InstallShield 2008 Evaluator Guide

Part Number: ISP-1400-EG00
Product Release Date: 23-May-2007

Copyright Notice

Copyright © 2006–2007 Macrovision Corporation and/or Macrovision Europe Ltd. All Rights Reserved.

This product contains proprietary and confidential technology provided by and owned by Macrovision Europe Ltd., UK, and Macrovision Corporation of Santa Clara, California, U.S.A. Any use, copying, publication, distribution, display, modification, or transmission of such technology in whole or in part in any form or by any means without the prior express written permission of Macrovision Europe Ltd. and Macrovision Corporation is strictly prohibited. Except where expressly provided by Macrovision Europe Ltd. and Macrovision Corporation in writing, possession of this technology shall not be construed to confer any license or rights under any of Macrovision Europe Ltd. and Macrovision Corporation’s intellectual property rights, whether by estoppel, implication, or otherwise.

ALL COPIES OF THE TECHNOLOGY and RELATED INFORMATION, IF ALLOWED BY MACROVISION CORPORATION, MUST DISPLAY THIS NOTICE OF COPYRIGHT AND OWNERSHIP IN FULL.

Trademarks

Macrovision, FLEXnet, InstallShield, InstallShield Developer, InstallShield DevStudio, and InstallShield Professional are registered trademarks or trademarks of Macrovision Corporation, or other Macrovision companies, in the United States of America and/or other countries. All other brand and product names mentioned herein are the trademarks and registered trademarks of their respective owners.

Restricted Rights Legend

The software and documentation are “commercial items,” as that term is defined at 48 C.F.R. §2.101, consisting of “commercial computer software” and “commercial computer software documentation,” as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.2702, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §227.2702-1 through 227.7202-4, as applicable, the commercial computer software and commercial computer software documentation are being licensed to U.S. government end users (A) only as commercial items and (B) with only those rights as are granted to all other end users pursuant to the terms and conditions set forth in the Macrovision Corporation standard commercial agreement for this software. Unpublished rights reserved under the copyright laws of the United States of America.
# Contents

1 - Introduction ........................................... 1
   About This Evaluator Guide ................................. 1
   About InstallShield 2008 ................................ 1
   Upgrade Overview ........................................... 4

2 - Getting Started ........................................ 7
   Installing InstallShield 2008 ............................... 7
   Using the InstallShield 2008 Interface ..................... 7
      Start Page .................................................. 8
      Project Assistant .......................................... 10
      Installation Designer ...................................... 11
   Selecting a Project Type .................................. 12

3 - Tutorial on Support for Windows Vista and Windows Installer 4.0 ....... 15
   Part A: Creating a New Basic MSI Project in InstallShield .............. 15
   Part B: Targeting Windows Vista Systems ..................... 15
   Part C: Adding Files to Your Project .......................... 17
   Part D: Changing the Dialog Theme .......................... 19
   Part E: Creating Log Files for Installations that Are Run with Windows Installer 4.0 .................. 20
   Part F: Creating a Product Configuration and a Release .................. 22
   Part G: Specifying Whether the .msi File Requires Administrative Privileges .................. 24
   Part H: Specifying the Required Execution Level for the Setup Launcher .......... 25
   Part I: Building a Release .................................. 27
   Part J: Running the Installation ............................. 29
## Contents

### 4 Walkthrough of Additional InstallShield Features .................................................. 31

- Part A: Digitally Signing Files .................................................................................. 31
- Part B: Performing Validation .................................................................................... 34
  - Specifying Whether Validation Should Be Performed at Build Time ....................... 35
  - Specifying Which ICEs, ISICEs, and ISBPs Should Be Run During Validation .......... 35
  - Validating an Installation Package or Merge Module on Demand .......................... 36
  - Viewing Validation Results ...................................................................................... 37
- Part C: Using the UAC Shield Icon on Buttons ......................................................... 38
- Part D: Minimizing Reboots Through the Restart Manager Infrastructure ................. 39
- Part E: Adding and Sequencing Custom Actions ....................................................... 40
- Part F: Adding .NET Framework Redistributables to Projects .................................. 41

### 5 Additional Information ......................................................................................... 43

- Additional Resources ............................................................................................... 43
- System Requirements ............................................................................................... 44
  - For Systems Running InstallShield ....................................................................... 44
  - For Target Systems (Desktop Computers) ............................................................... 45
  - For Target Systems (Mobile Devices) .................................................................... 45
- Customer Feedback ................................................................................................. 47

Index ......................................................................................................................... 49
Introduction

About This Evaluator Guide

This evaluator guide introduces evaluators and reviewers to key functionality of InstallShield 2008. It provides information about and instructions for using some of its new features. This guide also provides references to additional resources that offer more detailed information about the product.

About InstallShield 2008

InstallShield 2008 contains many new features and enhancements. It also contains support for the latest technology.

New Features

InstallShield 2008 includes the following new features.

- New end-user dialog themes for Basic MSI projects
- Digital signing improvements
- InstallShield Best Practice Suite available in Premier Edition
- Support for the UAC shield icon on dialog buttons (Basic MSI projects)
- Ability to require end users to scroll through the EULA in the LicenseAgreement dialog
- Support for 64-bit self-registration of COM servers (Basic MSI projects)
- Expanded operating system condition settings in the Setup Prerequisite Editor
- Ability to suppress build warnings
- FLEXnet Connect support
Enhancements

InstallShield 2008 includes the following enhancements.

- Usability enhancements for releases
- Usability enhancements for custom actions and sequences
- Usability enhancements for the Files and Folders view, the Registry view, and the Redistributables view
- Automation interface enhancements
- XML file change enhancements
- Faster Direct Editor, String Table editor, and Files subview
- Enhanced User Account Control support for InstallScript projects
- Shortcuts view enhancements
- Windows Vista and Internet Explorer 7 support for One-Click Install installations
- Enhanced support for the SecureCustomProperties property
- Automatic downgrade prevention entries in Basic MSI and InstallScript MSI projects
- Changes for ALLUSERS and for the Customer Information dialog
- Ability to change the product version from the command line or through an MSBuild task parameter
- Ability to override Windows Installer property values from the command line or through an MSBuild task parameter
- Windows Installer properties used for IIS data are stored in the registry by default
- New setting for specifying whether an IIS Web server should allow the CMD command to be used for SSI #exec directives
- New Host Header Name setting for IIS Web sites
- Ability to install an IIS Web site and its virtual directories to the next available new site number
- Ability to specify whether new SQL connections should share the same Windows Installer properties
- SQLLogin dialog enhancements
- Database Import Wizard enhancements
- New Windows Installer property for specifying SQL connections that should not be installed or uninstalled
- Ability to remove unreferenced directories from the .msi File
- Ability to add DIM references to merge module projects
- Ability to specify COM+ component file destinations from within the component services view
- New Windows Installer property for specifying COM+ applications to be installed after the InstallFinalize action
• Additional predefined system searches for Basic MSI and InstallScript MSI projects
• New and enhanced Setup.exe and Update.exe command-line parameters (InstallScript projects)
• Downloader Web release type supports more location options for .cab files
• Downloader and Install from the Web types of Web releases embed all transforms
• Enhancements for patch display information
• Ability to specify the minimum initialization time
• InstallScript enhancements and new functionality
  • New LaunchApplication and WaitForApplication functions to supersede LaunchAppAndWait for launching applications with elevated privileges
  • New SdRMFilesInUse dialog and OnRMFilesInUse event handler for minimizing reboots through the Restart Manager infrastructure in InstallScript MSI projects
  • Ability to specify additional compiler include paths on a per-project basis
  • Enhancements for .NET Framework 3.0 support
  • New and updated functions for SQL support
  • New Is constant for checking if a user is in the Administrators group
  • New functions that enable a .NET library loaded through InstallScript to be unloaded before the installation completes
  • Support for enabling update notifications during an upgrade when the original installation did not include FLEXnet Connect
  • New RegDBDeleteItem function
  • New GetStatus function for objects
  • Changes to the ListWriteToFileEx function and related constants
  • Expanded InstallScript text substitution functionality

Support for New Technology
InstallShield 2008 also includes support for key technologies such as the following:
• Internet Information Services (IIS) 7
• SSL certificates
• 64-bit versions of .NET Framework 2.0 and 3.0
• Visual C++ 8.0 SP1 merge modules
• Microsoft SQL Server 2005 Express SP1
• Windows Embedded CE 6.x
• Windows Mobile 5.0 for Smartphone
Upgrade Overview

The following sections provide information about possible migration issues that may occur when you migrate projects that were created with InstallShield 12 and earlier to InstallShield 2008. For more detailed information, see Knowledge Base article Q113162.

