



InstallShield®
AdminStudio™

InstallShield® AdminStudio™ Getting Started Guide

© 1999–2001 InstallShield Software Corporation. All Rights Reserved.

This document is being furnished by InstallShield Software Corporation for information purposes only to licensed users of the InstallShield AdminStudio software product and is furnished on an “AS IS” basis, that is, without any warranties, whatsoever, express or implied. InstallShield is a registered trademark and service mark of InstallShield Corporation. DemoShield is a trademark of InstallShield Software Corporation. Other brand and product names are trademarks or registered trademarks of the respective holders. Microsoft Windows, Windows NT, the Windows logo, the Windows NT logo and FrontPage are registered trademarks of the Microsoft Corporation. Adobe Acrobat is a registered trademark of Adobe Systems Incorporated. Information in this document is subject to change without notice and does not represent any commitment on the part of InstallShield Software Corporation. The software described in this document is furnished under a license agreement. The software may be used only in accordance with the terms of that license agreement. It is against the law to copy or use the software except as specifically allowed in the license. No part of this document may be reproduced or retransmitted in any form or by any means, whether electronically or mechanically, including, but not limited to the way of: photocopying, recording, or information recording and retrieval systems, without the express written permission of InstallShield Software Corporation.

InstallShield AdminStudio Getting Started Guide
Initial Publishing June 2001.
Printed in the USA
Part No. ASGSG01–0701

Visit our Web site at <http://www.installshield.com>.

Record Your Product Serial Number Here:

	—		—	
--	---	--	---	--

June 2001

Dear InstallShield AdminStudio Customer:

Thank you for choosing InstallShield AdminStudio 2.0. The Getting Started Guide, Online Reference, and Knowledge Base compose the comprehensive InstallShield AdminStudio documentation set.

For additional InstallShield AdminStudio information, please visit our Web site at <http://www.installshield.com/isas>. The latest InstallShield news, hot installation topics, technical white papers, and downloadable files are now available to you 24 hours a day, seven days a week.

We are committed to providing you with the most innovative software distribution technology. As such, we welcome your comments and suggestions about InstallShield AdminStudio and its documentation, or any other input you might have. Your feedback is invaluable. Please contact us through the Support link of the InstallShield Web site at <http://www.installshield.com>.

Thank you for using InstallShield AdminStudio.

Sincerely,

Viresh Bhatia
CEO, InstallShield Software Corporation

Alaks Sevugan
Product Director

Eric Thomas
Technical Writer

Table of Contents

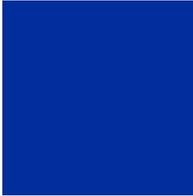
Welcome to InstallShield AdminStudio	xiii
AdminStudio Features and Benefits	xv
InstallShield AdminStudio Benefits	xv
InstallShield AdminStudio Features	xvi
Repackaging	xvi
Authoring	xvi
Tuning	xvii
Solving Conflicts	xvii
About InstallShield Software Corporation	xviii
Fast Facts about InstallShield	xviii
InstallShield's Products	xix
Installation Development	xix
InstallShield Developer	xix
InstallShield Professional	xix
InstallShield Express	xx
InstallShield Multi-Platform Edition	xx
Administrator Solutions	xx
InstallShield AdminStudio	xxi
Multimedia Authoring	xxi
DemoShield	xxi
Technical Support Resources	xxii
Online Help	xxii
Help Library	xxii
Context-Sensitive Help	xxii
Web-Based Help	xxii
InstallShield Knowledge Base	xxii
InstallShield Newsgroups	xxiii
InstallShield Customer Care—Technical Support	xxiii
Support Programs	xxiii
Contacting InstallShield Customer Care	xxiv
InstallShield Professional Services	xxv
InstallShield Consulting Services	xxv
InstallShield Training	xxvi

Chapter 1	Installing AdminStudio	1-1
	Document Conventions	1-3
	System Requirements	1-3
	Installation Procedure	1-4
	Uninstallation Information	1-4
	Register Your Software	1-4
Chapter 2	InstallShield AdminStudio Concepts	2-1
	Windows Installer	2-3
	Traditional Setups vs. Windows Installer Setups	2-3
	Benefits of Windows Installer	2-3
	Feature Advertisement	2-4
	User-Accessible Information	2-4
	Componentized Installation	2-4
	Rollback	2-4
	Single File	2-4
	Merge Modules	2-4
	Open Architecture	2-5
	Native to Current Windows Platforms	2-5
	Windows 2000 Logo Compliance	2-5
	Self-Repair	2-5
	Clean Uninstallation	2-5
	Predeployment Process	2-5
	AdminStudio Interface	2-6
	Repackaging	2-7
	Authoring and Modifying Installations	2-7
	Typical Elements of a Setup	2-7
	Perform File Transfer	2-7
	Display a User Interface	2-8
	Create Shortcuts	2-8
	Register File Associations	2-8
	Register COM, COM+, and DCOM Files	2-8
	Register Product for Uninstallation	2-8
	Installation Customization	2-9
	Installation Package	2-10

