Creating Next-Generation User Experience with Windows Aero, Windows Presentation Foundation and Silverlight on Windows Embedded Standard 7

Windows Embedded Standard uses the latest Technology included with Windows 7 to allow for rich, powerful user experiences. In the world of embedded devices, the user experience is a differentiation points that OEM's use to set their products apart from one another. This paper will discuss some technologies that Windows Embedded Standard 7 supports to help create that experience.

In this paper, we will discuss features such as the following:

Windows Aero

Windows Touch

Windows Presentation Foundation

Silverlight

Windows Sensor and Location Platform

Other features, not discussed in this paper, but important for User Experience consideration are as follows:

Custom Shell Component

Dialog Filter

Message Box Default Reply

Unbranded Startup Screens

Shell Launcher

For more information about these, see the paper Custom Shell in Windows Embedded Standard 7 <hyperlink>

Windows Aero

Windows 7 brings the latest in graphics technology to systems, and at the core of the operating system in both Windows 7 and Windows Embedded Standard 7, one can use the default Explorer shell for their device. This uses the Windows Aero Theme set. Windows Aero was introduced in Windows Vista. Windows Aero represents the largest user interface change for Windows since Windows 95. There are several key elements in Windows Aero that bring rich experiences to the desktop.

Visual Effects

Windows Aero uses in 7 include Aero Peek, Aero Shake, and Aero Snap. Other features include the optimization of windows for a Touch UI experience, such as using slightly larger buttons in the title bar. Windows appear slightly translucent in the normal and maximized mode. The taskbar also appears translucent and shows the desktop wallpaper through. The “show desktop” that formerly appeared in the quick launch for Windows XP is now on the right side of the taskbar, and appears as a translucent button, hovering over the button causes the “Aero Peek” function, and clicking it minimizes all windows to the taskbar.

Aero Wizards

The previous standard for wizards in Windows was Wizard 97, used thru Windows Server 2003. Aero Wizards replace this standard, and allow for things such as the check box that enables a user to select the check box to disable the display of the welcome page in a wizard for future uses. Aero wizards can be resized now, where they were a fixed size before. The purpose of the wizard is now more clearly stated at the top of a wizard window. The Commit page is new in Aero Wizard. This introduces the concept of enabling a user to “commit” the final action for the Wizard.

Notifications

The balloon notifications in the notification area in the Aero themes are less intrusive by gradually fading in and out, and not appearing at all if an application is in full-screen, or a screensaver is running. When these events occur, the balloon notifications are queued and displayed later.

Font

The primary font for the Aero UI is Segoe UI Typeface. The default font size has increased from 8pt to 9pt for Latin, Greek and Cyrillic character sets, to increase readability.

Phrasing Tone

Beginning in Windows Vista the user experience guidelines address the "tone" in which the writing of text is composed, in order to maintain consistency across the operating system. The guidelines call now for messages that are more objective. This includes using the second person, and using active voice in grammar. (such as "transfer photos to your hard drive") the guidelines also require avoiding using empathetic words like "please" and "sorry".

Hardware requirements

Aero also has documented hardware requirements for the system, such as CPU speed, memory requirements, minimum video capabilities, minimum hard disk size and free space requirements, Audio output, Internet access requirements. The details of these requirements are outside the scope of this paper.

Windows Touch

Windows Touch is a new feature that has been introduced in Windows 7, and is available in Windows Embedded Standard 7. The computer has supported touch input for many years. However the touch driver implemented by the hardware manufacturer typically used the HID driver model. Windows Touch represents a new method for input, outside the typical HID class. Primarily Windows touch supports touch input by using more than one finger, and can recognize concurrent touches on the screen. Windows touch also introduces new API elements to the operating system. (See MSDN for WM_TOUCH, and WM_GESTURE) The requirement requires at least two touches to be recognized. However, certain touch digitizers support more. Gestures are also included in the spec for Windows touch. This includes Pan, flick and Pinch. Windows Embedded Standard 7 enables OEMs to create custom gestures, at their discretion. The API elements for Windows touch are similar to the Microsoft Surface platform. However, applications targeted to Windows Touch do not run on Microsoft Surface There are elements of windows touch that are built into the core of the operating system. However, to take full advantage of Windows touch, developers must use the Windows Touch API. To learn more about the Windows touch API and programming for Windows Touch, please see the Programming Guide on MSDN

Windows Touch requires multitouch hardware, in order to be implemented. Please see the Microsoft Web Site for a list of compatible multitouch hardware. Windows Touch must also be able to support inertia, to correctly interpret acceleration in gestures. The programming guide explains this in more detail.

Windows Presentation Foundation

The Windows presentation foundation was implemented by using the .NET Framework 3.0 in Windows Vista. It was designed to present an alternative graphics interface in programming to GDI. WPF is built from DirectX. It enables the graphical elements to be offloaded to a GPU, for better performance, while keeping the CPU free to perform other tasks. WPF also implements a new design/develop methodology, separating the design elements graphically, from the business logic on the back end. The designer uses the Expression Blend tool to create a rich User Interface to an application that is written in the XAML markup language, similar to HTML. The developer uses Visual Studio to create the business logic of the application in Visual Studio, in the C# programming language.

WPF applications can target the desktop or the browser. Apps also enable rich control. This includes 2D and 3D graphical rendering and animation, and vector and raster graphics. WPF applications support data binding, which enables the application to read/update from external sources, such as XML documents. One of the stand out features in WPF, is the ability to embed external audio and video into the application. WPF is available natively on Windows Vista, Windows 7, and Windows Embedded Standard 7. There is also a version of the .NET Framework that supports WPF for Windows XP SP2 or later versions. Windows Server also Supports WPF.

