



Wise Installation Studio, Wise Installation Express Getting Started Guide

Notice

Wise Installation Studio and Wise Installation Express 7.0

© 1994-2006 Wise Solutions, Inc. All Rights Reserved.

This documentation and the accompanying software are copyrighted materials. Making unauthorized copies is prohibited by law. No part of the software or documentation may be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any human or computer language without prior written permission of Wise Solutions, Inc. Wise Solutions, Inc. asserts its "Moral Right" to be identified as the author of this work, in all jurisdictions which recognize the "Moral Right."

Unless otherwise provided by written agreement with Wise Solutions, Inc., this publication, and the software sold with this publication, are provided "as is" without warranty of any kind either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. The entire risk arising out of the use or performance of this publication and software remains with you. In no event will Wise Solutions, Inc., or any of its suppliers, be liable for any lost profits, lost savings, direct, incidental or indirect damages or other economic or consequential damages, even if Wise Solutions, Inc., or its suppliers, have been advised of the possibility of such damages. Wise Solutions, Inc. reserves the right to modify this document at any time without obligation to notify anyone. In no event shall Wise Solutions, Inc.'s or its suppliers' liability under this agreement exceed the sum of any amounts paid hereunder by the customer to Wise or the supplier.

Wise Solutions, Inc. owns a number of trademarks and service marks (the "Marks"). These Marks are extremely valuable to Wise Solutions, Inc. and shall not be used by you, or any other person, without Wise Solutions, Inc.'s express written permission. The Marks include, but are not necessarily limited to the following: Application Isolation Wizard™; ApplicationWatch™; ConflictManager®; ExpressBuild™; Installation Development Life Cycle™; InstallBuilder®; InstallMaker®; InstallManager®; InstallTailor™; MSI Debugger™; MSI Script™; PackageManager™; Preflight Deployment™; SetupCapture®; SmartMonitor™; SmartPatch™; Software Distribution Made Easy™; Software Installations Made Easy®; Unwise™; Virtual Capture™; Visual MSIDiff™; WebDeploy®; Wise Installation System®; Wise MSI Editor™; Wise Package Studio®; Wise Software Repository™; Wise Solutions®; WiseScript™; WiseScript Express™; WiseUpdate®; WiseUser®; and the Wise Solutions® logo.

In addition to Wise Solutions, Inc.'s Marks, some Wise Products may include Trademarks or Service Marks owned by other corporations. These other Marks include, but are not necessarily limited to Microsoft® Windows® and Microsoft® Visual Studio® .NET, which are registered Trademarks of Microsoft Corporation.

You shall not use any of the Trademarks or Service Marks of Wise Solutions, Inc., Microsoft Corporation, or any other entity, without the express written permission of such Trademark or Service Mark owner.

Wise Solutions, Inc., a wholly owned subsidiary of Altiris, Inc.
47911 Halyard Drive; Plymouth, Michigan 48170 USA
Phone: +1 734 456 2100 • Fax: +1 734 456 2456 • www.wise.com

Contents

Preface	4
Documentation Roadmap	4
Getting Help and Product Support	5
Chapter 1: Installing Wise Installation Studio and Wise Installation Express	7
Introduction	7
System Requirements	8
Connecting to a Wise Software Repository	9
Installing an Upgrade	10
Installing Wise Installation Studio or Wise Installation Express	10
Installing the Windows Installer Editor Language Pack	12
Starting the Software	12
Converting an Evaluation Version to a Production Version	12
Registering Your Software	13
Chapter 2: Basic Tutorial: Creating a Windows Installer Installation	14
Create a New Installation Project	14
Open a Project and Set Options	15
Enter Product Details and General Information	16
Create and Organize Features	17
Assemble the Installation	18
Add Files to the Installation	18
Add Registry Keys	20
Add a Shortcut	21
Create a File Association	21
Customize the Installation's Behavior on the Destination Computer	22
Set System Requirements	22
Select Dialogs That Appear to End Users	22
Configure Installation Types	23
Define Releases	23
Create Releases	24
Customize the Demo Release Properties	25
Compile and Test the Installation	26
Chapter 3: Advanced Tutorial: Using MSI Script in a Windows Installer Installation ..	28
Call a .DLL With a Custom Action	28
Set a Directory With a Custom Action	32
Call an .EXE With a Custom Action	36
Performing Advanced Tasks	39
Chapter 4: Tutorial: Creating a Windows Installer Installation in Visual Studio Integrated Editor	40
Open the C# QuickFacts Solution and Create an Installation Project	40
Open the Installation Project and Set Options	42
Enter Product Details and General Information	43
Add Files to the Installation	43
Index	46

Preface

This product offers you two choices for authoring installations: Wise Installation Studio or Wise Installation Express. Both products build on established Wise technologies.

Wise Installation Studio is an advanced installation authoring tool that supports both Windows Installer (.MSI) and WiseScript installations.

Wise Installation Express is a basic installation authoring tool that supports Windows Installer installations.

Documentation Roadmap

This documentation assumes that you are proficient in the use of the Windows operating system. If you need help using the operating system, consult its user documentation.

Use the following sources of information to learn this product.

Online Help

The online help contains detailed technical information and step-by-step instructions for performing common tasks.

Access help in the following ways:

- To display context-sensitive help for the active page or dialog, press F1.
- To select a help topic from a table of contents, index, or search, select Help menu > Help Topics.

Each tool has its own Help menu and its own online help.

If you need help and cannot find the answer in the documentation, see [Getting Help and Product Support](#).

Reference Manual

All the material in the online help is also available in a .PDF-format reference manual, which you can access by selecting Help menu > Reference Manual.

Each tool has its own Help menu and its own reference manual.

Getting Started Guide

The *Getting Started Guide* contains system requirements, installation instructions, and a tutorial. You can access a .PDF version of the *Getting Started Guide* from the Windows Start menu.

Windows Installer SDK Help (Windows Installer Editor only)

You can get technical details about Windows Installer from its own help system, which is written by Microsoft for a developer audience. In Wise for Windows Installer, select Help menu > Windows Installer SDK Help.

Version 3.1 of the Windows Installer SDK Help is provided. If you have obtained a later version, links from the Wise product documentation to the Windows Installer SDK Help might not work.

To access the Windows Installer SDK Help in Visual Studio, select Help menu > Wise Help > Windows Installer SDK Help. Windows Installer SDK help topics are also available within the Visual Studio .NET help collection.

Release Notes

A release notes document, in .HTM format, covers new features, enhancements, bug fixes, and known issues for the current version of this product. Access the release notes in the following ways:

- Browse the product CD.
- Select it from the Altiris program group on the Windows Start menu.
- If you are a registered customer, visit <http://support.wise.com> to enter the Support Center, and then click the Downloads link.

Getting Help and Product Support

If you need help and cannot find the answer in the documentation (see [Documentation Roadmap](#)) you can use one of the many support resources available to you as a registered customer.

Use the Technical Support Center

Registered customers can use the Technical Support Center, located at <http://support.wise.com>, to submit online support requests, register products, manage customer information, download updates, or search the Knowledgebase. The Knowledgebase contains how-to procedures, answers to common support questions, and workarounds.

Visit Our Newsgroups

Visit <http://forums.altiris.com/>. Newsgroup postings by your peers contain answers, tips, analysis, and other comments. Contribute your own expertise to help others.

Ask Our Support Team

If you can't find an answer in our online resources, you can obtain support by phone or online at <http://support.wise.com>. Flexible payment options are available to meet your support needs. For additional details about our support services, visit www.wise.com/supportoptions.asp or call 1-734-456-2600.

Before you contact technical support, obtain the following:

- Serial number and product version, which you can find by selecting Help menu > About.
- Operating system version and service pack version if applicable.
- A description of what you do before the problem occurs.
- The text of any error messages that appear.
- Your name, company name, and how to contact you.

- Contract number or payment information, if applicable.

Take Advantage of Our Consulting and Training Services

When you have a challenging repackaging or installation problem, our consultants can help with script writing, repackaging, installation development, and other solutions that are fully customizable to fit your project and budget. Visit www.wise.com/consulting.asp for details.

To upgrade your installation and packaging skills, consider training. Our certified instructors draw from practical experience to provide relevant course content. Visit www.wise.com/training.asp for course descriptions and schedules.

Contact Our Sales Department

Contact our Sales department to purchase additional products, upgrades, support services, or consulting and training services.

U.S.: +1 800 554 8565

EMEA: +8000 ALTIRIS (2584747) or +49 211 68 773 222

Other International: +1 734 456 2100

E-mail: wisesales@altiris.com

Web Site: www.wise.com/ordercentermain.asp

Chapter 1

Installing Wise Installation Studio and Wise Installation Express

Refer to the information in this section to get Wise Installation Studio or Wise Installation Express installed and running.

Topics include:

- [Introduction](#)
- [System Requirements](#)
- [Connecting to a Wise Software Repository](#)
- [Installing an Upgrade](#)
- [Installing Wise Installation Studio or Wise Installation Express](#)
- [Installing the Windows Installer Editor Language Pack](#)
- [Starting the Software](#)
- [Converting an Evaluation Version to a Production Version](#)
- [Registering Your Software](#)

Introduction

This product offers you two choices for authoring installations: Wise Installation Studio or Wise Installation Express. Both products build on established Wise technologies.

In effort to simplify our product lines, we have discontinued the Wise for Windows Installer and Wise Installation System product lines. The technologies that were in those products are now included in Wise Installation Studio and Wise Installation Express.

Wise Installation Studio

Wise Installation Studio is an advanced installation authoring tool that supports both Windows Installer (.MSI) and WiseScript installations. Wise Installation Studio is a comprehensive collection of tools and technologies for creating software installations for Windows devices:

- **Windows Installer Editor**
This tool is comparable to Wise for Windows Installer - Professional and Enterprise Editions. Use it to create and edit Windows Installer installation packages.
- **WiseScript Package Editor**
This tool is comparable to Wise Installation System - all editions. Use it to create and edit installation packages based on the WiseScript scripting language.
- **Wise Virtual Package Editor**
Use this tool to create and edit a virtual software layer, a virtual software project file, or a virtual software archive file.