MSI and MST Files	2-10
Transform Files	2-10
Validation	2-11
Customization Steps	2-11
Conflict Identification and Resolution	2-12
Application Conflict Evaluators (ACEs)	2-13
ACE Types	2-13
ACE Descriptions	2-13

Chapter 3 InstallShield AdminStudio Tutorial **3-1**

Preparation	3-3
Launching Repackager Remotely	3-3
Repackaging a Traditional Installation	3-4
Modifying and Building a Windows Installer Installation	3-11
Opening the .inc File	3-11
Modifying the Project	3-15
Building the Release	3-16
Customizing an Installation Using Tuner	3-18
Opening the Package and Creating a Transform	3-18
Prevalidation	3-19
Customization	3-20
Specifying Default Destination and Organization Information	3-20
Configuring Features	3-21
Adding Files to the Installation	3-22
Creating Shortcuts	3-23
Configuring Windows 2000 Add/Remove Programs Settings	3-24
Postvalidation	3-25
Packaging	3-26
Further Customizations	3-29
Identifying and Resolving Conflicts	3-30
Opening ConflictSolver and Accessing the Database	3-30
Configuring ConflictSolver Options	3-31
Validating the Package and Transform	3-32
Identifying Conflicts	3-34



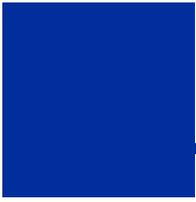
Resolving Conflicts 3-37
 Checking for Conflicts a Second Time 3-40
Importing the File 3-42
Conclusion 3-45

Chapter 4 Frequently Asked Questions 4-1

AdminStudio FAQ 4-3
 How Do I Launch the AdminStudio Features? 4-3
 How Do I Check for AdminStudio Updates? 4-3
 Why Does AdminStudio Look Different Than the Last Time I Used It? 4-3
Repackager FAQ 4-4
 Why Do People Use a Repackager? 4-4
 What Can I Do with a Repackaged Setup? 4-4
 Why is a “Clean” System Important for Repackaging? 4-4
 How Do I Repackage a Non-Windows Installer Setup? 4-4
 How Do I Repackage a Windows Installer Setup? 4-4
 How Can I Speed Up Repackaging? 4-5
 How Can I Use the Same Repackager Settings Again? 4-5
Author FAQ 4-5
 When Do I Need to Use Author? 4-5
 How Do I Import and Customize a Repackaged Application? 4-5
 How Do I Import and Customize an Existing MSI File? 4-6
 How Can I Ensure the Target Machine Has the Windows Installer Service? 4-6
 How Do I Specify Product or Component Default Destinations? 4-6
 How Do I Make IDE Changes Take Effect When I Run My Setup? 4-7
 How Do I Display a License Agreement? 4-7
 How Do I Create Shortcuts? 4-7
 How Do I Associate File Types with My Program? 4-8
 How Do I Launch an External Application from My Setup Project? 4-8
 How Do I Conditionally Install My Setup Based on the Target’s Operating System? 4-8
Tuner FAQ 4-9
 When Do I Use Tuner vs. Author? 4-9
 How Do I Add a File to a Windows Installer Package? 4-9
 How Do I Prevalidate a Windows Installer Package? 4-9



What Should I Do If MSI Prevalidation Fails?	4-10
How Do I Create a Setup.exe File for My Package and Transform?	4-10
How Do I Prevent a Feature from Displaying During Custom Installation?	4-11
ConflictSolver FAQ	4-11
What Types of Conflicts are Detected?	4-11
When Should I Import Merge Modules?	4-11
How Do I Import a Package into the Conflict Database?	4-11
How Do I Import a Merge Module into the Conflict Database?	4-12
How Do I Identify Conflicts?	4-12
Identifying Conflicts with an External Package	4-13
Identifying Conflicts Between Existing Packages	4-13
How Do I Validate the Package?	4-13
How Do I Change Which Conflicts Are Checked?	4-14
How Do I Automatically Resolve Conflicts?	4-14
Glossary	G-1
Index	I-1





Welcome to InstallShield AdminStudio

Table of Contents

Welcome to InstallShield AdminStudio! You have purchased the quickest and most comprehensive installation predeployment solution available. This chapter describes some of the many features and benefits of InstallShield AdminStudio, and provides information on how to get help when you need it.