Migrating Projects Created in Earlier Versions of InstallShield

If you use InstallShield 2008 to open a project that was created with an earlier version, InstallShield displays a message box that asks you if you want to convert the project to the new version. If you reply that you do want to convert it, InstallShield creates a backup copy of the project with a file extension such as .765 before converting it. Delete the .765 part from the original project’s file name if you want to reopen the project in the earlier version of InstallShield. Note that you cannot open InstallShield 2008 projects in earlier versions of InstallShield.

You can migrate projects that were created with the following versions of InstallShield to InstallShield 2008: InstallShield 12 and earlier, InstallShield DevStudio 9, InstallShield Professional 7 and earlier, and InstallShield Developer 8 and earlier. Note that projects that were created with InstallShield MultiPlatform or InstallShield Universal cannot be migrated to InstallShield 2008.

End of Support for Windows 9x, Windows NT 4, and Windows Me on Target Systems

InstallShield no longer supports the creation of installations for Windows 9x, Windows NT 4, and Windows Me systems. If end users have one of these operating systems on their computer and they try to run an installation that was built with InstallShield 2008, unexpected results may occur, unless the project includes launch conditions that prevent end users from running the installation on any of these legacy operating systems.

Although the legacy operating systems are no longer supported, you may have projects from earlier versions of InstallShield that target these operating systems. When you upgrade these projects to InstallShield 2008, InstallShield does not remove any references to any of the legacy operating systems. Therefore, an **Always show legacy platforms** check box has been added throughout the InstallShield interface wherever operating systems are listed. This check box determines whether the legacy operating systems are displayed in all of the operating system lists throughout InstallShield. If you select this...
check box, you can see whether the legacy operating systems are selected, and you can remove any references to the no-longer-supported platforms. Two examples of areas in InstallShield that have this check box are:

- The Installation Requirements page of the Project Assistant (Basic MSI and InstallScript MSI projects)
- The Platforms tab on the Project Settings dialog box (InstallScript projects)

For Basic MSI and InstallScript MSI projects, you may want to consider adding launch conditions that display a message if end users are running your installation on any of the legacy systems. For InstallScript projects, you may want to consider adding InstallScript code that uses the SYSINFO structure variable to check for these legacy systems, and then display a message if any of these legacy systems are present.

The **Always show legacy platforms** check box is an InstallShield interface–wide setting. If you select this check box from anywhere within any InstallShield project, InstallShield shows the legacy operating system check boxes throughout InstallShield for all projects that you open on your computer. Similarly, if you clear this check box, InstallShield hides the legacy operating system check boxes throughout InstallShield for all projects.

**COM Extraction**

When you use InstallShield to extract COM information from a COM server, InstallShield now puts the data in the **Registry** table, instead of in the **TypeLib** table. Microsoft strongly advises against using the **TypeLib** table, as described in the “TypeLib Table” topic (http://msdn2.microsoft.com/en-us/library/aa372092.aspx) on the MSDN Web site.

**Unused Directories Automatically Removed from the .msi File at Build Time by Default**

Note that if you upgrade a Basic MSI, InstallScript MSI, or Merge Module project that was created in InstallShield 12 or earlier to InstallShield 2008, the new Keep Unused Directories setting on the Build tab in the Releases view is set to No. Therefore, if a directory that is listed in the Directory column of the **Directory** table is not referenced in any known location in the .msi file, InstallShield removes it from the **Directory** table of the .msi file that it creates at build time. For Basic MSI and InstallScript MSI projects, this occurs after any merge modules are merged, but only directories that are present in the .msi file are removed; therefore, if a merge module contains new unused directories in its **Directory** table, the new unused directories are added to the installation.

In some cases, you may want to change the value of the Keep Unused Directories setting to Yes. For example, if your project includes custom actions that use directories that are not referenced in any other area of the project, you may want to select Yes for the Keep Unused Directories setting. If only a small number of directories are not referenced, you may want to consider leaving No for the Keep Unused Directories setting, and creating **CreateFolder** table entries for those directories.

**New UAC Shield Icon Property for Dialog Buttons (Basic MSI Projects)**

In the Dialog Editor of Basic MSI projects, a new Show UAC Shield Icon property is available for all button controls. If you select True for this property, the User Account Control (UAC) icon is displayed on the button when end users run the installation on Windows Vista systems.
Chapter 1: Introduction
Upgrade Overview

For any new Basic MSI projects that you create, the Show UAC Shield Icon property is set to True for the Install button on the ReadyToInstall dialog. If you upgrade a Basic MSI project that was created with InstallShield 12 or earlier to InstallShield 2008, the default value for the Install button’s Show UAC Shield Icon property is False. You can override the value for this button, or for any other buttons, as required.

**New Default Value for the Cache Path Setting for a Release**

The default value for the Cache Path setting for a compressed release in the Releases view is now set to [LocalAppDataFolder]Downloaded Installations. The previous default value was [WindowsFolder]Downloaded Installations, which may not be available to users on locked-down systems. If you migrate a project from InstallShield 12 or earlier to InstallShield 2008, the Cache Path setting is not automatically changed. Therefore, you may want to change that value. To modify the Cache Path setting, use the Local Machine panel in the Release Wizard.

**Windows Server 2003 Conditions and 64-Bit Windows XP Conditions for Setup Prerequisites**

The operating system version number is 5.2 for both Windows Server 2003 and 64-bit Windows XP. As a result, prerequisites that were created in InstallShield 12 detected 64-bit Windows XP as Windows Server 2003. Therefore prerequisites that required Windows Server 2003 could be installed on 64-bit Windows XP systems, and those that required Windows XP could not be installed on 64-bit Windows XP systems.

To resolve this issue, the Setup Prerequisite Editor in InstallShield 2008 has been enhanced to enable you to specify whether the target system is required to be a workstation, a server, or a domain controller.

To resolve this issue for an existing prerequisite that includes a Windows Server 2003 requirement or a 64-bit Windows XP requirement, open the prerequisite in the Setup Prerequisite Editor in InstallShield 2008. On the Conditions tab, select the condition that needs to be corrected and click Modify. In the **Select the operating system on which to run the setup requirement** box, select the appropriate operating system requirement. Doing this correctly sets the new Product (OS) Type setting to the appropriate workstation, server, or domain controller value.

**InstallScript One-Click Install Installations**

Setup.exe is no longer used as the setup player for InstallScript One-Click Install installations; Setup.ocx is now used instead. In order for a Setup.ocx file to be included in an installation, the new Generate One-Click Install setting in the Releases view must be set to Yes. If you upgrade an InstallScript project from InstallShield 12 or earlier to InstallShield 2008 and the Create Default Web Page setting in the Releases view is set to Yes, InstallShield sets the Generate One-Click Install setting to Yes automatically during the upgrade. However, if the Create Default Web Page setting is set to No and you intend to distribute the installation over the Internet, you must manually select Yes for the Generate One-Click Install setting after upgrading the project.