When you choose to install Virtual Package Editor, the Altiris Software Virtualization Agent (SVS Agent) is installed also. However, if you install Wise Installation Studio on a 64-bit computer, the SVS agent is not installed and any functionality that requires this agent is disabled.

- Wise Mobile Device Package Editor - Use this tool to create a project file in .INF format and compile it to one or more .CAB files that install a mobile device application.

Wise Installation Express

Wise Installation Express is a basic installation authoring tool that supports Windows Installer installations. Wise Installation Express consists of the following tools:

- Windows Installer Editor
This tool is comparable to Wise for Windows Installer - Standard and Professional Editions. Use it to create and edit Windows Installer installation packages.
- WiseScript Editor (formerly named WiseScript Express)
This tool contains the same scripting interface as WiseScript Package Editor, but it does not contain the Installation Expert interface or other installation development features. Use it to create WiseScripts that are called by a Windows Installer custom action. You access WiseScript Editor from a WiseScript custom action in Windows Installer Editor.

System Requirements

Note

Wise Installation Studio and Wise Installation Express cannot be installed on any computer that contains any edition or version of Wise for Windows Installer, Wise for Visual Studio .NET, or Wise Package Studio.

The Visual Studio integrated editor requires Microsoft® Visual Studio® .NET (any edition). For Microsoft's requirements for installing Visual Studio .NET, consult the documentation that you received with Visual Studio .NET.

Requirement	Value
Computer/processor	PC with a Pentium II-class processor, 450 MHz
Operating system	Windows 2000 or later (SP4 recommended) Windows XP Professional (SP2 recommended) Windows Server 2003 Windows Vista *
Internet Explorer	5.5 or later or later* * Software Manager reports require 6.0 or later, or Microsoft XML Parser (MSXML) 3.0 Service Pack 7.
RAM	128 MB; 256 MB recommended

Requirement	Value
Drive	CD-ROM or DVD-ROM drive
Display	Super VGA (800 x 600) or higher-resolution monitor with 256 colors
Free space on boot drive	65 MB
Free space on installation drive	120 MB

Installations that you create with Windows Installer Editor can be run on any Windows operating systems that are supported by Windows Installer.

*** Note**

Windows Vista support in this product includes but is not limited to Altiris' product readiness statement for Windows Vista, which is based on a pre-production release (build 5600) and may be considered "forward-looking." Such forward-looking statements are based on current expectations and involve certain risks and uncertainties that could cause actual results to differ from such forward-looking statements. Such risks include potential changes to Windows Installer 4.0 or the core operating system (Microsoft Windows Vista). These risks should be taken into account when purchasing any software. We therefore cannot provide any assurance that such forward-looking statements will materialize according to their current form. However, Altiris is dedicated to supporting Microsoft Windows Vista with Wise Installation Studio and Wise Installation Express.

.NET Framework System Requirements

The .NET Framework is required for advanced .NET support in Windows Installer Editor. The .NET Framework requires Microsoft® Internet Explorer 5.01 or later and one of the operating systems supported by Wise Installation Studio and Wise Installation Express.

Connecting to a Wise Software Repository

If you have upgraded to Wise Installation Studio from the Enterprise Edition of Wise for Windows Installer, you can connect to a Wise Software Repository that has been configured for an installation of Wise Package Studio 7.0 SP1 or later.

When you connect to a Wise Software Repository:

- The default locations for Windows Installer installation resources are set to subdirectories of the share point directory that is associated with the repository. See *Installation Resources and Their Locations* on page 33.
- Repository-related features in Windows Installer Editor are enabled.

To connect to a Wise Software Repository, use the Repository Client Manager. See *Connecting to a Wise Software Repository* in the Windows Installer Editor Help.

Also see *Using a Wise Package Studio Repository* in the Windows Installer Editor Help.

Installing an Upgrade

To upgrade to Wise Installation Studio or Wise Installation Express 7.0 from an earlier version of a Wise product:

1. Uninstall the previous version.
2. Run the installation and, on the User Information dialog, enter a new serial number and the previous version serial number.

If you don't know the previous version serial number, open the product and select Help menu > About. The serial number appears on the About dialog. If the product is not installed on your computer, look for the serial number on the product box, CD case, or registration card you received with your original purchase.

Installing Wise Installation Studio or Wise Installation Express

1. Close any other applications that are running.
2. Start the installation program:
 - From a downloaded file: double-click the installation file.
 - From the product CD:
When you insert the product CD, a Welcome dialog appears. Click the button that installs Wise Installation Studio or Wise Installation Express. If the Welcome dialog does not appear, use Windows Explorer to browse to the CD, then open the file named Autoplay.exe.
3. On the installation's Welcome dialog, click Next.
4. On the License Agreement dialog, read the license terms, mark **I accept the license agreement**, and click Next. If you do not accept the license agreement, you cannot continue the installation.
5. Complete the remaining dialogs, which are described below.

User Information dialog

- **Serial Numbers**
If you are installing an evaluation version without a serial number, skip this section.
 - **Product**
Enter a production or evaluation serial number for the Standard or Professional Edition. If you are upgrading from a previous version, you should have received a new serial number for the upgrade.
 - **Previous Version**
If you are upgrading from a previous version of Wise for Windows Installer, enter the serial number for the previous version. If you don't know the previous version serial number, open Wise for Windows Installer and select Help menu > About. The serial number appears on the About dialog.
 - **Language Pack**
If you purchased the optional Language Pack for Windows Installer Editor, enter its serial number.

- **Install Evaluation Version**
Mark this to install an evaluation version without a serial number.
- **Full Name, Organization**
Enter this identifying information.

Evaluation dialog

This dialog appears only when you mark the **Install Evaluation Version** checkbox on the User Information dialog.

Mark the option for the product to install.

Select Features dialog

(Wise Installation Studio only.) Click the button to the left of each feature and select an option for installing that feature. Windows Installer Editor and WiseScript Package Editor are always installed.

Wise Installation Studio / Wise Installation Express Directory dialog

In **Destination Directory**, specify where to install the product's application files.

Ready to Install the Application dialog

Click Next to start installing.

Final dialog

When the installation finishes, a dialog indicates that the installation has been completed successfully.

On the final dialog, click Finish to exit the installation.

If you are prompted to restart your computer, do so.

Nalpeiron dialog

This dialog contains information about the Nalpeiron PRO-Tector Trialware solution, which lets you create copy-protected trial and demo versions of your Windows applications. The behavior of the checkbox on the Nalpeiron dialog depends on what you installed:

- The production version of Wise Installation Studio includes one free copy of Nalpeiron PRO-Tector. Mark the checkbox to be directed to a Web page where you can download a copy. To visit this Web page later, go to www.wise.com/wis/7_0/home/nalpeiron.aspx.
- When you install the evaluation version of Wise Installation Studio, mark the checkbox to be directed to a Web page where you can download an evaluation copy of Nalpeiron PRO-Tector. To visit this Web page later, go to www.wise.com/wis/7_0/home/nalpeiron_eval.aspx.
- When you install Wise Installation Express, mark the checkbox to be directed to a Web page that provides information about Nalpeiron PRO-Tector. To visit this Web page later, go to www.wise.com/wie/7_0/home/nalpeiron.aspx.

Installing the Windows Installer Editor Language Pack

The Language Pack is an add-on to Windows Installer Editor. It provides pre-translated strings for 20 additional languages on the Languages page, which lets you effortlessly ship your installations in other languages.

Options for Installing the Language Pack

- If you purchase the Language Pack at the same time you purchase Wise Installation Studio or Wise Installation Express, the Language Pack is installed with the main application. Enter the Language Pack serial number on the User Information dialog.
- If you purchase the Language Pack after you install Wise Installation Studio or Wise Installation Express, then install the Language Pack separately.
 - a. When you purchase the Language Pack, you receive a serial number.
 - b. Download the Language Pack installation file from the Wise Web site or insert the original product CD into your CD drive and navigate to the Language Pack installation file. The filename is WIS_Lang_x.msi, where x is the product version number.
 - c. Run the installation from the Language Pack installation file. The installation process is an abbreviated version of the main product's installation wizard. Enter the Language Pack serial number on the User Information dialog and step through the remaining dialogs.

Starting the Software

- **Wise Installation Studio**
Select Start menu > Programs > Altiris > Wise Installation Studio, and then select the tool to start.
- **Wise Installation Express**
Select Start menu > Programs > Altiris > Wise Installation Express > Windows Installer Editor.

Typically, the tool opens to its Home page. For instructions on using each tool, see its documentation.

Converting an Evaluation Version to a Production Version

You can convert an evaluation version of Wise Installation Studio or Wise Installation Express to a production version without uninstalling the product.

1. Purchase the product and obtain a production serial number.
2. Start any Wise Installation Studio or Wise Installation Express tool and click **Serial Number** on the Evaluation dialog.
3. Enter the production serial number and click OK.

The evaluation version is converted to a production version. When you start a tool, the Evaluation dialog no longer appears.

Registering Your Software

It is very important that you register your software. Registering qualifies you to receive technical support via newsgroups and e-mail, plus notifications about future product upgrades.

To register:

1. Connect to the Internet.
2. Do either of the following:
 - In Windows Installer Editor, go to the Home page and click the link to Product Registration. (In Wise Installation Studio, you can do this in any of the editor tools.)
 - In your browser, go to <http://support.wise.com>.
Your browser opens to the Technical Support Center page on the Wise Web site.
3. Click the link to create a new user profile, or log onto an existing user profile.
4. Follow the instructions on the Web page to complete the registration.

Chapter 2

Basic Tutorial: Creating a Windows Installer Installation

This tutorial guides you through the creation of a Windows Installer installation. A sample application, QuickFacts, which is included with Windows Installer Editor, provides a realistic practice scenario. The QuickFacts program is a simple text viewer. QuickFacts does not make system changes to your computer and is safe to install.