AdminStudio Features and Benefits	xv
InstallShield AdminStudio Benefits	xv
InstallShield AdminStudio Features	xvi
Repackaging	xvi
Authoring	xvi
Tuning.	xvii
Solving Conflicts.	xvii
About InstallShield Software Corporation	xviii
Fast Facts about InstallShield.	xviii
InstallShield's Products	xix
Installation Development.	xix
InstallShield Developer	xix
InstallShield Professional.	xix
InstallShield Express	xx
InstallShield Multi-Platform Edition.	xx
Administrator Solutions	xx
InstallShield AdminStudio	xxi
Multimedia Authoring	xxi
DemoShield	xxi
Technical Support Resources	xxii
Online Help.	xxii
Help Library.	xxii
Context-Sensitive Help	xxii
Web-Based Help	xxii
InstallShield Knowledge Base	xxii
InstallShield Newsgroups	xxiii
InstallShield Customer Care—	
Technical Support	xxiii
Support Programs	xxiii
Contacting InstallShield Customer Care . .	xxiv



InstallShield Professional Services xxv
InstallShield Consulting Services xxv
InstallShield Trainingxxvi



AdminStudio Features and Benefits

InstallShield, the most trusted name in Windows setup authoring for more than ten years, now introduces the InstallShield AdminStudio Version 2.0. AdminStudio was developed specifically for systems administrators, desktop engineers, and corporate IT project teams, to ease the complex and time consuming tasks involved in rolling out an application. AdminStudio gives you the best way to repackage, customize, and resolve conflicts with your applications prior to deployment, while maintaining complete standards compliance. That means you won't run into trouble when it comes time to update or apply a patch to your software. AdminStudio provides an easy-to-use wizard-driven interface to quickly navigate repackaging of complex applications. AdminStudio adheres to Microsoft Best Practices and follows the guidelines of Logo Compliance. Plus, now you can leverage the value added functions of the Windows Installer Service such as application repair, advertising (install-on-demand), rollback, and clean uninstallation-which translate into cost of ownership savings for your organization and fewer headaches for you.

AdminStudio allows you to easily customize installations without having to edit Windows Installer database tables directly, or to even write a single line of script. Customizations such as-changing registry settings, adding templates, configuring ODBC, or the creation of a response file for silent installs-are all easily made through an intuitive interface that removes much of the complexity of working with a typical setup.

And that's just the start. Now you can see how well an application "plays with others" before you deploy it. AdminStudio's ConflictSolver compares the contents of multiple Windows Installer packages to insure their compatibility. Should conflicts be found, the Resolution wizard will report the issue, and if you choose, automatically resolve the issue. This gives you a complete solution to all of the deployment issues you face, prior to broadcasting the installation package.

InstallShield AdminStudio Benefits

- Leverages the power and reliability of InstallShield, with features designed specifically for corporate IT professionals.
- Hides the complexities of migrating applications to a Windows 2000 standard installation package.
- Decreases your time to deployment.
- Rapidly builds silent installations.
- Easily modifies any part of your silent installation without having to repackage.
- Eliminates single target packages, with custom installations that work on all the latest versions of Windows.
- Identifies and resolves conflicts prior to distribution, improving reliability of production desktop environments.
- Leverages investment in existing deployment solutions such as SMS, Tivoli, Novadigm, and Marimba.
- Completes the promise of Windows 2000 and Active Directory's reduction in total ownership costs.



InstallShield AdminStudio Features

AdminStudio is a collection of capabilities that enable you to perform important tasks in the pre-deployment process including repackaging, authoring, customizing or tuning, conflict solving. All of these capabilities give you completely standards-compliant Windows Installer setups that carry on InstallShield's more than ten-year tradition of quietly enabling users to reach their objectives.

Repackaging

Collect before and after snapshots of an application's installation. The resulting delta (or group of changes) serves as the initial input for your new custom setup package. Repackaging captures all critical system changes such as registry modifications, shortcuts, file additions, environment variables, driver changes, and NT services. You can use the Repackager to migrate applications to the new Windows 2000 standard and to quickly build a silent installation.