**Automation Interface**

If you use the automation interface with InstallShield or the Standalone Build, update your existing code to reflect the new ProgIDs: IswiAuto14.ISWiProject or SAAuto14.ISWiProject.

The Display Save Options dialog setting was removed from the Releases view. Therefore, the WebSaveOptionsDlg property, which corresponds with that setting, is no longer available for the ISWiRelease object of the automation interface.
Getting Started

Installing InstallShield 2008

If you obtain a download version of InstallShield 2008, you can install it by double-clicking the executable file.

If you obtain the CD version of InstallShield 2008, place the CD into your CD-ROM drive. The CD Browser that is automatically displayed has links for installing InstallShield 2008, viewing the release notes, and opening InstallShield 2008 documentation.

Using the InstallShield 2008 Interface

InstallShield 2008 has three main work areas: the Start Page, the Project Assistant, and the Installation Designer. You can navigate from one section to the other easily, using the tabs at the top of the interface and other embedded links.
Start Page

The Start Page provides both new and experienced users with quick access for launching recently opened projects, creating new ones, or learning more about various aspects of installation creation.

Figure 2-1: Start Page

The Start Page includes the following sections:

Table 2-1: Sections on the Start Page

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Tasks</td>
<td>Click a project task to quickly create a new project, open an existing project, or browse to one of the sample projects included with the InstallShield installation.</td>
</tr>
<tr>
<td>Help Topics</td>
<td>Frequently accessed help topics are listed in this section. To access the entire InstallShield Help Library from the Start Page, press F1 or click the Help Library link in the Resources section.</td>
</tr>
<tr>
<td>(Recently Opened Projects)</td>
<td>The section in the middle of the Start Page lists your most recently accessed projects, the project types, and the dates on which they were last modified.</td>
</tr>
<tr>
<td>Getting Started</td>
<td>Click the Getting Started heading for guidance on what areas of the InstallShield Help Library to read, based on your level of experience with InstallShield and installation-authoring tools.</td>
</tr>
</tbody>
</table>
Table 2-1: Sections on the Start Page (cont.)

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources</td>
<td>The Resources section contains a number of links to connect you to helpful InstallShield information.</td>
</tr>
<tr>
<td>Help Library</td>
<td>Displays the InstallShield documentation.</td>
</tr>
<tr>
<td>Support</td>
<td>Opens the Support page on the Macrovision Web site.</td>
</tr>
<tr>
<td>InstallShield Community</td>
<td>Provides a Web-based forum where you can join other InstallShield users, post questions, and search for answers.</td>
</tr>
<tr>
<td>Webinars</td>
<td>Directs you to free Web-based seminars that help you evaluate InstallShield and gain the most from your Macrovision products.</td>
</tr>
<tr>
<td>Downloads</td>
<td>Offers a place where you can download the latest InstallShield merge modules, objects, and setup prerequisites; service packs; patches; and more for the version of InstallShield that you are using.</td>
</tr>
<tr>
<td>Release Notes</td>
<td>Connects you to the release notes that are posted to the Knowledge Base on the Macrovision Web site.</td>
</tr>
<tr>
<td>Known Issues</td>
<td>Displays the Knowledge Base article that lists the known issues for the version of InstallShield that you are using and provides information about workarounds and resolutions.</td>
</tr>
<tr>
<td>RSS Feeds</td>
<td>Directs you to the Web page where you can subscribe to RSS feeds of InstallShield Knowledge Base articles.</td>
</tr>
<tr>
<td>Contact Us</td>
<td>To provide feedback about InstallShield or join our Customer Experience Improvement Program, click one of the links listed in this section.</td>
</tr>
</tbody>
</table>
Project Assistant

The Project Assistant helps you quickly and easily build a basic installation project by providing a simplified view of installation project tasks. The navigation links at the bottom of each page in the Project Assistant guide you through the installation creation process. Each page focuses solely on each task, and provides pertinent information along the way. Many users with straightforward installation requirements are able to build their project completely within the Project Assistant. Even users with complex design needs can get a quick start on their installation authoring by beginning with the Project Assistant; this helps them walk through the basic aspects of their project creation before they move on to the Installation Designer and the added flexibility that it provides.

Figure 2-2: Project Assistant
Installation Designer

The Installation Designer displays all of the views in the InstallShield 2008 interface that are available for your project type. Here you can add more complex and powerful elements to your installation project. You can create your installation project using the Project Assistant and then use the Installation Designer to fine-tune project elements. The Installation Designer and the Project Assistant run simultaneously. Any changes made in one are reflected instantly in the other. For example, if you remove a feature while in the Installation Designer, that feature is no longer present in your installation project and does not appear in the Project Assistant.

The Installation Designer has installation task groups, simplified drag-and-drop views, and more. For detailed information about each of the InstallShield 2008 views, refer to the InstallShield Help Library. You can press F1 from any view to see the help library topic that pertains to that view.

Figure 2-3: Installation Designer with the View List Displayed on the Left
Selecting a Project Type

InstallShield 2008 enables you to create many different types of installation projects, but for the beginner there is just one significant choice to make: Do you want to build your installation with Windows Installer or InstallScript technology? Deciding which project type is right for you depends upon your experience with Macrovision software and installation development and on your installation and deployment needs. Three project types are described below. For details about the other available project types, see “Project Types” in the InstallShield Help Library.

Basic MSI

Basic MSI projects use the Microsoft Windows Installer engine to drive the entire installation, including calls to any custom actions (InstallScript, VBScript, JScript, .exe files, .dll files). This project type is recommended when you want to do any of the following:

- You want to meet the Certified for Windows Vista program requirements.
- You want to avoid writing scripting code and want to instead set properties and make table entries.
- You want to maximize compatibility with administrative tools such as SMS.
- You want to maximize the ability of users or other third parties to customize the installation for redeployment.
- You want to upgrade an existing Basic MSI project.

InstallScript

InstallScript projects use InstallScript to control the installation. This project type is recommended for any of the following scenarios:

- You have advanced requirements for the end-user experience (the end-user dialogs), such as multimedia elements.
- You want to use full-screen billboards during the run time of your installation.
- You prefer authoring the project using a procedural language at its core rather than a set of database tables.
- You want to perform actions before and/or after the main installation has been run.
- You want to upgrade an existing InstallScript project.

InstallScript MSI

InstallScript MSI projects use both the Windows Installer engine and the InstallScript engine to drive the installation. This project type is recommended when you want to do any of the following:

- You want to meet the Certified for Windows Vista program requirements.
- You have advanced requirements for the end-user experience (the end-user dialogs), such as multimedia elements.
- You prefer authoring the project using a procedural language at its core rather than a set of database tables.
• You want to perform actions before and/or after the main installation has been run.
• You want to upgrade an existing InstallScript MSI project.

Tip: Repackager is a project conversion tool that is available in InstallShield 2008 Premier Edition. You can use this tool to convert InstallScript projects and InstallScript MSI projects to Basic MSI projects.

The Welcome Assistant

InstallShield 2008 provides a Welcome Assistant to help you determine which project type is best for your installation needs. The first time you launch InstallShield 2008, the Welcome Assistant requests information about your previous experience with Macrovision products, your software installation requirements, and your familiarity with various technologies. Based on this information, the Welcome Assistant recommends a project type to suit your needs. After you select the appropriate option provided through the Welcome Assistant, you can select the Start Page tab at the upper left of your screen to display the Start Page.