This tutorial takes about 45 to 60 minutes to complete. If you stop before finishing, save the installation and resume the tutorial at a later time.

If you use Visual Studio integrated editor, first follow this basic tutorial using the Wise editor. Then follow the [Tutorial: Creating a Windows Installer Installation in Visual Studio Integrated Editor](#) on page 40, which contains procedures that are specific to the Visual Studio integrated editor.

In this tutorial, you will use Windows Installer Editor's Installation Expert to:

- [Create a New Installation Project](#)
- [Assemble the Installation](#)
- [Customize the Installation's Behavior on the Destination Computer](#)
- [Define Releases](#)
- [Compile and Test the Installation](#)

Note

This tutorial does not cover details of Installation Expert Pages and dialogs. Press F1 on any page or dialog for context-sensitive documentation.

Before You Start

Install Wise Installation Express, or the Windows Installer Editor feature of Wise Installation Studio. See [Installing Wise Installation Studio or Wise Installation Express](#) on page 10.

We assume that you are following this tutorial for the first time. Follow the instructions in this tutorial exactly as presented; otherwise, your results can deviate from the information presented here.

Caution

If you have followed this tutorial previously, delete any files you might have created then. If you previously installed the QuickFacts sample application, uninstall it before proceeding with the tutorial.

Create a New Installation Project

In this section, you set options for using Windows Installer Editor. You then create a new installation, enter descriptive information about it, and structure the features in the installation.

Tasks in This Section

- Setting options.
- Creating a new installation project.
- Entering product details.
- Entering general information.
- Entering information for Add/Remove Programs.
- Setting up features.

Open a Project and Set Options

You use the Wise Options dialog to customize the behavior of Windows Installer Editor for all installations you create.

Windows Installer projects are comprised of features, and normally, you view folders and registry keys per feature. In this procedure, you open a new project and set options that let you view folders and registry keys for all features simultaneously, making it easier to see the installation as a whole.

1. Select Windows Start menu > Programs > Wise > Windows Installer Editor.
 2. Select File menu > New.
The New Installation File dialog appears.
 3. On the New Installation File dialog:
 - a. Select the Windows Application icon.
 - b. Mark **Create .WSI or .WSM project file that can be compiled into an .MSI or .MSM**. This option lets you work in a project file (.WSI), which describes and compiles the actual installation file (.MSI).
 - c. In the Target Platform section, mark **Create installation for 32-bit platform**.
 4. Click OK.
A new project opens.
 5. Select Tools menu > Options.
 6. On the Wise Options dialog, click the Installation Expert tab and mark the following options:
 - **View directories for all features on Files page**
This lets you see all directories in Installation Expert > Files page, regardless of what feature each directory was created for.
 - **View registry keys for all features on Registry page**
This lets you see a composite view of all registry keys in Installation Expert > Registry page, regardless of what feature the registry key was created for.
-
- Note**
To display context-sensitive help, press F1.
-
7. Leave the defaults for all other options.
 8. Click OK.
 9. Save the installation with the name QuickFacts.wsi.

Enter Product Details and General Information

To enter information about the installation, you use pages in the Project Definition page group in Installation Expert.

1. Click Installation Expert at the lower left of the application window.

Throughout this tutorial, you will use pages in Installation Expert. The pages are listed in the left pane; they are organized in groups that are listed in the order in which you usually will use them.

2. Under the Project Definition page group on the left, click Product Details.

The Product Details page appears, where you enter meta data that is used in the installation dialogs to identify the product to the end user who is installing it.

3. Enter this information in the Package Meta Data table:

- **Product Name**

Enter: QuickFacts

- **Manufacturer**

Enter your company name.

- **Application Type**

Make sure this is set to **Win32 (non .NET)**.

- **Installation Target**

Make sure this is set to **Windows-based desktop/server PC**.

4. Under the Project Definition page group on the left, click General Information.

The General Information page appears, where you set the information that appears when the end user right-clicks the installation file in Windows Explorer and selects Properties.

5. Enter this information on the General Information page:

- **Title**

Enter: QuickFacts

- **Author**

Enter your name.

6. Under the Project Definition page group on the left, click Add/Remove Programs.

The Add/Remove Programs page appears, where you enter information to support the capabilities of the Add/Remove Programs control panel.

7. Enter this information on the Add/Remove Programs page:

- **Display in Add/Remove Programs list**

Make sure this is marked.

- **Product Updates URL**

Enter: No support provided.

8. Save the installation.

Now that you've entered basic information about the installation, you can define the features that provide structure to the installation.

Create and Organize Features

After entering product details and general information for a new installation, you should define its structure.

To create the structure of an installation, you define:

- What features make up the application.
- How those features are presented to the end user during installation.
- What conditions must be true for portions of features to be installed.

Determine features and conditions before configuring other aspects of the installation, because all other system changes (adding files, adding registry entries, and so on) that you define later must be associated with a feature. Many pages in Installation Expert have a **Current Feature** drop-down list that lets you set options on a per-feature and per-condition basis.

The QuickFacts installation consists of 4 features. The feature named *Core*, which is always installed, contains the QuickFacts program and online help. The other features, *Birds*, *Cats*, and *Dogs*, are optional and contain fact files about those animals. Each will appear as an optional feature during the QuickFacts installation.

To create and organize features:

1. Under the Project Definition page group on the left, click Features.

The Features page appears.

A feature named Complete already exists, because every installation must contain at least one feature.

2. Double-click the Complete feature or select it and click the Details button.

The Feature Details dialog appears.

3. Complete the dialog:

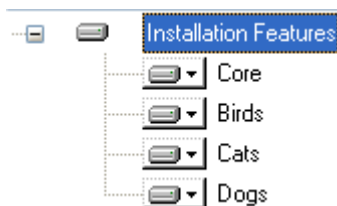
- **Name**
Override the existing name by typing *Core*, and then press Tab.
- **Title**
Enter: *Core*
- **Parent**
Leave the default.
- **Target Platform**
Leave the default.
- **Description**
Enter: QuickFacts program and help files
- **Required Feature**
Mark this.
- Leave the defaults for the rest of the fields and click OK.

4. Use the Add button to enter information for 3 additional features as shown in the following table.

Name and Title	Parent	Description	Required Feature
----------------	--------	-------------	------------------

Birds	Select <None>.	Facts about birds	Leave the default.
Cats	Select <None>.	Facts about cats	Leave the default.
Dogs	Select <None>.	Facts about dogs	Leave the default.

The Features page should display 4 features; all features should be on the same level in the tree. If not, display the feature details and change the **Parent** field until all features are on the same level.



5. Save the installation.

The installation is now set up and structured into features. In the next section, you add files and other system changes to the installation.

If you are following the Visual Studio Integrated Editor tutorial, return to [Add Files to the Installation](#) on page 43.

Assemble the Installation

In this section, you add files and other system changes to the installation. Use the pages in the Feature Details group in Installation Expert.

Tasks in This Section

- Adding files to the installation.
- Adding registry keys.
- Adding a shortcut.
- Creating a file association.

Before performing these tasks, you must have set up an installation project by following the procedures in [Create a New Installation Project](#) on page 14.

Add Files to the Installation

On the Files page, you specify the directories and files that will be installed on the destination computer. You specify files per feature; the files that you add are installed on the destination computer only if the feature is installed. Because this installation has multiple features, be sure to specify the feature for which you are adding files. You do this by selecting it from the **Current Feature** drop-down list.

To add files to the Core feature:

1. Under the Feature Details page group, select the Files page.

On the Files page, the upper 2 list boxes show the directories and files available to your computer, including your local drives, mapped drives, and local network. The lower 2 list boxes represent the directory structure and files to be installed on the destination computer.

Note

To display context-sensitive help, press F1.

2. From the **Current Feature** drop-down list, select **Core**.
3. In the lower-left list box, select the Program Files directory and click New at the lower left of the window.
4. On the Create New Folder dialog, enter QuickFacts and click OK.
A directory named QuickFacts appears under the Program Files directory.
5. In the lower-left list box, select the QuickFacts directory you created.
6. In the upper-left list box, navigate to and select the Tutorial\QuickFacts subdirectory of the Windows Installer Editor installation directory.
The files in the QuickFacts directory appear in the upper-right list box.
7. Make sure the QuickFacts directory is still selected in the lower-left list box. In the upper-right list box, select License.txt, QckFacts.exe, and ReadMe.txt and click Add File.
The 3 files are added to the QuickFacts directory in the installation.
8. In the upper-left list box, expand the QuickFacts directory and select the Help subdirectory.
9. Make sure the QuickFacts directory is still selected in the lower-left list box, and click Add Contents.
10. On the Add Contents dialog, leave the defaults and click OK.
The Help directory and its contents are added to the installation.
11. In the lower-left list box, expand the Windows directory and click the System32 subdirectory.
12. In the upper-left list box, click the System32 subdirectory under the QuickFacts directory.
13. Drag the file QFGUITAM.DLL from the upper-right list box to the lower-right list box.

In the **Current Feature** drop-down list, the number (6) should appear next to the Core feature name. This indicates that the Core feature contains 6 files.

This completes the file additions for the Core feature. Now add files to each of the remaining features.

To add files to the remaining features:

1. Add files to the Birds feature:
 - a. From the Current Feature drop-down list, select **Birds**.
 - b. In the lower-left list box, click the QuickFacts directory and click New. Name the new directory FactFiles and click OK.

Note

If the QuickFacts directory does not appear in the lower-left list box, select Tools menu > Options, click the Installation Expert tab, and mark the checkboxes for **View directories for all features on Files page** and **View registry keys for all features on Registry page**.

