Building Mobile Applications

...for Java developers

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Introduction

Mobile devices and smart phones have become powerful pieces of hardware that can run increasingly complex applications. As a Java developer, you are probably familiar with the Android operating system that runs on many mobile phones, or Java Platform, Micro Edition (Java ME) for building solutions on other Java-enabled devices. The Microsoft® mobile platform is Windows® Phone 7. The Windows Phone 7 application platform provides two frameworks for you to develop applications: Microsoft Silverlight® and the Microsoft XNA® Framework.

If you are an Android developer, you will be pleased to learn that there are many similarities between the Android and Windows Phone 7 platforms, although there are some significant differences, too.

If you are a developer with Java ME experience, you will find that the Microsoft platform offers a much more modern approach to application development.

Contents

The Platforms and Frameworks

It is important to understand the fundamental differences between the Android and Java ME platforms, and Windows Phone 7. Applications in Windows Phone 7 are created as Microsoft .NET Framework assemblies that are written in a programming language such as C# or Visual Basic.

The Windows Phone 7 platform supports two different frameworks: Silverlight and XNA.

Silverlight

Windows Phone 7 implements the full feature set of Silverlight that is available for building graphical user interface (GUI) applications that can run on the desktop and in a Web browser. However, there are some slight differences with platform specifics.

Silverlight-based applications have many similarities to Android applications. A typical Silverlight application consists of one page or more; a page is similar to the concept of an **Activity** in the Android framework.

On a page, you construct the user interface from familiar controls such as the **Button**, **TextBox**, and **Label** controls. These are equivalent to **Widgets** in the Android platform.

Did you know?

The Silverlight implementation for Windows Phone 7 enables you to write applications that have the same look and feel across the phone, the Web, and the desktop, and maintain a common code base across all three platforms.

Android and Silverlight both implement mechanisms for separating the user interface from the programming logic. Silverlight follows the Windows Presentation Foundation (WPF) model for implementing a user interface. In a Silverlight application, you write XAML code to define the visual presentation and you implement the programming logic in a code-behind class that is written in a language such as C#. This model is equivalent to the practice of storing the user interface markup in standard XML files and the programming logic in the **Activity** class files in an Android application.

The Silverlight runtime uses user interface markup and a retained mode graphics system to define and render the content. A Silverlight application defines a model that consists of a set of classes and binds this model to elements in the user interface. When the data in the model changes, the user interface is updated automatically. The graphical content that Silverlight displays is vector-based and can produce very rich and modern user interfaces. This development model is completely different to developing by using any of the Java ME GUI application programming interfaces (APIs), which typically require programming at a lower level and do not provide the same graphical resolution.

XNA

XNA is a more graphics-focused framework that enables developers to create two-dimensional (2-D) and three-dimensional (3-D) games applications. In addition, XNA is a more advanced framework that is much more extensible and provides a higher granularity of device interaction than is possible by using Silverlight. However, the XNA Framework does not include any predefined controls for user interface items and relies on the developer to create these elements from scratch. In many ways, developing by using the XNA Framework is similar to building graphical applications by using the Mobile 3D Graphics API for Java ME.

A typical XNA-based game application defines objects that are equivalent to the Android **SurfaceView** and **GLSurfaceView** objects. In addition, XNA games target Microsoft Direct3D®, which is a Microsoft technology that is based in 3-D. This differs from the Android and Java ME platforms, which target OpenGL.

Platform Independence

Windows Phone 7 builds on the success of the modern Windows development platform. You can use the existing infrastructure and consume existing software services as they are. If you have any services that were developed by using Java, you can access these services from Windows Phone 7 by taking advantage of Windows Communication Foundation (WCF).

Any application code that you write in Silverlight or XNA is also portable and involves only a few small changes in most cases. You can port graphics code that has been written in XNA to the PC, Xbox 360®, and Zune® media devices, enabling a single game to run across a multitude of platforms.

The Windows Phone 7 devices offer a choice of hardware, but they all adhere to a capability baseline to offer full compatibility with all devices.

Tools

Microsoft provides a set of tools for Microsoft Visual Studio®, enabling you to build Windows Phone 7 applications quickly and easily. These tools include the Windows Phone 7 project templates for Visual Studio 2010 along with an emulator for testing your applications. The emulator is actually a small virtual machine that is running the latest version of the operating system.

Did you know?

In addition to a fully featured code editor and interface designer, Visual Studio 2010 also offers full debugging support, which enables you to set breakpoints and step through your application while it is running.

These tools and templates are analogous to the items that the Android software development kit (SDK), or the Eclipse Mobile Tools for Java (MTJ) kit, provide. These SDKs include templates to assist in building and testing applications and, in the case of the Android SDK, provide an emulator for testing. If the commercial version of Visual Studio is not available, you can download the Windows Phone Developer Tools from the Microsoft download site. These tools include Visual Studio Express (the free version of Visual Studio) together with the templates, emulator, and other utilities.

User Experience

One of the key differences between Windows Phone 7 and Android is the user interface styling. Windows Phone 7 offers a clean, elegant, and modern user interface that takes design cues from Metro, the Windows Phone 7 design language. This visual style is built into the controls that you use with the phone design tools. This means that your Silverlight-based applications will implement this interface immediately and offer users the same look and feel as the built-in functionality on the device.

Summary

There are many similarities and differences between the Windows Phone 7 platform and Android or Java ME.

The richness of the Windows Phone 7 platform enables you to be very creative about the applications that you develop.

Features such as live tiles that update dynamically and fully featured customizable controls that are available out of the box will really make your applications stand out.