Figure 2-4: Welcome Assistant

Each subsequent time that you launch InstallShield 2008, the InstallShield 2008 Start Page is displayed. However, you can launch the Welcome Assistant from the Help menu at any time.
**New Project Dialog Box**

If you decide not to use the Welcome Assistant, you can select one of the project types found in the New Project dialog box. InstallShield displays this dialog box when you click New on the File menu.

![New Project Dialog Box](image)

**Figure 2-5:** New Project Dialog Box
Tutorial on Support for Windows Vista and Windows Installer 4.0

This section of the InstallShield 2008 Evaluator Guide contains step-by-step procedures that introduce you to a few of the new features in InstallShield 2008, as well functionality related to Windows Vista and Windows Installer 4.0. The instructions lead you through the creation of a Basic MSI project for Othello, an application whose files are installed to the InstallShield Program Files folder.

Part A: Creating a New Basic MSI Project in InstallShield

In Part A, you will create a new Basic MSI project.

Task

To create a new Basic MSI project:

2. On the Common tab, click Basic MSI Project.
3. In the Project Name box, type Othello.
4. Click OK.

InstallShield creates a Basic MSI project and displays the Project Assistant.

Part B: Targeting Windows Vista Systems

In Part B, you will specify that the installation that you created in Part A requires Windows Vista on the target machine.
Task **To specify that the Othello installation requires Windows Vista:**

1. In the navigation at the bottom of the Project Assistant, click **Installation Requirements.**
2. For the **Does your application require any specific operating systems** question, click **Yes.**
3. Clear all of the check boxes except the **Windows Vista** check box.

**Figure 3-1:** Installation Requirements Page of the Project Assistant

**Note:** Several of the predefined software requirements that are shown in Figure 3-1 are new for InstallShield 2008: Adobe Reader 6, Adobe Reader 7, and Internet Explorer 7.0. If your product requires any of these applications, you can use the Installation Requirements page to specify those requirements. You can also click the System Search hyperlink if you want to specify your own custom condition.

4. On the toolbar, click the **Save** button.

**Tip:** InstallShield also enables you to create operating system requirements for features and components. If a particular operating system condition is not met on a target system, the corresponding feature or component is not installed. Use the Features view or the Components view to create this type of condition.
Part C: Adding Files to Your Project

In Part C, you will add the Othello application files to your Othello installation project.

Task  
To add the Othello application files:

1. Click the Installation Designer tab.
2. In the View List, under Application Data, click Files and Folders.
3. In the Source computer’s folders pane, select the folder that contains the Othello application files (blank.gif, blue.gif, Othello.exe, and red.gif) that are installed with InstallShield. The default location for these files is:
   
   C:\Program Files\Macrovision\IS2008\Samples\Basic Install\Data Files

4. In the Destination computer’s folders pane, select My Product Name [INSTALLDIR].
5. In the Source computer’s files pane, select each of the files and drag them to the Destination computer’s files pane. If InstallShield prompts you to create a feature for those files, select OK.

Figure 3-2: Files and Folders View
Tip: InstallShield 2008 includes some new enhancements for the Files and Folders view:

- You can right-click a file in the Destination computer’s files pane and then click the new Open Containing Folder command. Doing so opens a Windows Explorer window and displays the folder that contains the file that you right-clicked.
- A new Add command is available when you right-click the Destination computer’s files pane. Use this command to display an Open dialog box that lets you browse to the file that you want to add to your project.
- The upper-right corner of this view has a new link (either Show Source Panes or Hide Source Panes). Use this new link to show or hide the two top panes—the Source computer’s folders pane and the Source computer’s files pane—in this view. You can hide the two panes, open a Windows Explorer window, and drag and drop files from the Windows Explorer window to the two remaining panes in InstallShield.
Part D: Changing the Dialog Theme

Dialog themes, a new feature for Basic MSI projects in InstallShield 2008, are predefined sets of images that give your end-user dialogs a unified and distinctive look. With the click of a button, you can select one of the available themes for your project, and InstallShield applies that theme to all of the interior and exterior dialogs, as well as the Setup.exe initialization dialog, in your project. You can easily preview each dialog from within the Dialogs view to see how it looks with the selected theme.

For Part D, you will preview the available themes and then select one of them for your Othello installation.

Task To preview themes and select one for your project:

1. In the View List under User Interface, click Dialogs.
2. In the Dialogs explorer, expand the Themes folder.
3. Click one of the themes under the Themes folder. In the right pane, InstallShield shows a screenshot of a sample exterior dialog with the selected theme.
4. In the Dialogs explorer, right-click the theme that you want to use, and then click Select.

Figure 3-3: Themes in the Dialogs View

InstallShield applies the selected theme to the dialogs in your project. In addition, InstallShield displays a red check mark on the selected theme’s icon in the Dialogs explorer.
Tip: You can also change the theme by clicking a button: In the Dialogs explorer, click the theme that you want to use. Then, in the right pane, click the Select button.

Some of the themes are available in both the Premier and Professional editions of InstallShield, and some are available in only the Premier edition. For more information, including sample screen shots of exterior and interior dialogs with each of the themes, see the “Dialog Themes” section in the InstallShield Help Library.

Part E: Creating Log Files for Installations that Are Run with Windows Installer 4.0

InstallShield provides support for the option to easily log installations that are run with Windows Installer 4.0 on a project-wide basis without having to use the command line or configure log parameters through the registry.

For Part E, you will specify that the Othello installation should be logged. You will also customize the types of messages that are logged.

Task

To specify that the Othello installation should be logged each time that Windows Installer 4.0 is used to run it:

1. In the View List under Installation Information, click General Information.
2. Click the Create MSI Logs setting, and then click the ellipsis button (...). The Windows Installer 4 Logging Options dialog box opens.
3. Select the Yes (MsiLogging set to default value of voicewarmupx) option.
4. Click OK.

InstallShield sets the Create MSI Logs setting to Yes and populates the MsiLogging property with the default value of voicewarmupx.
Chapter 3: Tutorial on Support for Windows Vista and Windows Installer 4.0
Part E: Creating Log Files for Installations that Are Run with Windows Installer 4.0

Figure 3-5: Create MSI Logs Setting in the General Information View

If the installation is run on a target system that has Windows Installer 4.0, the installer creates a log file and populates the MsiLogFileLocation property with its path. In addition, a **Show the Windows Installer log** check box is added to the SetupCompleteSuccess, SetupCompleteError, and SetupInterrupted dialogs.

Figure 3-6: SetupCompleteSuccess Dialog with the Windows Installer Log Check Box
**Part F: Creating a Product Configuration and a Release**

Once you have configured the files and other elements of your installation project, you are ready to create a product configuration and a release for your installation. A product configuration provides a means for grouping together releases that share similar properties, such as the product name, product code, and package code. You can configure each release for a different media type and according to different sets of requirements.

For Part F, you will use the Releases view in InstallShield to create a product configuration and a release for your Othello installation.
Chapter 3: Tutorial on Support for Windows Vista and Windows Installer 4.0
Part F: Creating a Product Configuration and a Release

Task  To create a product configuration and a release for the Othello installation:

1. In the View List under Media, click Releases.
2. Right-click the Releases explorer and click New Product Configuration. InstallShield adds a new product configuration with the default name Product Configuration 1.
3. Right-click the product configuration and click New Release. InstallShield adds a new release with the name Release 1.
4. Enter Othello as the name of the release.

![Figure 3-7: New Product Configuration and Release in the Releases View](image)

5. On the toolbar, click the Save button.

In InstallShield 2008, the release settings are now organized by category on several different tabs in the Releases view. The new organization makes it much easier to configure a release.
Part G: Specifying Whether the .msi File Requires Administrative Privileges

InstallShield includes a Require Administrative Privileges setting that lets you specify whether your .msi package requires administrative privileges for the Execute sequence of your installation. The default is Yes. If administrative privileges are required, Windows Vista may display a User Account Control (UAC) prompt for consent or credentials at run time when Windows Installer reaches the Execute sequence.