- c. In the upper-left list box, click the FactFiles subdirectory of the QuickFacts directory.
 - d. In the upper-right list box, select the files:
 - Birds.bmp
 - Birds.qft
 - e. Click Add File to add the files to the lower-right list box.
2. From the Current Feature drop-down list, select **Cats**. Then add the following files to the FactFiles directory for this feature:

- Cats.bmp
- Cats.qft

Because you selected **View directories for all features on Files page** in Wise Options, you can still see the FactFiles directory in the lower-left list box, even though you added it to a different feature.

3. From the Current Feature drop-down list, select **Dogs**. Then add the following files to the FactFiles directory for this feature:

- Dogs.bmp
- Dogs.qft

4. To see all the files you added to the installation for all features, select **All Features (Modify/Delete only)** from the **Current Feature** drop-down list. The number (12) should appear next to the All Features name, indicating that the installation contains 12 files.

You can navigate through the installation directories in the lower-left list box to see the files that you have added.

This completes the file entries for this installation.

Add Registry Keys

To specify the registry entries to be installed or edited on the destination computer, you use the Registry page. You can add registry keys and values manually, but for this tutorial, you import them from a registry file (.REG) that is provided.

1. Select the Registry page.

The upper 2 list boxes show keys and values in your computer's registry. The lower 2 list boxes represent the keys and values to be installed on the destination computer.

Note

To display context-sensitive help, press F1.

2. From the **Current Feature** drop-down list, select **Core**.
3. Click Add at the lower left of the window and select Import.

4. On the Select Registry File to Import dialog, navigate to this product's installation directory, select Tutorial\QuickFacts\Registry File\QuickFacts.reg, and click Open.

QuickFacts.reg contains all the registry keys required for QuickFacts.

5. In the lower-left list box, right-click HKEY_CURRENT_USER and select Expand Children.

This expands the subkeys that you just imported. You can click through the subkeys to see the values that have been added.

This completes the registry entries for this installation.

Add a Shortcut

1. Select the Shortcuts page.
2. From the **Current Feature** drop-down list, select **Core**.
3. Click Add at the right of the window.
4. On the Shortcut Type dialog, make sure the **File in the installation** and **Advertised** options are marked and click Next.
5. On the Shortcut File Selection dialog, select the QuickFacts directory in the left pane, select QckFacts.exe in the right pane, and click Next.
6. On the Shortcut Destination Directory dialog, make sure the Programs directory under the Start Menu is selected, and click Finish.
7. On the Shortcut Details dialog, change the **Name** field to **QuickFacts** and click OK.

A shortcut for QckFacts.exe is added to the installation and will appear under Program Files in the Start menu when the product is installed.

Create a File Association

You can associate file extensions with executables to determine which application to launch when the end user double-clicks a file with a certain extension. The QuickFacts application has its own extension for fact files: .QFT.

1. Under the Feature Details page group, select the File Associations page.
2. From the **Current Feature** drop-down list, select **Core**.
3. Click Add at the right of the window and select New.
The File Association Details dialog appears.
4. Click the Extension Details tab.
5. Browse to the QuickFacts directory, select the file QckFacts.exe, and click OK.
6. In **Extension**, enter: qft
7. Leave the defaults for the rest of the fields and click OK.

The extension .QFT is added to the installation. When an end user double-clicks a file with this extension on the destination computer, the QuickFacts application launches.

8. Save the installation.

The installation now contains files and other system changes that will be installed on the destination computer. In the next section, you determine how the installation behaves on the destination computer.

Customize the Installation's Behavior on the Destination Computer

In this section, you determine how the installation behaves on the destination computer.

Tasks in This Section

- Setting system requirements.
- Selecting the dialogs that appear during installation.
- Configuring installation types.

Before performing these tasks, you must have added information to the installation by following the procedures in [Assemble the Installation](#) on page 18.

Set System Requirements

You can specify minimum hardware and software requirements for the installation and set warning messages that appear to the end user if the destination computer does not meet specified requirements.

1. Under the Target System page group, select the System Requirements page.
2. On the System Requirements page, double-click **Windows Version**.

The **Windows Version** requirement sets requirements for non-NT based operating systems, while the **Windows NT Version** requirement sets requirements for NT-based operating systems.

The Minimum System Requirements dialog appears.

3. In **Windows Version**, select a version that is older than the operating system on your computer. If you select one that is newer, the installation won't run on your computer.
4. In **Message Text**, enter:

This installation requires Windows <version> or later.

where <version> is the version you selected in the **Windows Version** field above.
5. Click OK.

Select Dialogs That Appear to End Users

To determine the level of control the end user has over the installation, you specify which dialogs appear during installation. In this procedure, you set up the QuickFacts installation to let the end user choose which features to install.

1. Under the User Interface page group, select the Dialogs page.
2. Mark the checkboxes for the Installation Type Dialog and the Select Feature Dialog.
3. Leave the defaults for the rest of the dialogs.

The selected dialogs will appear during installation. The Installation Type Dialog lets the end user choose from a Typical, Complete, or Custom installation, and the Select Feature Dialog lets the end user select features in a custom installation.

In the next section, you define which features are installed with each type of installation.

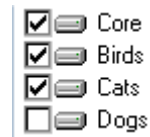
Configure Installation Types

You can define which features are installed when the end user selects Complete, Typical, or Custom on the Installation Type dialog. On the Installation Types page, you turn features on or off for each type of installation.

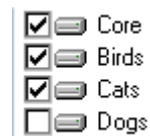
1. Under the User Interface page group, select the Installation Types page.

The upper-left list box shows 3 installation types by default, Typical, Complete, and Custom, which correspond to the 3 radio buttons presented to the end user on the Installation Types dialog during installation.

2. In the upper-left list box, select **&Typical**. (The & indicates which letter will be underlined for the shortcut key.)
3. Make sure the Default Installation Type button is disabled; if not, click it. This sets the Typical installation as the default.
4. In the list on the right, make sure the checkboxes are marked for **Core**, **Birds**, and **Cats**, as shown below. Clear the **Dogs** checkbox. (The features might be listed in a different order in your installation.)



5. In the upper-left list box, select **&Complete** and, in the list box on the right, make sure the checkboxes are marked for all features.
6. In the upper-left list box, select **C&ustom** and, in the list on the right, make sure the checkboxes are marked for the **Core**, **Birds**, and **Cats**. Make sure the **Dogs** checkbox is cleared, as shown below. (The features might be listed in a different order in your installation.)



7. Save the installation.

You have configured the installation to require a specific version of Windows and to let the end user select which features to install. In the next section, you create a standard and a demo edition of the installation.

Define Releases

In this section, you will create a standard edition and a demo edition of the QuickFacts installation. You also will customize properties for the demo release so that the product name reflects that it's a demo.

To generate more than one installation from a single project, you create multiple releases and then configure each release. You can create releases in Windows Installer Editor project files (.WSI) only; release options are disabled in .MSI files.

When you create a release, you can specify whether the compiled installation program is an .EXE or an .MSI and whether to pre-install the Windows Installer software before running the installation. For this tutorial, you will create .MSI files and you will not pre-install Windows Installer.

Tasks in This Section

- Creating releases.
- Customizing the demo release properties.

Before performing these tasks, you must have customized the installation by following the procedures in [Customize the Installation's Behavior on the Destination Computer](#) on page 22.

Create Releases

In this procedure, you create 2 releases for this installation: a standard edition and a demo edition.

1. Under the Release Definition page group (you might need to scroll down to see it), select the Releases page.

A release named Default already exists, because every installation must contain at least one release.

2. Double-click the Default release.

The Release Details dialog appears.

3. Complete the dialog:

- **Release Name**
Change this to Standard.
- **.MSI File Name**
Enter: QuickFacts.msi
- **Description**
Enter: QuickFacts Standard Release
- **Installation Theme, Compression Type, Release Type**
Leave the defaults.
- **Build this release during compile**
Make sure this is marked.

4. Click OK on the Release Details dialog.
5. Click Add at the right of the window to add the demo release.

The Release Details dialog appears.

6. Complete the dialog:

- **Release Name**
Enter: Demo
- **.MSI File**
Enter: QuickFacts_Demo.msi

- **Description**
Enter: QuickFacts Demo Release
- **Installation Theme, Compression Type, Release Type**
Leave the defaults.
- **Build this release during compile**
Mark this checkbox.

Note

An Edit Media button appears at the bottom of the dialog. This button is enabled only when you first add a new release. You don't need to use it in this tutorial.

7. Click OK.

The installation now contains 2 releases. In the next section, you customize the demo release.

Customize the Demo Release Properties

When you create new releases, they retain the properties of the default release. In this installation, the default release is now called the Standard release. In this procedure, you customize the demo release so that:

- It includes limited features.
- The product's name indicates that it is a demo.
- The product's summary informs the end user that this demo release includes limited features.

To customize the demo release:

1. Under the Release Definition page group, select the Release Settings page.
2. From the **Current Release** drop-down list, select **Demo**.
3. To turn off the features that will not be included in the demo release, clear the following checkboxes under the Features icon:
 - **Cats**
 - **Dogs**
4. To change the ProductName property to reflect that this release is a demo edition:
 - a. Select the Properties icon.
 - b. Click Add at the right of the window. The Property Settings Override dialog appears.
 - c. From **Name**, select **ProductName**.
 - d. In **Value**, enter: QuickFacts Demo
 - e. Click OK.
5. To change the Summary settings to inform the end user that this demo release includes limited features:
 - a. Select the Summary icon.
 - b. Click Add. The Summary Settings Override dialog appears.

- c. From **Name**, select **Comments**.
 - d. In **Value**, enter:

This is a demo version; it contains limited features.

This text appears if the end user views properties on the compiled QuickFacts_Demo.msi.
 - e. Click OK.
6. Save the installation.

Congratulations! You have just completed an installation, without having to edit even one table. In the next section, you compile and test the installation.

Compile and Test the Installation

Now that you have completed an installation project, you need to compile it into an .MSI. Because the installation consists of 2 releases, compiling will result in 2 .MSI files, one for each release. After compiling, you can test the installation. Before performing this task, you must have created releases by following the procedures in [Define Releases](#) on page 23.

