

White Paper ▶

Building Kiosks using .Net

Introduction

Kiosks (Touch Screen Monitor) seem to be everywhere these days. Whether you go to an airport and check in, get cash out of an ATM, or purchase groceries in an automated check-out line, kiosks are now an important part of everyday life. However, innovation with kiosks is far from over. This paper will provide an overview of these new technologies and relate them to benefits that a developer will see for creating rich kiosk solutions. This paper also deals primarily with the software development component of building kiosk solutions.



Kiosks Defined

Microsoft Encarta defines a kiosk as a “small roofed street booth” that sells merchandise (for example, a kiosk that sells fruit in a bazaar). But for the purposes of this paper, we will define a kiosk as an enclosed device made up of hardware and software that allows the user to interact with it through simple interfaces (for example, touchscreen, keypad, and so on). The internals of the kiosk are usually based on some sort of personal computer technology. Usage of the kiosk requires very little, or no, training.

Kiosks come in all shapes and sizes, depending on the tasks they were designed to handle.



Enabling Kiosks with New Microsoft Technology

Microsoft has a number of technologies coming in the future that will enable new retail kiosk scenarios. Retailers should consider these products and technologies when considering how to build out next generation kiosks strategy.

Windows Presentation Framework (WPF)

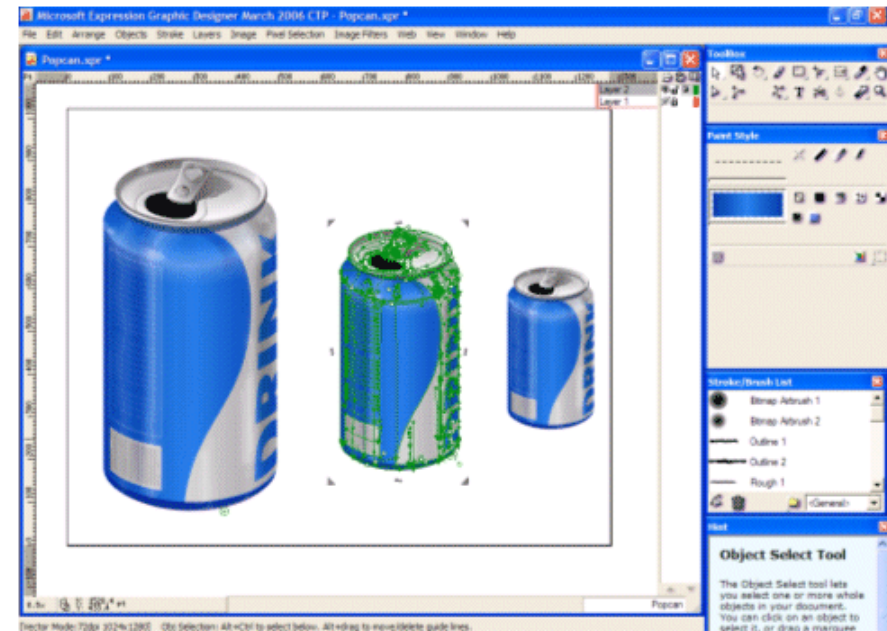
The Microsoft Windows Presentation Foundation provides the foundation for building applications and high fidelity experiences in Windows, blending together application UI, documents, and media content. WPF introduces a new XML forms engine call XAML (XML Application Markup Language, pronounced “zamel”) for defining how an application UI should be displayed. XAML uses XML to define a form through XML tags, and allows the developer to set the properties of a tag with attributes.

WPF introduces the following abilities that are directly relevant to kiosk development

Video - High-definition video (HD) or regular video (NTSC/PAL) can be hosted on a form, and manipulated from code. Effects such as inverting the video, dynamically sizing the video, rotating the video, and animating the video within the form are now all possible with WPF through a simple API.

Benefits for kiosks - Product video can now be easily incorporated into kiosk applications, in order to display product information to an interested consumer. Video can be used to gain consumer interest, and it can then be responded to when a user “clicks” the video.