Note that if you select No but your .msi package tries to perform a task for which it does not have adequate privileges, Windows Installer may display a run-time error.

For Part G, you will ensure that administrative privileges are required for Othello. This enables end users to install Othello for all users on a target machine.

Task

To specify that administrative privileges are required for the Othello installation:

1. In the View List, under Installation Information, click General Information.
2. In the General Information explorer in the center pane, click Summary Information Stream.
3. Set Require Administrative Privileges to Yes.

Figure 3-8: Require Administrative Privileges Setting in the General Information View
Part H: Specifying the Required Execution Level for the Setup Launcher

To support UAC functionality on Windows Vista systems, InstallShield lets you specify the minimum execution level required by your installation’s Setup.exe file for running the installation (the setup launcher, any setup prerequisites, and the .msi file) on Windows Vista platforms. You can configure this for each individual release in your project.

InstallShield embeds a Windows Vista application manifest in the Setup.exe launcher. This manifest specifies the selected execution level. Operating systems earlier than Windows Vista ignore the required execution level. If you do not include a setup launcher with your installation, InstallShield does not embed the Windows Vista application manifest in the Setup.exe launcher.

**Note:** Windows Installer 4.0 is available on all Windows Vista systems; only Windows Vista supports this version of Windows Installer. Therefore, Windows Installer 4.0 is not available as a redistributable.

**Tip:** To learn more about how the manifest and other InstallShield settings affect whether Windows Vista displays a UAC prompt to elevate privileges, see “Minimizing the Number of User Account Control Prompts During Installation” in the InstallShield Help Library.

For Part G, you will specify for the Othello release that a setup launcher is required but that the Windows Installer engine is not. You will also configure this release so that the Setup.exe file is run with the least privileges.

### Task

**To specify the setup launcher requirements and the required execution level for a release:**

1. In the View List under **Media**, click **Releases**.
2. In the **Releases** explorer, click the **Othello** release.
3. Click the **Setup.exe** tab.
4. For the **Setup Launcher** setting, select **Yes (no MSI engine included)**.

   The Othello installation targets Windows Vista machines, which already have Windows Installer 4.0. Therefore, it is not necessary to include a Windows Installer engine with the Othello release.
5. Set **Required Execution Level** to **Invoker**.

![Figure 3-9: Required Execution Level Setting in the Releases View](image)

The available options for the **Required Execution Level** setting are:

- **Administrator**—`Setup.exe` runs only for administrators.
- **Highest available**—`Setup.exe` runs with the highest privileges that the current user can obtain.
- **Invoker**—`Setup.exe` runs with the least privileges. This is the default option.

The benefit of elevating the required execution level is that privileges can be elevated only once if necessary to run `Setup.exe`, and that these privileges can be carried over to all of the installation’s prerequisites and the `.msi` file without requiring multiple prompts for approval. Thus, if two of your prerequisites require administrative privileges, for example, you can change this setting to **Administrator**, and then end users are prompted only once during the installation, before Windows Installer runs the `Setup.exe` file.

**Important:** *If your installation elevates privileges and also launches the application at the end of the installation, the elevated privileges are carried over to the application. In most cases, running an application with elevated privileges on Windows Vista platforms is discouraged. If elevated privileges are needed for your installation but not your application, your installation should not launch the application.*

Note that an end user’s installation experience is more secure when installations are run with only the permissions that they need. Unless an application is designed to be run only by system administrators, it should be run with the least privilege.
Part I: Building a Release

Once you have designed your project in InstallShield, you are ready to build the release. InstallShield offers several methods for building a release:

- Use the Release Wizard.
- Use the Releases view.
- Use TSCmdBld.exe from the command line.
- Use the Standalone Build. The Standalone Build is available only in the Premier edition of InstallShield.

For Part I, you will use the Releases view to build the release that you created in Part F of this tutorial.

To build the Othello release through the Releases view:

1. In the View List under Media, click Releases.

2. In the Releases explorer, right-click the Othello release and click Build.

InstallShield builds the release. The Output window opens across the bottom of the InstallShield user interface. This window also provides information about your project during validation and project conversion. The following tabs appear in the Output window:

Table 3-1: Output Window Tabs

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build</td>
<td>Stores distribution output information and displays build output; a link to the output file saved as a text file is included.</td>
</tr>
<tr>
<td>Validate</td>
<td>Displays validation results.</td>
</tr>
<tr>
<td>Results</td>
<td>Displays project conversion information.</td>
</tr>
<tr>
<td>Compile</td>
<td>Displays InstallScript information when you click Compile on the Build menu.</td>
</tr>
<tr>
<td>Tasks</td>
<td>Provides descriptions of error and warning codes when you build, compile, or validate your project; each error or warning code is a link to a Knowledge Base article.</td>
</tr>
</tbody>
</table>

Tip: You can right-click a code for an error or warning and select Direct Editor. The affected area associated with the error or warning will be highlighted in the Direct Editor.
Figure 3-10: Releases View and Output Window

If you have not configured InstallShield to perform validation at build time, the Build tab includes a “Windows validation is disabled” message to alert you that validation has not been performed. More details about validation are discussed in Chapter 4 of this evaluator guide.

Tip: To learn more about building releases, including details about how to use the Release Wizard, the command line, and the Standalone Build for building, see the “Building, Testing, and Deploying Installations” section in the InstallShield Help Library.
Part J: Running the Installation

For Part J, you will run the Othello installation on a Windows Vista system.

To install Othello:

1. On the toolbar, click the **Open Release Folder** button. InstallShield opens the folder that contains the built installation.

2. Copy all of the files from the installation to the Windows Vista system or to a network folder that can be accessed from the Windows Vista system.

3. Double-click the `Setup.exe` file of the installation.

The Othello installation launches.

**Tip:** InstallShield enables you to test the end-user dialogs from the system that has InstallShield without copying any files to the target system: On the toolbar, click the Test button.
Walkthrough of Additional InstallShield Features

This section of the InstallShield 2008 Evaluator Guide walks you through some of the additional InstallShield features that are not included in the tutorial in Chapter 3. You can use the Othello project that you created in the Chapter 3 tutorial as you learn about these additional features.

Part A: Digitally Signing Files

InstallShield 2008 lets you digitally sign any files—including your product’s executable files—in your project at build time. This is helpful, since all executable files (including .exe, .dll, .ocx, .sys, .cpl, .drv, and .scr files) in an installation must be digitally signed for the Certified for Windows Vista program.

In addition, InstallShield 2008 enables you to use a personal information exchange file (.pfx) for digital signatures. If you specify a .pfx file for signing, InstallShield uses SignTool.exe to sign your files. If you specify an .spc file and a .pvk file, InstallShield uses Signcode.exe to sign your files. Using a .pfx file is often the preferred method, since it is more likely to work in many different environments (such as locked build machines). If you specify the digital signature password in InstallShield, you will never see a password prompt if you are using a .pfx file. However, if you are using .spc and .pvk files, a password prompt may be displayed.

The new Signing tab in the Releases view is where you specify the digital signature information—including the digital signature files granted to you by a certification authority—that InstallShield should use to sign your files. The Signing tab is also where you specify which files in your installation should be digitally signed. You can also use the Release Wizard to specify all of the digital signature information.
Task  To configure digital signing for your release and its files:

1. In the View List under Media, click Releases.
2. In the Releases explorer, click the release that you want to sign.
3. Click the Signing tab.