Note

This procedure guides you through testing the installation on your computer. If you were really distributing this installation, it would require more thorough testing on multiple operating systems.

When you test the installation, it runs in test mode. The installation appears as it does to the end user, but no files are actually installed or modified, and no changes are made to the system, although the installation otherwise performs all the tasks in the installation.

You can then run a full installation, which installs files and modifies the system. After you run an installation, you must uninstall before you can run the installation again. This is because Windows Installer detects when a product is installed and runs the subsequent installation in maintenance mode.

To compile and test the installation:

1. Test the standard release.
 - Click Test at the lower right of the window.
 - Because this installation has 2 releases, you must select which one to test. On the Select an .MSI File dialog, select **QuickFacts.msi** and click OK.

The installation is compiled, and then a simulated installation is run for the standard release.
2. Step through the dialogs of the installation until the Select Installation Type dialog appears.
3. Click **Custom** and click Next.

Notice that all features are visible on the Select Features dialog, though one of them has an "X" icon, indicating that it will not be installed. This is the result of the settings you entered on the Installation Types page.
4. Proceed by accepting the defaults on the remaining dialogs.

5. When the test finishes, test the demo release. Select the demo release in the Select an MSI File dialog and repeat the preceding steps, including selecting **Custom** on the Select Installation Types dialog.

Notice that the product name on the installation dialogs includes "Demo." Also, only 2 features are visible on the Select Features dialog. This is the result of the features you set on the Release Settings page.

6. Proceed by accepting the defaults on the remaining dialogs.
7. Now run the installation of the standard release, which installs the files.
 - a. Click Run at the lower right of the window and select Run from the button menu.
 - b. On the Select an .MSI File dialog, select **QuickFacts.msi** and click OK.

The standard release of the QuickFacts installation is run. Accept all dialog defaults and click Finish when the installation finishes.

8. Select Windows Start menu > Programs > QuickFacts.

The QuickFacts application starts.

- The upper-left list box lists the fact files that are available for viewing. Because you installed a Typical installation, you do not see the Dogs fact files. (The Core feature, which comprises the QuickFacts application and help, is installed also.)
- Exit QuickFacts and double-click a .QFT file in Windows Explorer, which should launch QuickFacts. The .QFT files are in Program Files\QuickFacts\FactFiles.

Note

Do not run an actual installation of the demo release. The demo release cannot be installed until the standard release has been uninstalled.

This completes the basic Windows Installer Editor tutorial. Now you know how to create an installation using Installation Expert. The Advanced Tutorial describes how to add custom actions to the installation using the MSI Script view.

Chapter 3

Advanced Tutorial: Using MSI Script in a Windows Installer Installation

This tutorial guides you through the basics of creating custom actions in a Windows Installer installation. Because this tutorial builds on the installation you created in the previous chapter, you must first complete [Basic Tutorial: Creating a Windows Installer Installation](#) on page 14.

This tutorial takes about 30 to 45 minutes to complete. If you stop before finishing this tutorial, save the installation and resume the tutorial at a later time.

This tutorial contains examples of how to:

- [Call a .DLL With a Custom Action](#)
- [Set a Directory With a Custom Action](#)
- [Call an .EXE With a Custom Action](#)

These exercises demonstrate the power and versatility that custom actions add to the installation-building process.

This section also describes some of the advanced tasks you can perform using Setup Editor in Windows Installer Editor. See [Performing Advanced Tasks](#).

Call a .DLL With a Custom Action

In this section, you create a custom action that displays a dialog during the installation of files. The dialog shows the value of the property `INSTALLDIR`, which contains the default installation directory. To do this, you call a standard Windows .DLL that already exists on the destination computer.

This exercise demonstrates calling a .DLL, not displaying a message dialog. There is an easier way to display a message dialog—use the Display Message action in MSI Script.

Note

The property `INSTALLDIR` stores the default installation directory. It defaults to the first directory you create in the installation. You can see it in the **Default Directory** field in Installation Expert > Product Details page. During installation, the end user can change the value of `INSTALLDIR` on the Single Feature Destination dialog.

To call a .DLL during installation:

1. In Windows Installer Editor, open QuickFacts Installer.wsi, which you created during the basic tutorial.

It may already be open if you are continuing from the basic tutorial.

2. Click MSI Script at the lower left of the Windows Installer Editor window.

MSI Script™ contains the sequences of actions that make up an installation.

3. Make sure **Normal Installation** is selected in the **Installation Mode** drop-down list.

Normal Installation refers to a regular installation or uninstallation of the product.

4. Click the Execute Immediate tab at the bottom of the **Installation Sequence** pane.

You will add the action to the Execute Immediate sequence immediately before the installation of files begins.

5. Locate the InstallFiles action in the Execute Immediate script:

- a. Press Ctrl+F to open the Find dialog.
- b. In the Find dialog, enter **InstallFiles** in the **Find What** field and make sure **Normal Installation** is selected in the **In What** drop-down list.
- c. Click Find Next.
- d. Click Cancel after the InstallFiles action is found.

6. Select the REM statement above the InstallFiles action.

When you add a new action, it always appears above the line that is selected in the right pane.

7. In the **Actions** list, double-click Call Custom DLL from Destination.

Call Custom DLL actions let you send parameters while Call DLL actions do not.

The Call Custom DLL From Destination dialog appears.

Note

Although this tutorial guides you through calling a particular Windows .DLL function, it is not the purpose of this tutorial to teach you about .DLLs. Consult Windows developer documentation or other documentation to learn more about calling system .DLLs or the MessageBoxA API.

8. Click the Details tab and complete the dialog:

Note

This tutorial does not explain all the options available on this dialog. Press F1 on the dialog for details on all options.

■ Custom Action Name

Enter: DisplayMessageBox

■ DLL File

◆ (Windows NT, 2000, or XP.) Enter: SystemFolder\user32.dll

◆ (Windows 95, 98, or Me.) Enter: System16Folder\user32.dll

The predefined directories, SystemFolder and System16Folder, hold paths to the \System32 and \System directories, respectively. All predefined directories are listed in the Directory table in Setup Editor > Tables tab. If you would actually deploy this action, you'd have to construct If Statement structures that checked the operating system and then referred to the correct directory.

■ Function Name

Enter: MessageBoxA (case-sensitive), which is the name of a function within User32.dll.

■ **Parameters List**

In the parameters list, use the Add button to add the 4 parameters listed below. Click OK on the DLL Parameter Details dialog after entering each parameter.

◆ First parameter:

Parameter type: long

Value Source: Constant with NULL value

Because **Constant with NULL value** passes a value of NULL regardless of the parameter type, the Details tab displays Not applicable for the parameter type.

◆ Second parameter:

Parameter type: string pointer

Value Source: Property

Property Name: Enter: INSTALLDIR

◆ Third parameter:

Parameter type: string pointer

Value Source: Constant

Constant Value: Enter: The value of INSTALLDIR is:

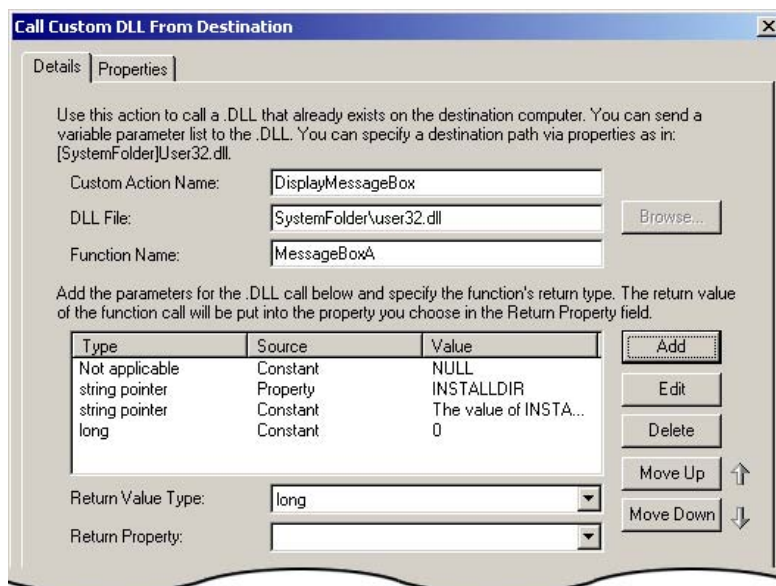
◆ Fourth parameter:

Parameter type: long

Value Source: Constant

Constant Value: Enter: 0

9. Leave the defaults for the Return fields and the Properties tab. The dialog should look like this:



10. Click OK on the Call Custom DLL From Destination dialog.

The action is added to the sequence, immediately after the MoveFiles action. The next steps demonstrate how to enclose the new action in an If Statement, which sets a condition on the action—the action runs only if the condition is true.

11. Select the Call Custom DLL From Destination action that you just created, and then in the **Actions** list, double-click If Statement.

The If Settings dialog appears.

12. In **If Condition**, enter NOT Installed (case sensitive) and click OK.

The Windows Installer property named *Installed* is true if the product is already installed, so this action will run only on the initial installation of the product, not on subsequent maintenance installations. See *Installed Property* in the Windows Installer SDK Help.

13. Select the line below the Call Custom DLL From Destination action and double-click End Statement in the **Actions** list.

This encloses the custom action in an If block, which is located between MoveFiles and InstallFiles and should look like the image below. If not, double-click the actions to edit their settings or use the Move Up and Move Down commands on the right-click menu to rearrange the lines.

```

REM Create empty folders for components that are set to be installed.
CreateFolders
REM Locate existing files on the destination computer and move or copy those files to a new location.
MoveFiles
If NOT Installed then
  Call Custom DLL From Destination SystemFolder\user32.dll Function MessageBoxA (DisplayMessageBox)
End
REM Copy files from the source directory to the destination directory.
InstallFiles

```

To test the custom action:

1. Save the installation.
2. Compile the installation by clicking Compile at the lower right of the main window.
3. To run the installation, click the Run button, select **Run** from the button menu, and when prompted, select QuickFacts.msi.