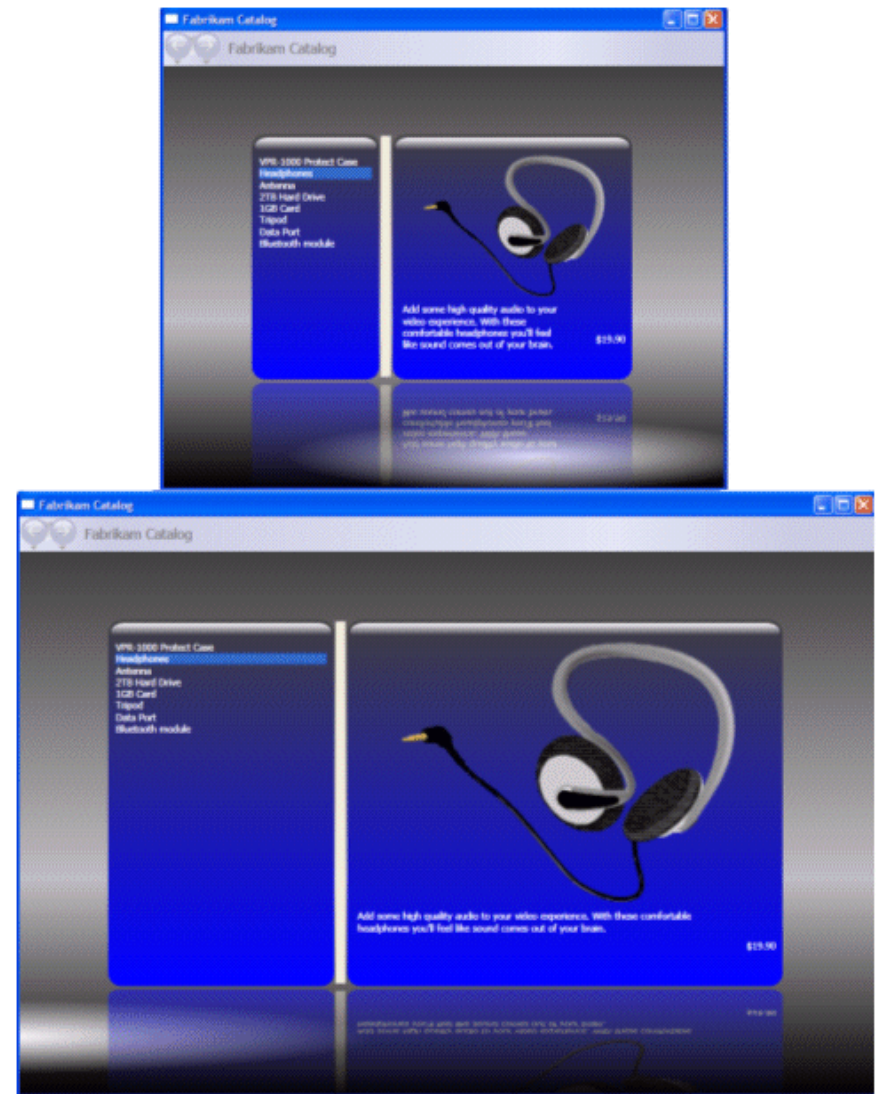
3D vector graphics - Graphic files that use vectors instead of bitmaps to display an image can be used in forms. This enables the developer to resize these graphics without loss of resolution due to “pixelization” of the bitmap.



Benefits for kiosks - A single product graphic can now be used in multiple scenarios in an application. For example, a single graphic could be used as a product thumbnail, and when selected, expand to a larger size that reveals more detail. Historically, two graphic files would be needed: one for the thumbnail, and another for the larger size. Graphics can be resized dynamically, according to the requirements of your application.

Resolution agnostic - it's common for rich applications to be designed for a specific resolution. The challenge to a developer comes when a rich application needs to run on multiple resolutions, or when future requirements dictate a resolution that was not anticipated in the original design. WPF applications are designed to “grow and shrink” dynamically, based on the needs of the application and the target display resolution.

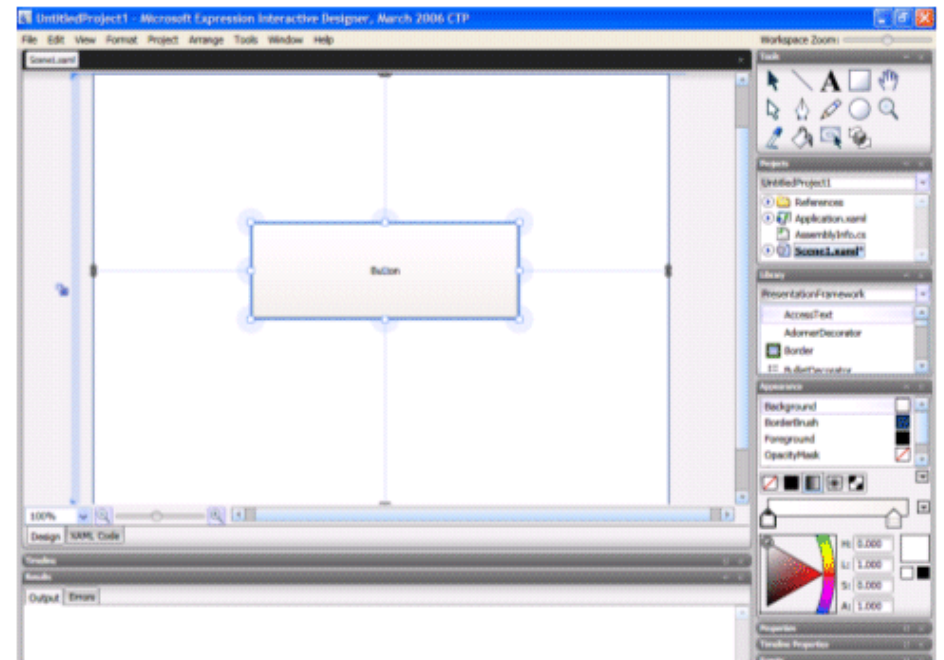
The first image represents the application after it first launches. The second image is a screen capture of the application after the form has been resized and enlarged. Notice how the application responds appropriately. The panels are resized, the text in the right details pane is reflowed, and the image of the headphones is resized, without any loss of resolution. Even the reflection below the details pane is resized correctly.



Forms defined by “markup” - By defining forms using XAML, it is possible to use a variety of tools to generate the form, based on a targeted skill set. For example, a User Interface designer has different needs than a developer. By sharing a common forms markup language, different tools targeting different roles can be utilized on a single project. A designer can have a tool focused on creating a certain “look and feel,” while a developer can connect the user interface to business and data tier logic.

Benefit for kiosks - Designers can use graphical design tools targeted to their needs.

Any application that supports XAML-based forms can open and render the form, creating a platform for a variety of tools vendors to create solutions to solve specific technical requirements.



Conclusion

The next-generation kiosk will require a new set of tools to enable new, compelling business scenarios in retail that weren't possible before. Because of these new platforms and tools, new kiosk scenarios are enabled through rich graphics, video, workflow, reporting, and connectivity.

 [*info@hexacorp.com*](mailto:info@hexacorp.com)

 www.hexacorp.com