Key Functionality

- **Fully Customizable Exclusion List**—Use a wizard to select standard setting or completely customize the system areas scanned. Exclude unnecessary directories, files, registry entries, or shortcuts so only required data is included in the resulting image.
- **Superset Installers**—Repackage multiple installations into a single Windows Installer package.
- **Output Preview**—View Repackager output in simple text files. Snapshot changes can be viewed, edited, or printed prior to final package generation.
- **Flexible Scanning Process**—Modify how the scan is performed between the initial and final snapshot. Stop for editing between the scans or move through all steps of the process automatically.
- **Application and System Level Changes**—Create installations of both application and system level modifications.

Authoring

Directly generate a setup package, or build on your Repackager output to add conditions or combine packages together. All of the traditional InstallShield capabilities are at your fingertips, including debugging, validation, third-party redistributables, and ensuring that all of an application's required files are bundled in your package. Yet you work in a development environment that's tailored to your needs as an IT Professional. You still have access to InstallScript and other advanced capabilities, but everything you need is wrapped in a point and click interface.

Key Functionality

- **Installation Integrity**—Validate and debug your installation, and most importantly, check your installation to ensure that you have all the necessary underlying files included in your package. Plus the Best Practices Wizard alerts



you to deviations from setup guidelines and gives you recommendations for changes.

- **InstallScript Custom Actions**—The Custom Action Wizard helps you build custom actions from InstallScript to include executables, dynamic-link libraries, and perform any of 300 installation-related functions that you might need.
- **Reusable Components**—Create project components for reuse across multiple application packages or use one of over 40 existing third-party technology merge modules that ship with the product.
- **Custom Installation Dialogs**—Easily create custom dialogs or modify an existing dialog panel.
- **Installation Conditions**—Add conditions that check for previously installed files and applications, to keep from potentially overwriting something important.

Tuning

Once you've repackaged a traditional setup, or if you already have a Windows Installer package, AdminStudio's installation Tuner gives you the fastest way to make changes to the existing package. Your changes are saved in a separate file that modifies the base setup package during the installation.

Key Functionality

- **Silent Installations**—Once you have a Windows Installer package, tuning is the fastest way to build silent installs, or to make other modifications.
- **Drag and Drop Interface**—Visually add or remove files, registry entries, and shortcuts by simply dragging and dropping them into the Outlook style interface.
- **Registry Import**—Import existing REG and INI files into your setup package.
- **Advanced Control**—View and edit the underlying tables of the Windows Installer package.
- **SMS Support**—Build packages for deployment via Microsoft's Systems Management Server (SMS).

Solving Conflicts

Eliminate trial and error deployments. Use ConflictSolver to check for potential conflicts between your setup packages, before you deploy. ConflictSolver resolves many resource conflicts that could affect proper application behavior. This process dramatically reduces the risks of introducing a new application into your production environment.



Key Functionality

- **Broad Conflict Type Management**—Test for variety of different conflicts, including registry entries, INI Files, ODBC Resources, NT Services, Product Code/Package Code/Product Version, file extensions, and many more.
- **Configurable Conflict Evaluation**—Run conflicts checks between an application and any set of applications already stored in the conflict database.
- **Customizable Conflict and Resolution Rules**—Add new conflict rules and resolutions based on your own requirements and policies.
- **Quick Fixes**—Automatically resolve identified conflicts.
- **Advanced Reporting**—Print detailed reports on the contents of the database or known conflicts. Use the standard reports or build custom ones.
- **Database Options**—Use an Access 2000 or SQL Server 7.0 database to store your application images.

About InstallShield Software Corporation

Founded in 1987 by Viresh Bhatia, CEO, and Rick Harold, CTO, InstallShield is most widely recognized for standardizing the way traditional software is installed on over 250 million PCs around the globe. Today, the company continues evolving that standard, providing Independent Software Vendors, Corporate Developers, Systems Administrators, and Digital Content Producers with cutting-edge technologies, products, training and consulting services.

InstallShield products enable the distribution, management, and consumption of software and digital content via the Internet and throughout the corporate enterprise. Our customers span every industry, from multi-national corporations to small businesses. Despite our industry-leading position and global presence, we maintain an entrepreneurial culture to address the full spectrum of software installation and deployment needs.