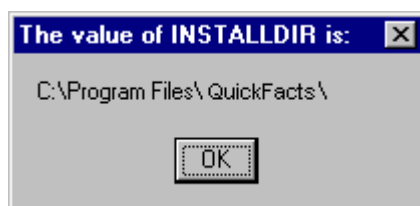
If QuickFacts is already installed, the Application Maintenance dialog appears. Mark **Remove** and click Next to uninstall it. Then run the QuickFacts.msi again.

Note

Avoid uninstalling by selecting one of the other options from the Run button, which either uninstalls and then reinstalls, or forces a reinstall over an installed product.

4. In the installation wizard, click Next until the installation of files begins.

During installation of files, a small dialog appears when Windows Installer calls the function MessageBoxA from user32.dll.



5. Click OK to close the dialog and then click Finish at the end of the installation.

If the dialog does not appear as expected, verify that the parameters and function name in the custom action match the instructions in this tutorial. Also verify that the custom action is in the correct position in the sequence.

Set a Directory With a Custom Action

In this section, you create a custom action that changes the location of the installation based on end user interaction with the installation wizard. To do this, you:

- Create a checkbox on the Single Feature Destination dialog.
- Create a custom action that redefines the installation directory so that it is located under the Common Files directory.

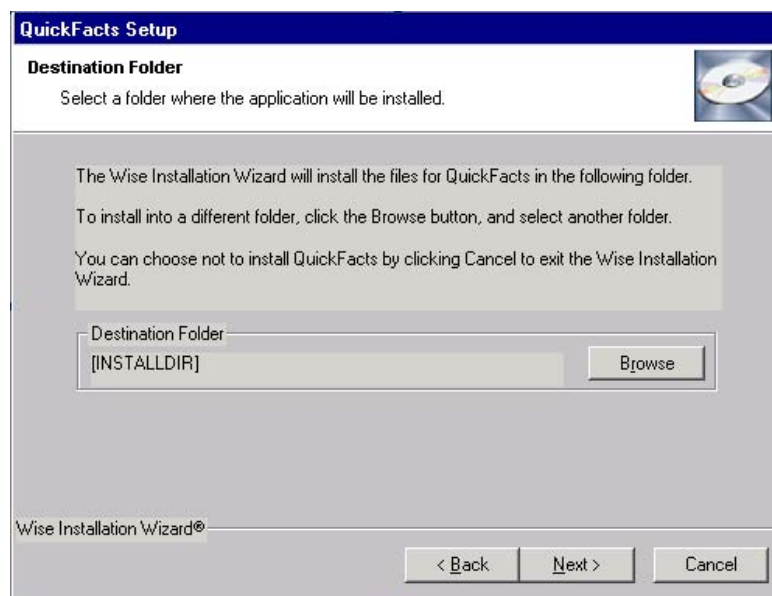
During installation, the end user can mark or clear the checkbox, thereby triggering or skipping the custom action.

Before you perform this task, complete the task [Call a .DLL With a Custom Action](#) on page 28.

To create a checkbox on the Single Feature Destination dialog:

1. Click Setup Editor at the lower left of the Windows Installer Editor window.
2. In the right pane, click the Dialogs tab, and then find and select the dialog named Single Feature Destination, which appears under Install Dialogs.

The dialog, which is labeled Destination Folder, appears in the upper-right pane.



3. In the blank area under the **Destination Folder** group box, right-click and select Add > Checkbox.

The Properties for Checkbox dialog appears, where you set a property and control text associated with the checkbox.

4. In **Control Text** on the Control tab, enter:

Put installation under the Common Files folder

This is the text label for the checkbox.

5. Click the New button to the right of the **Property** drop-down list.

The Property Details dialog appears, where you create a new property to hold the value of the checkbox.

6. Complete the dialog:

- a. In **Name**, enter COMMONDIR (all uppercase).

- b. Leave the **Value** field blank.

The checkbox value must evaluate to blank (nothing) initially so that the checkbox is not marked when end users first see it.

- c. Leave the defaults in the remaining fields.

- d. Click OK.

Note

Technically, leaving a property blank is not allowed according to Windows Installer guidelines, but in this tutorial, you leave it blank so that the checkbox is initially unmarked. Because the property value is blank, the property does not appear under the Properties icon in Setup Editor > Product tab.

7. Click OK on the Properties for Checkbox dialog.

The checkbox appears on the Destination Folder dialog.

8. If necessary, drag the checkbox so it is aligned with other controls on the dialog.

In the next procedure, you create a custom action that tests the value of the COMMONDIR property during installation. If COMMONDIR is true (which means that the checkbox is marked), the custom action resets the value of the installation directory (INSTALLDIR) so that it is placed in the Common Files directory.

To add a custom action that defines INSTALLDIR based on user input:

1. Select MSI Script.
2. Make sure **Normal Installation** is selected in the **Installation Mode** drop-down list.
3. Click the User Interface tab at the bottom of the **Installation Sequence** pane.

For this tutorial, you add the action to the User Interface sequence immediately after the Welcome_Dialog If block. The Welcome_Dialog action contains all the wizard dialogs the end user sees during installation. The User Interface sequence gathers end user input and sets installation properties.

4. Scroll to near the end of the script and select the "If Installed AND (RESUME OR Preselected) AND NOT PATCH then" statement. You will place a new If block above this line.
5. In the **Actions** list, double-click Set Directory.

The Set Directory dialog appears.

6. Click the Details tab and complete the dialog:

- **Custom Action Name**

Enter: ChangeDirectory

- **Directory**

Browse to the QuickFacts directory, which is the installation directory represented by the INSTALLDIR property. Click OK.

INSTALLDIR appears in the **Directory** field.

If the end user marks the **Put installation under the Common Files folder** checkbox during installation, this is the directory whose location is redefined.

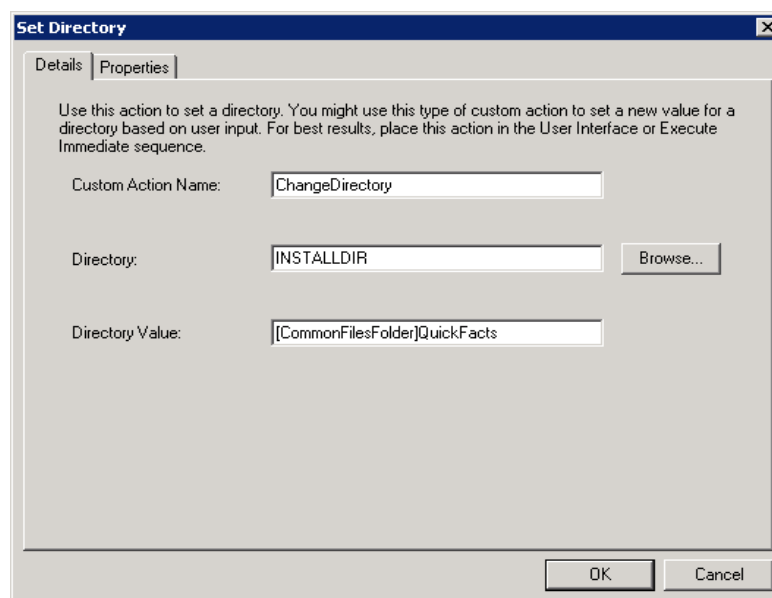
- **Directory Value**

Enter (case-sensitive):

[CommonFilesFolder]QuickFacts

This will be the new location of the installation directory if the end user marks the checkbox during installation. CommonFilesFolder is a directory property that represents the Common Files directory on the destination computer.

7. Leave the defaults on the Properties tab. The dialog should look like this:



8. Click OK on the Set Directory dialog.

The action is added to the sequence, immediately after the Welcome_Dialog If block. In the next steps, you enclose this action in an If Statement, which sets a condition on the action—the action runs only if the condition is true.

9. Make sure the Set Directory custom action you just created is selected. In the **Actions** list, double click If Statement.



The If Settings dialog appears.

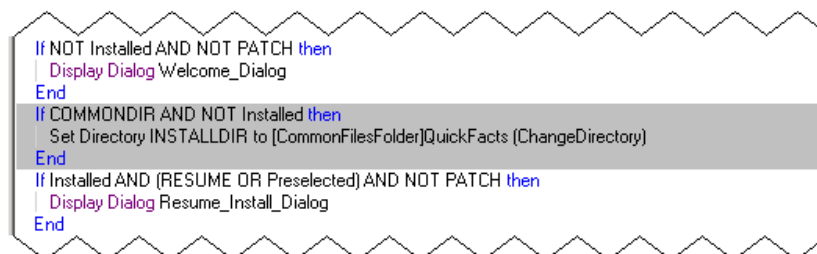
10. In **If Condition**, specify 2 conditions joined by “AND”. Enter the following (case-sensitive) and click OK:

```
COMMONDIR AND NOT Installed
```

The first condition checks the COMMONDIR property you attached to the **Put installation under the Common Files folder** checkbox, which is true if the end user marks it during installation. The second condition checks a built-in Windows Installer property, named Installed, which is true only if this is the initial installation of the product, not a reinstall or maintenance installation.

11. Place an End Statement below the Set Directory custom action.

The If Statement, custom action, and End Statement should look like the image below. If not, double-click the actions to edit their settings or use   to rearrange the lines.



To test the custom action:

1. Save the installation.
2. Compile the installation by clicking Compile at the lower right of the main window.
3. To run the installation, click the Run button, select **Run** from the button menu, and when prompted, select QuickFacts.msi.

If QuickFacts is already installed, the Application Maintenance dialog appears. Mark **Remove** and click Next to uninstall it. Then run the QuickFacts.msi again.

Note

Avoid uninstalling by selecting one of the other options from the Run button, which either uninstalls and then reinstalls, or forces a reinstall over an installed product.

4. In the installation wizard, click Next on all the dialogs, leaving the **Put installation under the Common Files folder** checkbox cleared.