Figure 4-1: Signing Tab in the Releases View

4. Configure the following settings as appropriate:
   - Certificate URL
   - Digital certificate file (SPC or PFX)
   - Private file key (PVK)—Note that if you specify a .pfx file, you do not also need to specify a .pvk file.
   - Certificate password
5. Select the appropriate check boxes to indicate which files in your release you want InstallShield to sign.

If you select the Sign files in package check box, you must configure which files or file patterns (such as all *.exe files) should be signed. You can also specify which files and file patterns should not be signed.

Note that the files and file patterns that should not be signed override any files and file patterns that should be signed. For example, if you specify *.exe in the Include patterns and files box and in the Exclude patterns and files box, InstallShield does not sign any .exe files.

Tip: For detailed information about any of the settings on the Signing tab, see “Signing Tab” in the InstallShield Help Library.

At build time, InstallShield signs the files as specified on the Signing tab. If the release is for an installation that includes merge modules, note that the files are signed before the merge module is merged.
Part B: Performing Validation

Validating a project involves applying a set of internal consistency evaluator (ICE) rules to your installation or merge module package. The ICEs are designed to ensure that the package contains a valid database that performs its actions correctly. If a package or merge module fails one or more validation rules, InstallShield reports the specific rules that were violated and offers additional information to help you troubleshoot the problem.

The InstallShield Best Practice Suite, the Certified for Windows Vista Validation Suite, and the Certified for Windows Vista Merge Module Validation Suite are three types of validation that are available in InstallShield.

InstallShield Best Practice Suite

InstallShield 2008 Premier Edition includes a new set of ICE validators called the InstallShield Best Practice Suite. The InstallShield Best Practice (ISBP) validators in this suite alert you if your installation violates best-practice guidelines.

Certified for Windows Vista Validation Suites

Microsoft established a list of requirements that a product and its installation must fulfill in order to become certified for Windows Vista. The requirements outline criteria that help make a product more compatible, reliable, and secure when running on Windows Vista systems.

Products that meet the Certified for Windows Vista program requirements can carry the Certified for Windows Vista logo.

Certified for Windows Vista Validation Suites in InstallShield

Validating your installation package or merge module may help you identify whether your product meets installation requirements for Microsoft’s Certified for Windows Vista program.

Therefore, if you are interested in being able to use the Certified for Windows Vista logo, it is recommended that you use the Certified for Windows Vista Validation Suite to validate your installation package. If you create a merge module in InstallShield, you can use the Certified for Windows Vista Merge Module Validation Suite to validate your merge module.

Qualifying for the Certified for Windows Vista Program

Products must be independently tested by a Microsoft-approved testing authority before they can be certified. To learn how to qualify for the Certified for Windows Vista program, visit http://www.InnovateOnWindowsVista.com. This Web site has a document that lists all of the technical requirements for the Certified for Windows Vista program. It also has a document that contains all of the test cases that a product must pass.

Methods for Performing Validation

InstallShield offers two methods for performing validation:

- You can run validation on demand for the release that was last built in the project.
- You can configure InstallShield to validate the installation package every time that you build a release. This behavior is disabled by default.
Specifying Whether Validation Should Be Performed at Build Time

InstallShield 2008 enables you to specify whether installation packages and merge modules should be validated each time that a release is successfully built. InstallShield 2008 also lets you specify the validation suite that should be used for validation.

To configure validation settings:

1. On the Tools menu, click Options. The Options dialog box opens.
2. Click the Validation tab.
3. Select the check boxes for the types of validation that you would like InstallShield to perform at build time. If you prefer to validate on demand, and not at build time, clear the check boxes.
4. If you select the Perform validation using check box or the Perform Merge Module validation using check box, select the validation suite that should be used.

Specifying Which ICEs, ISICEs, and ISBPs Should Be Run During Validation

InstallShield 2008 lets you customize the list of ICEs, ISICEs, and ISBPs that should be used for installation package validation and merge module validation.

To specify which ICEs, ISICEs, and ISBPs that should be run during validation:

1. On the Tools menu, click Options. The Options dialog box opens.
2. Click the Validation tab.
3. Click **Customize**. The **Customize Validation Settings** dialog box opens.

4. In the **Select CUB file to customize** list, click the suite that you want to customize.

5. In the list of ICEs and ISBPs, select the check box for each of the ICEs and ISBPs that should be used to evaluate the installation package or merge module. Clear the check box for each one that should not be used for the validation.

![Figure 4-3: Customize Validation Settings Dialog Box](image)

If you customize the list of ICEs, ISICEs, and ISBPs for a validation suite, anytime that validation is performed—whether it occurs at build time or on demand—only the selected ICEs, ISICEs, and ISBPs are used.

**Validating an Installation Package or Merge Module on Demand**

InstallShield 2008 enables you to validate an installation package or merge module separately from the build process. Doing so is especially helpful if you configured InstallShield so that validation is not performed as part of every successful build, but you do need to run it at some point to test your product for Windows logo compliance or for InstallShield Best Practices.

**Task**

To validate your release on demand:

1. Complete a successful build of a release.

2. On the **Build** menu, point to **Validate**, and then click the type of validation that you want to perform.

InstallShield validates the release.
Viewing Validation Results

InstallShield displays the results of the validation scan on the Validations tab of the Output window. In addition, InstallShield saves the results to a text (.txt) file in a `ValidationFiles` folder within the release folder. You can view this file either by navigating to your build directory, or by navigating to the Releases view and selecting the Validations folder under your release.

**Validation Messages**

Validation messages are broken down into three categories:

- **Errors**—Describe problems with your database, such as having duplicate component GUIDs.
- **Warnings**—Describe problems in your database that may occur in certain circumstances.
- **Failures**—Occur when your database has severe enough problems that the validation tool might not be able to run.

If the scan results for your project include validation messages, the messages and associated codes are also listed on the Tasks tab of the Output window.

**Note:** If you want to see validation warnings that apply to your installation or merge module, you must perform validation on demand; this type of validation message is not available if you perform validation at build time. InstallShield reports the other types of validation messages—errors and failures—during both validation methods.

**Tip:** You can click a validation code on the Tasks tab of the Output window to see the corresponding Knowledge Base article or HelpNet topic.

In addition, you can click a validation message on the Tasks tab to jump to the area of the Direct Editor that corresponds to the validation message. This functionality is available for ICEs, ISICEs, and ISBPs.

![Figure 4-4: Validation Errors](image-url)
Part C: Using the UAC Shield Icon on Buttons

In the Dialog Editor of InstallShield 2008 Basic MSI projects, a new Show UAC Shield Icon property is available for all button controls. If you select True for this property, the UAC icon is displayed on the button when end users run the installation on Windows Vista systems. If you are using InstallShield on a Windows Vista system, you can see the shield icon on the button in the Dialog Editor as it will be displayed at run time. The shield icon signals to end users that elevated privileges may be required.

For any new Basic MSI projects that you create in InstallShield 2008, the Show UAC Shield Icon property is set to True for the Install button on the ReadyToInstall dialog. If you upgrade a Basic MSI project that was created with InstallShield 12 or earlier to InstallShield 2008, the default value for the Install button’s Show UAC Shield Icon property is False. You can override the value for this button, or for any other buttons, as required.

Task: To show or hide the UAC shield icon on a dialog button:

1. In the View List under User Interface, click Dialogs.
2. In the Dialogs explorer, expand the All Dialogs folder, and then expand the dialog item for a dialog that has the button that you want to edit.
3. Click a language under the dialog item that you expanded. The Dialog Editor in the center pane shows the dialog in the selected language.
4. Click the button that you want to edit. InstallShield displays the button’s properties in the right pane.