Silverlight

Silverlight is the Web-based implementation of the Windows Presentation Foundation. Silverlight, currently in Version 3 is available for Window7 and Windows Embedded Standard. Silverlight uses the same design/development principles as WPF. This includes design in XAML and development in C#. The functionality is currently available in browser only, but provides a competitive solution to Adobe's Flash product. Silverlight integrates graphics, multimedia, animations, and rich interactivity into a single runtime. Silverlight is compatible with multiple browsers, and across multiple platforms. These include Windows, Linux (moonlight) and Mac OS. The fundamentals of Silverlight are also available to the forthcoming Windows Phone 7 Series, and Windows CE, in a product called Silverlight for Windows Embedded.

Silverlight supports WMA, WMV, and MP3 across all browsers, without actually using the Media player, Windows Media Player ActiveX or Windows Media browser plugins. As Windows Media 9 format is part of the VC-1 standard, Silverlight also supports VC-1 Video, but only in the ASF format. Silverlight 3 implemented playback of H.264 video.

Silverlight also supports the ability to dynamically load content from a XML document, than can be manipulated through the DOM interface. This model is consistent with AJAX technologies. Silverlight also exposes a downloader object that makes it able to load scripts, media assets, or other data types that are required by the Silverlight application.

Silverlight 2 and later versions introduced a technology called Deep Zoom that is a derivative of the Microsoft SeaDragon technology. It lets a user to pan and zoom in on a high resolution image, or a collection of multiple images. This technology reduces the time for initial load by downloading only the region that is being viewed at that time. Designers use animation elements to hide the jitter in panning during load.

Windows Sensor and Location Platform

The Windows Sensor and Location Platform, a new feature for Windows 7, brings the ability to computing devices to better be aware of the environment they are put in and to become aware of their surroundings, based on location.

For years, computing devices have used sensors to perform various tasks such as a light sensor to brighten and dim an LCD panel on a notebook, or a shock sensor for detecting sudden movement, and parking hard drives, to prevent damage. Hardware OEM's integrated sensors from various manufacturers into their designs, to bring a value add to the customer. The designs frequently were elegant in function for a specific purpose. However, they made it difficult for ISVs to innovate using these same sensors. Each sensor was implemented by using a service in Windows, that one generally would see running in the notification area. These services had context specific settings, and opened up a specific set of API's for those functions. Unless the OEM provided an SDK for these sensors, it was almost impossible to take advantage of the sensor in a given device. The overall value of the sensor is diminished, by making it a single purpose sensor.

For location awareness, while the device to achieve location is indeed a sensor, Microsoft implements a specific set of location API's in Windows, to do such things as help retrieve location, while improving user privacy. APIs in the platform include the ability for applications to prompt for users' permission to publish location data back to the application, which types of sensors to use, and the preferred order to use, if there are multiple location sensors (GPS, Wi-Fi, and so on) This makes the developer's job easier, by not having to manage the location sensors, instead they can focus on the application itself.

Windows 7 includes a Control Panel item that lets users enable or disable the sensors across the whole system, or by user. This also lets administrators decide which applications have access to the sensors. This, of course is an optional component for Windows Embedded Standard 7.

Microsoft implements a standard way to implement sensors and location awareness in Windows 7, by providing a standard Windows driver model to integrate sensors, and also a standard set of APIs by which all ISV's can take advantage of exposed sensors. The Windows 7 WDK has information published for hardware manufacturers on how to architect drivers for Sensor and Location. Developers can learn how to take advantage of these sensors, by installing the Windows 7 SDK.

Examples of types of sensors that can be implemented:

Sensor type	Sensor placement	Typical use
GPS	Built into the computer or connected to the computer	Location and orientation
Accelerometer	Built into the computer or attached to a person	Motion, orientation, and location
Proximity	Built into the computer or nearby	Presence and spatial awareness
Light	Built into the computer, nearby, or remote	Ambient light conditions, such as room lighting or outdoor daylight
RFID	Built into the computer or attached to a person	Identification of physical items, automatic logon, or location of people
Compass	Attached to a person	Orientation
Camera	Built into the computer, attached to a person, or remote	Recording, object identification, or facial recognition
Microphone	Built into the computer, attached to a person, or remote	Ambient noise
Temperature	Built into the computer or remote	Indoor or outdoor ambient temperature and cooking temperature
Moisture	Remote	Detection of water leaks or lawn sprinkler operation
Motion detector	Built into the computer or remote	Security systems or sensing of motion near the computer
Traffic	Remote, usually public	Travel time predictions
Weather station	Remote, possibly public	Weather conditions and predictions

As designers and developers work with User Interfaces in Windows Embedded Standard 7, individual sensors, or even combinations of sensors can be used to improve the overall experience. For example a digital sign can use a proximity sensor to detect when someone walks in front of the sign, to change to an attractor loop in the video, and when someone approaches a camera sensor can use facial recognition to determine whether the subject is male or female. Based on the sex of the subject, different content can be played. The sign could also use a GPS sensor, or use Wi-Fi triangulation to better determine the type of ads it should play based on geography. The possibilities are almost limitless.

Next Generation devices, created by using Windows Embedded Standard 7 have the built-in ability to use the Sensor and Location Platform, thanks to Standard 7 being derived from Windows 7. Location awareness of devices is also a way to improve physical security or help recover lost or stolen devices, much like LoJack™ does with vehicles and notebooks, and more

Conclusion

Windows Embedded Standard 7 brings the most powerful version of Windows yet to the Embedded World. Multiple new technologies that enable developers to implement a rich, powerful user experience are built into this operating system, and in most cases are in the core of the OS itself. OEM's can now differentiate themselves from their competition, by taking advantage of the latest hardware, and the latest, up to date programming models. More information on technologies built into Windows Embedded Standard 7 is available at http://msdn.microsoft.com/embedded