Because of the custom action you added earlier in this tutorial, a dialog displays the value of INSTALLDIR during installation. After installation, notice that the QuickFacts directory is in the Program Files directory.

5. Now uninstall QuickFacts and run the installation again, this time marking the **Put installation under the Common Files folder** checkbox.

When this installation is finished, the QuickFacts directory should appear in your Common Files directory.

Note

If you are running Windows 95 or Windows 98, and this custom action does not work, go to the Merge Modules page and add the merge module CondFix to the installation. This merge module fixes a Windows Installer limitation with setting properties during the UI Sequence. See the description of CondFix in the Windows Installer Editor help.

Call an .EXE With a Custom Action

In this section, you create a custom action that calls an .EXE and passes parameters to it. This action calls Notepad.exe and passes the name of the QuickFacts readme file to it, causing the readme file to be displayed during installation.

Before you perform this task, complete the task [Call a .DLL With a Custom Action](#) on page 28.

To create a custom action that opens a file in the installation:

1. Select MSI Script.
2. Make sure **Normal Installation** is selected in **Installation Mode**.
3. Click the Execute Deferred tab at the bottom of the **Installation Sequence** pane.

For this tutorial, you add the action to the Execute Deferred sequence immediately after the InstallFiles action. This action must run in deferred mode because it relies on an installed file (readme.txt), which is not present on the destination computer until after the InstallFiles action runs in the Execute Deferred sequence.

4. Locate the InstallFiles action in the Execute Deferred script:

- a. Press Ctrl+F to open the Find dialog.
 - b. In the Find dialog, enter **InstallFiles** in the **Find What** field and make sure **Normal Installation** is selected in the **In What** drop-down list.
 - c. Click Find Next.
 - d. Click Cancel after the InstallFiles action is found.
5. Select the REM statement below the InstallFiles action.
 6. In the **Actions** list, double-click Execute Program From Destination.
The Execute Program From Destination dialog appears.
 7. Click the Details tab and complete the dialog:

Note

This tutorial does not explain all the options available on this dialog. Press F1 on the dialog for details on all options.

■ **Custom Action Name**

Enter: OpenReadMe

■ **Working Directory**

Enter: INSTALLDIR

This sets the current working directory of the .EXE when it runs on the destination computer. In this case, Notepad.exe looks in the installation directory for the readme file you specify below.

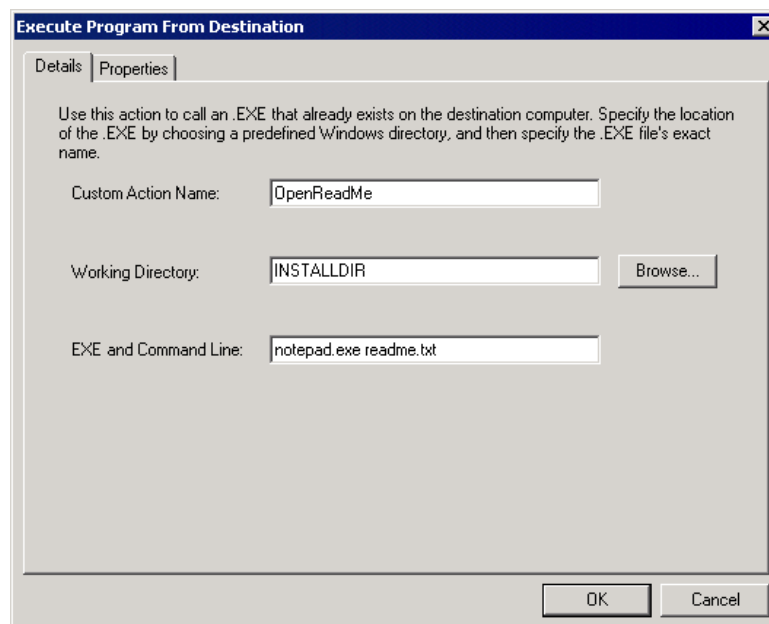
■ **EXE and Command Line**

Enter: notepad.exe readme.txt

Note

Because Notepad is in the PATH variable, you do not have to specify its full path. If you were calling an .EXE that is not in the PATH variable, you would have to specify the full path here, using bracketed property names to form the pathname. Also, you don't have to specify the full path to readme.txt because in the previous field you set Notepad's current working directory to the QuickFacts directory, where readme.txt resides.


The dialog should look like this:

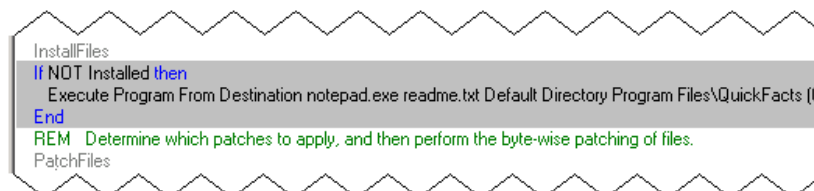


8. Click the Properties tab.
9. In **Processing**, select **Asynch, No wait**, which makes the installation continue after opening Notepad.
10. Click OK on the Execute Program From Destination dialog.

The action is added to the sequence immediately after the InstallFiles statement. In the next steps, you enclose the action in an If Statement, which sets a condition on the action—the action runs only if the condition is true.

11. Above the Execute Program From Destination action, add an If Statement action. In **If Condition**, enter: NOT Installed.
12. Click OK.
13. Below the Execute Program From Destination action, add an End Statement.

The If Statement, custom action, and End Statement, which are located between InstallFiles and PatchFiles, should look like the image below. If not, double-click the actions to edit their settings or use  to rearrange the lines.



To test the custom action:

1. Save the installation.
2. Compile the installation by clicking Compile at the lower right of the main window.
3. To run the installation, click the Run button, select **Uninstall ---> Install** from the button menu, and when prompted, select QuickFacts.msi.

4. In the installation wizard, click Next until the installation of files begins.

Because of the custom action you added earlier in this tutorial, a dialog displays the value of INSTALLDIR during installation.

After the installation of files, Notepad should appear and display the readme.txt file. If this does not work, check the order of the custom action. It must occur after the InstallFiles action, or it will not work. Also make sure it is in the Execute Deferred sequence and that all fields are entered correctly.

This completes the tutorial. The next section describes some of the advanced tasks you can perform in Setup Editor.

Performing Advanced Tasks

As you learned from the Basic and Advanced Tutorials, you can easily build complete Windows Installer installations without using Setup Editor. If you are developing complex installations, however, you might need to perform fine-tuning or advanced tasks in Setup Editor.

To access Setup Editor, click Setup Editor at the lower left of the window. In Setup Editor, you can:

- Edit tables of the Windows Installer database (recommended for advanced Windows Installer developers only).
- Edit installation dialogs by adding and editing text and controls.
- Select dialogs to appear during maintenance installations (repair and uninstall).
- Create properties for use in custom installation logic.
- Create and edit components manually and assign them to features.
- Build complex conditions that must be met for an installation to occur.

For instructions on the tasks listed above, see Windows Installer Editor help.

Chapter 4

Tutorial: Creating a Windows Installer Installation in Visual Studio Integrated Editor

This tutorial is provided for users who plan to use the Visual Studio integrated editor to synchronize installations with other projects in a Visual Studio .NET solution.

First, follow the Basic tutorial using the Wise editor. (See [Basic Tutorial: Creating a Windows Installer Installation](#) on page 14.) Then follow this tutorial using the Visual Studio integrated editor. This tutorial includes only the procedures that have content that is specific to the Visual Studio integrated editor.

Note

If you install Wise Installation Studio or Wise Installation Express before you install Visual Studio, some of the files you need for this tutorial are not installed. To install those files, perform a repair on the Wise Installation Studio or Wise Installation Express installation.

This tutorial takes about 30 minutes to complete. If you must stop before finishing this tutorial, save the installation and resume the tutorial at a later time.

In this tutorial, you will use Windows installer Editor's Installation Expert to:

- [Open the C# QuickFacts Solution and Create an Installation Project](#)
- [Open the Installation Project and Set Options](#)
- [Enter Product Details and General Information](#)
- [Add Files to the Installation](#)

Open the C# QuickFacts Solution and Create an Installation Project

QuickFacts is a solution, written in C#, that consists of 3 projects:

- FactFinder, which compiles to FactFinder.dll.
- QuickFacts, which compiles to QuickFacts.exe.
- TextViewer, which compiles to TextViewer.dll.

QuickFacts is dependent on FactFinder, and FactFinder is dependent on TextViewer.

To start an installation project:

1. Start Microsoft Visual Studio .NET.
2. Select File menu > Open > Project. In Visual Studio 2005, select File > Open > Project/Solution.
3. Navigate to the Tutorial\Integrated Tutorial\QuickFacts subdirectory of this product's installation directory.
4. Select QuickFacts.sln and click Open.

- In Visual Studio .NET 2003 or later, a message appears asking if you want to convert this solution. Click Yes.
 - In Visual Studio 2005, the Visual Studio Conversion Wizard appears. Step through the wizard and accept all defaults.
 - If a message appears about some properties not being read, click OK.
5. Select Build menu > Build Solution to compile all projects.
 6. When the compile finishes, select File menu > New > Project.
The New Project dialog appears.
 7. On the New Project dialog:
 - a. In **Project Types**, select Wise Setup and Deployment Projects.
 - b. In **Templates**, select Setup Wizard.
 - c. In **Name**, enter QuickFacts Installer.
 - d. Before you set the **Location**, select the **Add to Solution** option.
In Visual Studio 2005, this is available from the **Solution** drop-down list. In earlier versions, this is a radio button option.
 - e. In **Location**, specify the Tutorial\Integrated Tutorial\QuickFacts subdirectory of this product's installation directory.
 - f. Click OK.The Wise Setup Wizard appears with pages for Overview, Project Type, Projects, and Main Project.
 8. Select the Overview page.
It contains a summary of the settings for the installation project. It indicates that the primary outputs for each project (TextViewer.dll, FactFinder.dll, and QuickFacts.exe) are added to the installation project.
 9. Select the Project Type page.
This page is used to specify the type of Windows Installer project to create. For this tutorial, leave **Windows Application** marked.
 10. Select the Projects page. The default settings should be correct.
 - **Select the projects to include in this package**
The solution's projects are selected.
 - **Scan Method**
With **Always Scan Solution** selected, if the primary outputs for the 3 projects change, the files in the installation will change accordingly.
 - **Bind installed files to the solution build configuration**
With this marked, if you change your build configuration from Release to Debug or vice versa, the installation project will draw its files from the corresponding directories.
 11. Select the Main Project page. The default settings should be correct.
 - **Main Project**
QuickFacts should be selected.

