



Developing an Operating System Design

Writer: Michael Stahl

Published: March 2011

Applies To: Windows Embedded Compact 7

Abstract

Describes how to design and build a Windows Embedded Compact 7 OS to deploy to a Windows Embedded Compact powered device or virtual CEPC.

- Using a design template or Board Support Package (BSP) as a starting point
- Using Platform Builder to select the catalog items to include in your design
- Using the Visual Studio IDE to build your OS design and run-time image

Contents

Introduction	3
Prerequisites	3
Designing an OS	3
Creating an OS Design	4
Creating an OS Design Project	4
Building an OS	5
Build Configurations	5
Generating Run-Time Images	6
Deploying a Run-Time Image	6
Conclusion	7
Additional Resources	7

Introduction

Windows Embedded Compact 7 (Compact 7) includes a suite of development tools that help you design, build, and deploy a Windows Embedded CE–based OS to a target hardware device. These tools are designed to make the development process straightforward as well as flexible.

Platform Builder, included with the Windows Embedded tools, customizes the Visual Studio integrated development environment (IDE). You can build custom embedded OS designs based on the Windows Embedded CE OS. Platform Builder includes an OS design wizard that makes it easy to select the components that you'll need for your OS. These components include predefined board support packages (BSPs) and design templates containing catalog items for the functionality that your OS can support.

You may implement your own BSP and boot loader prior to developing your OS, or you may choose from the set of preconfigured BSPs that ship with Compact 7.

Run-time images resulting from your design may be built either using Platform Builder or by using command-line tools. Once the run-time image is created, the image may be deployed to a Windows Embedded CE PC-based device (CEPC) or virtual CEPC.

See MSDN for more information on Platform Builder, BSPs, and how to get started with virtual CEPC.

Prerequisites

Before you can begin developing an OS design, you must install the following software in the order listed below.

1. Visual Studio 2008
2. Visual Studio 2008 Service Pack (SP) 1
3. Windows Embedded Compact 7

Designing an OS

The process of designing an OS includes creating the initial OS design, and then modifying and refining the design until the OS has all the functionality and support required for the target device.

A design template is a possible starting point for an OS design. You can begin an OS design by choosing a design template, or you can begin an OS design by selecting catalog items individually, depending on the requirements for your device.

Creating an OS Design

The first step in creating an OS design is to create an OS design project in Visual Studio. After you create the project, you configure your OS design using the Platform Builder OS Design Wizard. Through a series of interactive dialog boxes, you can use the wizard to perform the following tasks.

- Choose a BSP based on the desired target hardware.
- Select a base design template for your OS design.
- Select OS catalog items.

After you choose your initial settings, the Platform Builder OS Design Wizard sets up the environment with files that support the OS design, based on the design template that you selected.

Creating an OS Design Project

The process for creating an OS design project in Visual Studio is similar to creating other types of projects. The primary difference is that once the project is created, the Platform Builder OS Design Wizard is launched.

To create an OS design project

1. In Visual Studio, on the **File** menu, click **New** and then click **Project**.
 2. In the **Project types** pane, select **Platform Builder**.
 3. In the **Templates** pane, select the **OS Design** template.
 4. In the **Name** field, type a name for your OS design, and then click **OK**.
 5. When the OS design wizard appears, click **Next** to advance the wizard.
 6. Select the **Virtual PC : x86 BSP** and then click **Next**.
 7. Select the **Enterprise Device** design template category.
 8. Select the **Handheld** design template and then click **Next**.
 9. If there are design template variants available, select the variant that you want to use, and then click **Next**.
 10. If a limited selection of catalog items is presented, select the items you want to include in your OS design, and then click **Next**.
- or-
- If you want to accept all the default options that appear, click **Finish**.
11. If notifications associated with the catalog items in your OS design appear, review the notifications and click **Acknowledge**.
 12. On the final page of the wizard, click **Finish** to create the new OS design.