Figure 4-5: Properties for a Dialog Button
5. For the **Show UAC Shield Icon** property, select the appropriate value:

- To add the UAC shield icon to the button for Windows Vista systems, select **True**.
- To omit the UAC shield icon from the button, select **False**.

---

**Part D: Minimizing Reboots Through the Restart Manager Infrastructure**

Restarting the system after an installation is inconvenient for end users. One of the Windows Vista Quality Program requirements is that all installations must contain an option that enables end users to automatically close applications and attempt to restart them after the installation is complete.

To support this requirement, all Basic MSI projects include the MsiRMFilesInUse dialog by default. An installation displays the MsiRMFilesInUse dialog on a Windows Vista system if one or more files that need to be updated are currently in use during the installation. The dialog contains two options to allow end users to specify how to proceed:

- End users can choose to have the installation close the applications that are using those files and then attempt to restart the applications after the installation is complete.
- End users can avoid closing the applications. A reboot will be required at the end of the installation.

![MsiRMFilesInUse Dialog](image)

**Figure 4-6: MsiRMFilesInUse Dialog**

For the best end-user experience, your application should be instrumented to use the Restart Manager API; doing so allows the Restart Manager to effectively pause and resume your application exactly where the end user left it. For detailed information, see “About Restart Manager” ([http://msdn.microsoft.com/library/default.asp?url=/library/en-us/rstmgr/rstmgr/about_restart_manager.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/rstmgr/rstmgr/about_restart_manager.asp)) and the other Restart Manager documentation on the MSDN Web site.

**Tip:** In InstallShield 2008, a new **SdRMFilesInUse** dialog and a new **OnRMFilesInUse** event handler are available in InstallScript MSI projects. For more details, see “SdRMFilesInUse” and “OnRMFilesInUse” in the InstallShield Help Library.
Part E: Adding and Sequencing Custom Actions

For InstallShield 2008, the Custom Actions view and the Sequences view have been combined into a more robust view called the Custom Actions and Sequences view. The combined view supports drag-and-drop editing and copying:

- To sequence a new custom action, drag it from the Custom Actions explorer to the appropriate location in a sequence under the Sequences explorer.
- To move a dialog, standard action, or custom action to a different point in a sequence (or from one sequence to another), drag it from the old location to the new location.
- To copy a custom action from one sequence to another, press and hold CTRL while dragging the custom action from one sequence to another sequence.

The Custom Actions and Sequences view is available in Basic MSI, InstallScript MSI, MSI Database, Transform, and Web projects.

![Custom Actions and Sequences View](image)

**Figure 4-7:** Custom Actions and Sequences View

**Tip:** For more information about this view, or for instructions on configuring custom actions or sequences, see the “Customizing Installation Behavior” section of the InstallShield Help Library.
Part F: Adding .NET Framework Redistributables to Projects

If your product requires that the .NET Framework be installed on the target system, you can add the .NET Framework redistributable to your project. If the target system does not have the .NET Framework, it is installed during your installation.

You can also include redistributables for .NET Framework language packs in your project. The language packs contain translated text, such as error messages, for languages other than English.

New .NET Redistributables in InstallShield 2008

InstallShield 2008 now includes many new .NET-related setup prerequisites that you can add to Basic MSI and InstallScript MSI projects:

- .NET Framework 2.0 (x64)
- .NET Framework 2.0 (x64) Language Packs
- .NET Framework 2.0 (IA64)
- .NET Framework 2.0 (IA64) Language Packs
- .NET Framework 3.0 (x64)

In addition, an updated Microsoft .NET object is available in InstallShield 2008 for InstallScript projects. This object includes support for versions 1.0 (SP3), 1.1 (SP1), 2.0, and 3.0 of the .NET Framework, including 32-bit, 64-bit x64, and 64-bit Itanium versions. The object also includes all supported language packs, and the latest service packs of 1.0 and 1.1.

The object launches and completes the .NET Framework installation as the feature containing the .NET object installs. This allows the .NET Framework to be available as early as possible, in case it is needed to install or configure files that are subsequently installed during the installation.

Including .NET Support in Projects

The method for adding the .NET Framework and .NET Framework language packs to your project depends on the project type that you are using, as well as the version of .NET Framework that your application requires.

Basic MSI and InstallScript MSI Projects

If you want to include .NET support in a Basic MSI project or an InstallScript MSI project, do one of the following:

- **For .NET Framework 3.0 or 64-bit .NET Framework 2.0 redistributables**—Add the appropriate Microsoft .NET Framework setup prerequisite. You can add this to your project through the Redistributables view.

- **For .NET Framework 2.0, 1.1, or 1.0 redistributables (32 bit)**—Configure the .NET settings for the release in the Releases view. As an alternative, you can select the appropriate options through the Release Wizard.
To test whether the .NET Framework is already installed on the target system, you can use the built-in `MsiNetAssemblySupport` property. It is set to the version of a particular .NET DLL (`fusion.dll`) if the .NET Framework is installed, and it is not set if the .NET Framework is not installed.

**InstallScript Projects**

If you want to include one or more versions of the .NET Framework in an InstallScript project, use the Objects view to add the Microsoft .NET Framework object to your installation. This object also enables you to add one or more language packs to an InstallScript project.

To determine whether a particular version of the .NET Framework or a language pack is installed, use the `Is` function and pass the `DOTNETFRAMEWORKINSTALLED` predefined constant.

**Obtaining Setup Prerequisites and Objects**

Note that some of the setup prerequisites and objects are not installed with InstallShield. You may need to download them.

If you have an Internet connection, you can use the Check for Updates feature in InstallShield to obtain the latest InstallShield merge modules, objects, and setup prerequisites for the version of InstallShield that you are using.

---

**Task**

**To check for updates:**

On the **Tools** menu, click **Check for Updates**.

InstallShield launches FLEXnet Connect, which checks for updates. When an update is available, you can do the following:

- View the updates for your system.
- View descriptions about the updates.
- Download and install the updates that you select.
Additional Information

InstallShield, the original and most popular installation tool, combines power and flexibility with ease-of-use. First-time developers find it easy to create installations in minutes, while power users appreciate its sophisticated installation designer interface for developing their most complex installations. InstallShield 2008 is fully integrated with Visual Studio, enabling development of installations and applications within the same familiar Visual Studio interface. InstallShield 2008 continues InstallShield’s tradition as the industry standard for installation authoring by supporting the latest technology platforms.

Additional Resources

Several resources are available to help you learn about InstallShield 2008.

- **InstallShield Help Library**—The Help Library contains information on InstallShield 2008 and is easily accessible from the Help menu in InstallShield 2008. The Help Library is also available online at http://helpnet.macrovision.com.

- **Windows Installer Help Library**—InstallShield 2008 includes the full Microsoft Windows Installer Help Library, which provides technical information on the Windows Installer service.

- **Knowledge Base**—For answers to many commonly asked questions and new information about InstallShield 2008, visit the Knowledge Base at http://support.installshield.com/kb.

- **InstallShield Communities**—To access the free online InstallShield community forum that connects you to other InstallShield users and enables you to post questions and search for answers, visit http://community.macrovision.com.

- **Support**—A constantly updated support portal filled with answers, news, tips, and other resources for InstallShield 2008 users is available at http://www.macrovision.com/support/index.shtml.

- **Newsletters**—Subscribe to one or more Macrovision email newsletters, or view the archives of previously issued newsletters. The DevLetter newsletter offers announcements about InstallShield product and service updates, links to articles that provide product tips and tricks, advice from industry experts, links to white papers and webinars, announcements about industry events and
System Requirements

Following are the minimum requirements for systems that run InstallShield (the authoring environment), as well as for target systems that run the installations created with InstallShield (the run-time environment).

