instead uses a notification-based system that results in better performance and overall stability.

Connection Manager

Connection Manager in Windows Embedded Compact 7 is an extensive redevelopment of the Connection Manager that shipped in previous versions of Windows Embedded CE 6.0.

Connection Manager manages network connections for devices, regardless of the device and service provider used for establishing the connection. Connection Manager provides a fast and transparent way to make connection choices for an application. Users do not have to know which connection path is chosen, yet they can be assured that the most optimal connection is used at all times, based on availability and the most appropriate connections taking into consideration bandwidth, security, cost, or specific considerations for the application.

Connection Manager provides uniform application access to several data networks, including cellular data, Ethernet, Wi-Fi, wired, and Desktop Pass-through (DTPT). An application can specify its connection requirements and Connection Manager finds the best connection. If the connection becomes unavailable, Connection Manager notifies the applications.

Connection Manager is now easier to integrate into your implementation because the application developer doesn't need to understand or be concerned with the intricacies of a particular data connection. The previous distinction between "Work" and "Internet" connections is gone and the connection proxy settings improved, but they still need to be set on a per connection basis.

All Connection Manager requests are now routed through WinInet, which provides a uniform programming interface. Connection Manager can be customized and extended for OEM specific requirements such as supporting networking behavior for a specific bearer and to enable new networking technologies such as WiMAX.

Connection Manager also provides fail-over and automatic suspension and resumption of cellular data connections during voice calls.

IPSec

Internet Protocol Security (IPsec) is an open protocol suite for securing Internet Protocol (IP) communications between nodes. IPSec provides authentication and encryption of each IP packet, includes a mechanism to provide authentication at the beginning of a communications session, and negotiation of cryptographic keys used during the session.

Windows Embedded Compact 7 updates IPSec with a new implementation that is more robust and uses the Windows Filter Platform (WFP). IPSec is now compatible with the Windows 7 IPSec implementation and the configuration is similar. IPSec applications written for Windows 7 can be more easily migrated to Windows Embedded Compact 7.

The previous IPSec V4 API is deprecated, but is still available for backward compatibility.

Kerberos V5

The Kerberos protocol provides a mechanism for network clients to authenticate to a server in an open and insecure network.

Kerberos V5 is the primary security protocol for authentication within a domain. The Kerberos V5 protocol uses mutual authentication, which verifies both the identity of the user that is requesting authentication as well as the server providing the requested authentication.

The Kerberos V5 authentication mechanism issues tickets for accessing network services. These tickets contain encrypted data, including an encrypted password, that confirms the user's identity to the requested service. Except for entering a password or smart card credentials, the entire authentication process is invisible to the user.

An important service within Kerberos V5 is the Key Distribution Center (KDC). The KDC runs on each domain controller as part of the Active Directory service, which stores all client passwords and other account information.

Kerberos V5 supports Security Support Provider Interface (SSPI), which allows an application to use various security models available on a device or network without changing the interface to the security system.

In Windows Embedded Compact 7 Kerberos is RFC 4120 compliant.

Conclusion

Windows Embedded Compact 7 introduces an improved networking and connectivity platform. It includes a number of new and updated technologies that support several of the latest security and networking standards. It simplifies the connection of devices to desktop PCs using Windows Device Stage and supports a customized user experience.

The updated connectivity infrastructure in Windows Embedded Compact 7 makes the task of incorporating key networking functionality easier. From both an OEM and application developer perspective, the new and updated functionality makes for an easier implementation more in line with the features and development processes of Windows 7.

Windows Embedded Compact 7 includes a powerful, comprehensive technology set in a single integrated platform. From a developers perspective it is more similar to desktop Windows versions than ever before.

Additional Resources

Windows Embedded website (http://go.microsoft.com/fwlink/?LinkID=203338)