- **Automatically Update Product Information**

When this is marked, the version number and other summary information in the installation is updated whenever the main target file changes. The information is updated each time you build your solution.

- **Create Shortcut**

When this is marked, shortcut is created for the installation project.

12. Click Finish.

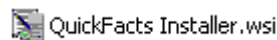
A new project named QuickFacts Installer is added to the other projects in Solution Explorer. (If Solution Explorer is not visible, select View menu > Solution Explorer.) FactFinder.dll, QuickFacts.exe, and TextViewer.dl are in the project's Source Files folder. They were added to the installation project because they are primary output files from the existing C# projects.

Open the Installation Project and Set Options

You use the Wise Options dialog to customize the behavior of Windows Installer Editor for all installations you create.

Windows Installer projects are comprised of features, and normally, you view folders and registry keys per feature. In this procedure, you open the installation project that you created and set options that let you view folders and registry keys for all features simultaneously, making it easier to see the installation as a whole.

1. In Solution Explorer, double-click QuickFacts Installer.wsi. Make sure you double-click the .WSI file, shown below, not the project name.



The Visual Studio integrated editor opens and displays QuickFacts Installer.wsi.

2. Select Tools menu > Options.

The Options dialog appears.

3. In the left pane, click Wise Options to expand the list and then select Installation Expert.
4. In Installation Expert Options, mark the following. If you marked these in the Basic tutorial, they will be marked already.

- **View directories for all features on Files page**

This lets you see all directories in Installation Expert > Files page, regardless of what feature each directory was created for.

- **View registry keys for all features on Registry page**

This lets you see a composite view of all registry keys in Installation Expert > Registry page, regardless of what feature the registry key was created for.

Note

To display context-sensitive help, click Wise Help on the Options dialog.

5. Click OK.

Enter Product Details and General Information

To enter information about the installation, you use pages in the Project Definition page group in Installation Expert.

1. If necessary, maximize screen space by closing other windows, such as Solution Explorer, Properties, Class View, Output, and Dynamic Help.
2. Click Installation Expert at the lower left of the Visual Studio integrated editor window.
3. Under the Project Definition page group on the left, click Product Details.

The Product Details page appears. It contains meta data that is used in the installation dialogs to identify the product to the user who is installing it. The **Product Name**, **Manufacturer**, **Version**, and **Default Directory** fields are populated with information from the main project.

4. In the **Application Type** field, select **.NET Application** in the Value column. Leave the defaults in the remaining fields.
5. Select the General Information page by clicking its name.

On the General Information page, you set the information that appears when the end user right-clicks the installation file in Windows Explorer and selects Properties.

6. Complete the General Information page:

- **Title**
Enter: QuickFacts
- **Author**
Enter your name.

7. Select the Add/Remove Programs page.

On the Add/Remove Programs page, you enter information to support the capabilities of the enhanced Add/Remove Programs control panel that was introduced with Windows 2000.

8. Complete the Add/Remove Programs page:

- **Display in Add/Remove Programs list**
Make sure this is marked.
- **Product Updates URL**
Enter: No support provided.

9. Save the installation by selecting File menu > Save QuickFacts Installer.wsi.

Now that you've entered basic information about the installation, you can define the features that provide structure to the installation. To do this, see [Create and Organize Features](#) on page 17. After you define features, you can add files to the installation.

Add Files to the Installation

On the Files page, you specify the directories and files that will be installed on the destination computer. You specify files per feature; the files that you add are only installed on the destination computer if the feature is installed. Because this installation has multiple features, be sure to specify the feature for which you are adding files. You do this by selecting it from the **Current Feature** drop-down list.

In the Visual Studio integrated editor, installations synchronize automatically with the other projects in the solution. Example: Adding .EXEs, .DLLs, .OCXs, and assemblies to the solution adds them to the installation. Therefore, this installation already contains files that are primary outputs of the 3 C# projects. You only need to add files that are not within the solution.

Before performing this procedure, you must add features as described in [Create and Organize Features](#) on page 17.

To add files to the Core feature:

1. Under the Feature Details page group, select the Files page.

Note

To display context-sensitive help, press F1.

2. From the **Current Feature** drop-down list, select **Core**.
3. In the lower-left list box, find and select Program Files\QuickFacts\QuickFacts.
It contains 3 files (FactFinder.dll, QuickFacts.exe, and TextViewer.dll) that are primary output files of the 3 C# projects. These are displayed in the lower-right list box.
4. In the upper-left list box, navigate to the Tutorial\Integrated Tutorial\Content Files subdirectory of this product's installation directory.
The files in the Content Files directory appear in the upper-right list box.
5. Make sure the QuickFacts\QuickFacts directory is still selected in the lower-left list box. In the upper-right list box, select License.txt and ReadMe.txt and click Add File.
The 2 files are added to the QuickFacts\QuickFacts directory in the installation.
6. In the upper-left list box, expand the Content Files directory and select the Help subdirectory.
7. Make sure the QuickFacts\QuickFacts directory is still selected in the lower-left list box, and click Add Contents.
8. On the Add Contents dialog, leave the defaults and click OK.
The Help directory and its contents are added to the installation.
9. Save the installation.
10. If the Copy Source Files dialog appears, click Cancel. This is related to Source Code Control, which is not used in this tutorial.

In the **Current Feature** drop-down list, the number (10) should appear next to the Core feature name. This indicates that the Core feature contains 10 files.

This completes the file additions for the Core feature. Now add files to each of the remaining features.

To add files to the remaining features:

1. Add files to the Birds feature:
 - a. From the Current Feature drop-down list, select **Birds**.
 - b. In the lower-left list box, click the QuickFacts\QuickFacts directory and click New. Name the new directory FactFiles and click OK.

- c. In the upper-left list box, click the FactFiles subdirectory of the Tutorial\QuickFacts directory.
 - d. In the upper-right list box, select the files:
 - Birds.bmp
 - Birds.qft
 - e. Click Add File to add the files to the lower-right list box.
2. From the Current Feature drop-down list, select **Cats**. Then add the following files to the FactFiles directory for this feature:
 - Cats.bmp
 - Cats.qft

Because you selected **View directories for all features on Files page** in Wise Options, you can still see the FactFiles directory in the lower-left list box, even though you added it to a different feature.

3. From the Current Feature drop-down list, select **Dogs**. Then add the following files to the FactFiles directory for this feature:
 - Dogs.bmp
 - Dogs.qft
4. To see all the files you added to the installation for all features, select **All Features (Modify/Delete only)** from the **Current Feature** drop-down list. The number (16) should appear next to the All Features name, indicating that the installation contains 16 files.

You can navigate through the installation directories in the lower-left list box to see the files that you have added.

This completes the file entries for this installation.

The rest of the procedures in the Basic tutorial are the same for both editors except for the following differences in the Visual Studio integrated editor:

- You don't need to add a shortcut on the Shortcuts page, because it is added automatically. However, on the Shortcuts page you should select the Core feature, double-click the shortcut, and change its destination directory to Windows\Profiles\Start Menu\Programs.
- To compile the installation, select Build menu > Build QuickFacts Installer.
- To test the installation, select Project menu > Start in Test Mode.
- To run the installation, right-click the QuickFacts Installer project icon in Solution Explorer and select Set as Startup Project. Then select Debug menu > Start Without Debugging.

For other differences between the Wise editor and the Visual Studio integrated editor, see the Windows Installer Editor Help.

Index

Symbols

.NET Framework requirements 9

A

Altiris
consulting 6
training 6
Altiris Software Virtualization Agent installing 8

C

consulting, Altiris 6

D

documentation, Wise 4

E

evaluation
converting to production 12

G

Getting Started Guide 4

H

help
Windows Installer SDK 4

I

installation
Language Pack 12
upgrade 10
integrated editor, tutorial 40

L

Language Pack, installing 12
launching Wise Installation Studio/Express 12

M

manual
accessing online 4
manual, reference 4
Mobile Device Editor 8
MSI Script, tutorial 28

N

new features
Refer to Release Notes
newsgroups 5

P

prerequisites
.NET Framework 9
installation 8
tutorial 14

R

readme
See release notes
reference manual 4
accessing online 4
registration, product 13
release notes 5

S

sales contact, Altiris 6
serial number 10
Software Virtualization Agent installing 8
starting Wise Installation Studio/Express 12
support
newsgroups 5
online support 5
support team 5
SVS Agent installing 8
system requirements
.NET Framework 9
Wise for Windows Installer 8
system requirements, Wise product
Refer to Getting Started Guide

T

technical support
newsgroups 5
online support 5
support team 5
training, Altiris 6
tutorial
advanced 28
basic 14
integrated editor 40
prerequisites 14
Refer to Getting Started Guide

U

upgrade
installing 10

V

Virtual Package Editor 7

Visual Studio integrated editor
tutorial 40

W

Windows Installer
developer documentation 4
help 4
Windows Installer Editor
about 7
Windows Installer SDK Help 4
Wise for Windows Installer 7
Wise Installation Express
about 8
Wise Installation Studio
about 7
Wise Installation System 7
Wise Mobile Device Editor
about 8
Wise Virtual Package Editor
about 7
WiseScript Editor
about 8
WiseScript Express
See WiseScript Editor
WiseScript Package Editor
about 7