For Systems Running InstallShield

Processor
Pentium III-class PC (500 MHz or higher recommended)

RAM
256 MB of RAM (512 MB preferred)

Hard Disk
500 MB free space

Display
Designed for XGA resolution at 1024 × 768 or higher

Operating System
Windows 2000 SP3 or later
Windows XP
Windows Server 2003
Windows Vista

Browser
Microsoft Internet Explorer 5.01 (IE 5.5 or later recommended)

Privileges
Administrative privileges on the system
**Mouse**
Microsoft IntelliMouse or other compatible pointing device

**For Target Systems (Desktop Computers)**
Target systems must meet the following minimum operating system requirement:
- Windows 2000
- Windows XP
- Windows Server 2003
- Windows Vista

**For Target Systems (Mobile Devices)**
InstallShield includes support for straight-to-device installations that do not use Microsoft ActiveSync or another desktop component.
InstallShield also includes support for adding mobile device installations to desktop installations.

**Windows Mobile Device Requirements**
InstallShield supports many Windows Mobile platforms and processors. The Windows Mobile platforms are:
- Windows Embedded CE 6.x
- Windows Mobile 5.0 for Pocket PC
- Windows Mobile 5.0 for Smartphone
- Windows CE .NET 5.0
- Windows CE .NET 4.x
- Pocket PC 2003
- Pocket PC 2002
- Pocket PC
- Palm-size PC 2.11
- Palm-size PC 2.01
- Handheld PC 2000
- Handheld PC Pro
- Handheld PC 2.0
- Smartphone 2003
- Smartphone 2002
Chapter 5: Additional Information
System Requirements

Note that if a platform is not included in the list, it does not mean InstallShield does not support it. It simply means that you cannot set conditions for that specific platform.

InstallShield includes support for the following Windows Mobile processors:

- ARM920
- ARM820
- ARM720
- Common Executable Format
- Hitachi SH4
- Hitachi SH3E
- Hitachi SH3
- i686
- i586
- i486
- MIPS R4000
- MIPS R3000
- MIPS R2000
- SHx SH4
- SHx SH3
- StrongARM-XScale

Palm OS Device Requirements
InstallShield supports Palm OS 3.5 and later.

Desktop Requirements for Windows Mobile Device Installations
Requirements for the desktop computers that are used to install applications on Windows Mobile devices are:

- Microsoft ActiveSync 3.x or later (ActiveSync 4.x is required for Windows Mobile 5.x devices)
- Administrative privileges

Desktop Requirements for Palm OS Device Installations
Palm HotSync is required for the desktop computers that are used to install applications on Palm OS devices.
Customer Feedback

Your feedback is essential to helping us deliver features that meet your needs. Many of the features for InstallShield 2008 come directly from the feedback and suggestions that customers have shared with our engineering and support teams. If there is anything we can do better, or if you need assistance, please do not hesitate to contact us at http://www.macrovision.com/feedback/index.shtml.
Chapter 5: Additional Information
Customer Feedback
Symbols

.NET Compact Framework 2.0 SP1 1
.NET Framework 41
.NET Framework language packs 41
.pfx file 31
.pvk file 31
.spc file 31

C

COM servers 1
component conditions for Windows Vista 16
contacting Macrovision 8
Create MSI Logs setting 20
creating
   Basic MSI project 15
   log files for Windows Installer 4.0 installations 20
   new project 14
   product configuration and release 22
   custom actions 40
Customer Experience Improvement Program 8
Customize Validation Settings dialog box 35
customizing
   Windows Installer 4.0 log files 22

D

Database Import Wizard 2
dialog button
   adding or removing shield icon 38
dialog themes 19
DIFx 2.01 support 1
digital signature 31
digital signing 31
DirectX 9.0c redistributable 1
downloads for InstallShield 8

Numerics

64-bit registration of COM servers 1
64-bit self-registration 1

A

adding files 17
Administrator option for Required Execution Level setting 26

B

Basic MSI project type 12
building a release 27
   and performing validation 35
Index

E
enhancements 2

F
feature conditions for Windows Vista 16
Files and Folders view 17
  enhancements for 18
FLEXnet Connect 1

G
General Information view
  Create MSI Logs setting 20
  Require Administrative Privileges setting 24

H
help topics 8
help, getting 8
Highest available option for Required Execution Level setting 26

I
ICEs 34
  specifying which should be run 35
IIS 7 1
Install button 38
Installation Designer 11
installing InstallShield 7
InstallScript 12
InstallShield Best Practice Suite 34
InstallShield Community 8
InstallShield Help Library 8
InstallShield ICEs 34
  specifying which should be run 35
interface of InstallShield 7
  Output window 27
  views 11
internal consistency evaluators (ICEs) 34
Internet Information Services 7 1
Invoker option for Required Execution Level setting 26
ISBPs 34
  specifying which should be run 34
ISBuild.exe 27
ISCmdBld.exe 27
ISICEs 34
  specifying which should be run 35

K
known issues in InstallShield 8

L
launching an application from an installation 26
logging Windows Installer 4 installations 20

M
Macrovision, contacting 8
merge module validation 35
migration information 4
MSDE 2000 SP4 1
MsiLogFileLocation property 20
MsiLogging property 20
MsiRMFilesInUse dialog 39
MSXML 6 SP1 1

N
new features 1
New Project dialog box 14

O
One-Click Install installation 2
OnRMFilesInUse 39
Output window 27

P
permissions
  elevated for installation 26
UAC 24
privileges
  elevated for installation 26
  UAC 24
product configuration, creating 22
Project Assistant 10
projects
  list of recently opened 8
  types of 12

R
ReadyToInstall dialog 38
reboots
  minimizing 39
redistributable
  none for Windows Installer 4.0 25
release notes on InstallShield 8
Release Wizard 27
  Digital Signature panel 31
release, creating 22
Releases view 22
  building a release from the 27
  digital signature settings 31
  enhancements for 22
  Required Execution Level setting 25
  Setup Launcher setting 25
Restart Manager 39
restarts
  minimizing 39
RSS feeds of KB articles 8
running an installation 29
run-time requirements
  for desktop computers 45
  for mobile devices 45

S
SdRMFilesInUse 39
sequences 40
setup launcher, specifying required execution level for 25
Setup Prerequisite Editor 1
shield icon on dialog buttons 38
Show the Windows Installer log check box 20
Show UAC Shield Icon property 38
signing 31
  files 31
Setup.exe 31
  Windows Installer package 31
Signing tab 31
SQL Server 2005 Express SP1 1
SQLLogin dialog 2
SSL 1
Standalone Build 27
Start Page 8
system requirements 44
  for systems running InstallShield 44
  for target systems (desktop computers) 45
  for target systems (mobile devices) 45

T
targeting Windows Vista systems 15
testing an installation 29
themes for dialogs 19

U
UAC security 24
UAC shield icon 38
upgrading projects to the latest version of InstallShield 4
User Account Control (UAC) security 24

V
validating
  on demand 36
  results of 34, 37
validation 34
  performing at build time 35
  performing on demand 36
  specifying when to perform 35
  viewing results 34, 37
Validation tab on Options dialog box 35
views in InstallShield 11
Visual C++ 8.0 SP1 merge modules 1

W
webinars on InstallShield 8
Welcome Assistant 13
Windows Embedded CE 6.x 1
Index

Windows Installer 4.0
  adding a Show the Windows Installer log check box for 20
  and Windows Vista support 25
  customizing log files for 22
Windows Mobile 5.0 for Smartphone 1
Windows Server "Longhorn" 1
Windows validation is disabled message 28
Windows Vista
  and Windows Installer 4.0 25
  creating feature and component conditions for 16
creating installations for 15
logo compliance 34
minimizing reboots on 39
MsiRMFilesInUse dialog 39
support for 1

XML file changes 2