Note

Choosing to build the configuration after you click **Finish** may take a significant amount of time, depending on development hardware and the particular components chosen for your

OS design. It is recommended that you save and review your project settings before beginning the build process.

After you have created an OS design, you can review special notifications at any point from the **Catalog Items View** window. Catalog items with special notifications are denoted by a red exclamation point next to their selection check box. To view the notification for a particular catalog item, right-click on the catalog item and select **Show Notification** from the context menu.

For additional information about the standard BSPs and design templates that ship with Windows Embedded Compact 7, see the Board Support Package (BSP) and Design Template topics in the product documentation.

Building an OS

At various points throughout the OS design process, you will want to build and test the resulting run-time image. Platform Builder provides a convenient way to build your run-time image through the IDE. The actual commands that Platform Builder executes are identical to the commands that you use to build on the command line.

To build a run-time image of your OS design using the Platform Builder IDE, select **Build Solution** from the **Build** menu. You may also build your run-time image by right-clicking on the project in **Solution Explorer** and selecting **Build Solution** from the context menu. This will execute Blddemo.bat, the root batch file that builds the project subcomponents.

Note

While building your OS design, build information in the IDE output window may halt and the compiler may appear to be frozen. You can verify that your build is still executing by observing the animated build icon on the bottom bar of the IDE.

Build Configurations

There are three basic build configurations: Debug, Retail, and Checked. A fourth build configuration for a final shipping build is available by enabling the Ship build option on a Retail build.

The following table lists the types of builds that are available and the attributes of each.

Build Configuration	Compiler Optimizations	Asserts	Retail Messages	Debug Messages
Debug	Off	On	On	On
Retail	On	Off	On	Off
Checked	On	On	On	On
Ship	On	Off	Off	Off

After creating an OS design, the Retail configuration is enabled by default.

To select the build type

1. Open the **Build** menu and select **Configuration Manager**.
2. In the **Active solution configuration** drop-down list box, select the desired build type.
3. Click **Close**.

To configure retail build for shipping

1. From the **Project** menu, select **Properties** for your OS design.
2. In the **Property Pages** window, open the **Configuration** drop-down list box and select the **Release** configuration.
3. Under **Configuration Properties**, select **Build Options**.
4. On the **Build Options** pane, select **Enable ship build** and set its value to **Yes**.
5. Click **OK** to accept the changes.

Generating Run-Time Images

By default, Platform Builder builds the OS design files, but does not generate run-time images. Building only the design files saves time during the process of developing an OS design. When you're ready to deploy your OS, however, you will need to enable generation of run-time images as part of the build process.

To enable run-time image creation

1. Open the **Build** menu and select **Targeted Build Settings**.
2. Select **Make Run-Time Image After Building**.

Deploying a Run-Time Image

After building the run-time image, the next step is deployment. In a typical development scenario, the run-time image is downloaded to the target hardware for testing. Compact 7 provides a way to deploy and test a run-time image using a virtual CEPC, so you can develop and test your OS design without requiring special hardware. Instead, you create a virtual machine on your development computer for deployment.

For a fully detailed description of how to set up a virtual CEPC, see [Getting Started with Virtual CEPC](http://go.microsoft.com/fwlink/?LinkId=199788) (<http://go.microsoft.com/fwlink/?LinkId=199788>), also available at **%ProgramFiles%\Windows Embedded Compact 7\Documentation**.

Conclusion

This paper described the Windows Embedded Compact 7 OS design process from beginning to end using Platform Builder. The paper also described how to build a run-time image for deployment to a Windows Embedded CE PC-based device (CEPC) or virtual CEPC.

Additional Resources

- [Windows Embedded website](http://go.microsoft.com/fwlink/?LinkID=183524) (http://go.microsoft.com/fwlink/?LinkID=183524)

This document is provided “as-is.” Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2011 Microsoft. All rights reserved.