Fast Facts about InstallShield

- Co-founded in 1987 by Viresh Bhatia and Rick Harold
- Corporate headquarters in Schaumburg, Ill. (outside of Chicago)
- Four time member of the prestigious “Inc. 500,” Inc. Magazine’s listing of America’s fastest growing private companies
- Recognized as one of greater Chicagoland’s fastest growing technology companies by Deloitte & Touche “Fast 50” Program
- Ranked by Media Metrix, Inc., as the sixth-most-frequently owned and eighth-most-frequently used software



InstallShield's Products

In 1990, Microsoft introduced Windows 3.0, which quickly became the standard for graphical operating systems. InstallShield Software Corporation (then named the Stirling Group) was among the first in the marketplace to respond, offering the Shield series of tools for Windows application development. From this series, InstallShield and DemoShield® emerged as market leaders, filling the need for Windows installation and demonstration development.

InstallShield Software Corporation has continued to lead all competitors through the years, providing products and services that meet and exceed industry demands for software distribution in a number of varied categories.

Installation Development

The first. The best. The industry standard. InstallShield Software Corporation is the undisputed leader for installation development.

InstallShield Developer

The best just got even better! Meet InstallShield's new Flagship solution: InstallShield Developer. InstallShield Developer is the most flexible and powerful installation-authoring solution, offering the greatest level of project control. It combines the power of InstallScript™ with Windows Installer features such as autorepair and rollback. Create and manage projects just the way you want using features such as source code control integration, automated authoring and build functions, and patch creation. Developer also enables extensive customization of the end-user experience. Use the graphical dialog editor, digital signature and password options, and billboard support to tailor the installation experience to your customers' needs. Over 20 task-based wizards makes combines ease of use with Developer's robust functionality. And Developer also keeps you current while preparing for the future with support for Windows XP, .NET, and 64-bit applications. Complete standard installations quickly and meet unique customization requirements as they emerge. No challenge is too great for InstallShield Developer.

InstallShield Professional

What you need is a powerhouse installation-authoring solution with the tools that help you do your job better than you've ever done it before. A solution that will give you total control over even the most intimidating installation endeavors. Simplify complex processes. And shrink development cycles. You need a solution that can help you deliver a more flexible end-user experience—and you need it now.

InstallShield Professional is a surefire installation-authoring solution that will propel you to new heights of productivity and success. In fact, it's the developer's number one choice for reliable software deployment projects, from the amazingly complex to the most routine. Professional lets you create and distribute reusable pieces of installation projects, compresses the time needed for installations, and makes setup-authoring a more flexible, reliable process for all your flagship and mission-critical applications.



If you were looking for the world's number one choice for standard installation development, you've found it. It's InstallShield Professional.

InstallShield Express

You require a straightforward installation that you can create quickly and deliver with confidence, assured it will install correctly every time. With InstallShield Express, you can achieve new levels of speed and productivity that will let you go from project to setup in less than a day!

InstallShield Express is a visual installation development solution that lets you quickly point and click your way to creating installations—that means no required scripting and no learning curve. With built-in support for many third-party technologies and an easy-to-use development environment, InstallShield Express 3.5 is the most economical solution for developers with short installation development cycles. It puts you on the fast track to building installations—right out of the box, in minutes—for applications targeting any flavor of Windows.

InstallShield Multi-Platform Edition

Recent advances in cross-platform software development have allowed InstallShield to dramatically improve the way software installations are managed and executed on many non-Windows platforms. These changes bring the familiar InstallShield user experience, pioneered on Windows over 10 years ago, to UNIX®, Linux®, and OS/400® platforms.

Full native support for each operating system's model of packaging and deploying software on several platforms used to mean that developers had to create and maintain a different process for each OS supported—with resulting higher costs, longer release cycles, and potentially reduced quality.

The InstallShield solution offers a single, unified technology that addresses the distinct requirements of each operating system while providing an elegant and consistent user experience. InstallShield Multi-Platform is the result of a collaborative effort by InstallShield and leading platform vendors to address the needs of organizations that target more than one operating system. InstallShield's goal in this effort was simple: allow developers to define and maintain a single packaging process without burdening them with platform-specific details. In order to succeed, InstallShield had to lower packaging and deployment costs, improve quality, and restore a sense of simplicity to installing software. The result? InstallShield Multi-Platform speaks for itself.

Administrator Solutions

Software distribution and InstallShield go hand in hand. So when your Systems Administrators take on the challenge of pre-deployment tasks like repackaging, authoring, customizing software distribution within the corporate network, and conflict solving, make sure they use solutions from the same expert that your developers trust to create industry-standard installations.



InstallShield AdminStudio

InstallShield AdminStudio is the easiest and most reliable solution for repackaging and customizing applications, as well as finding and resolving potential conflicts between applications. It is designed specifically to meet the application pre-deployment needs of System Administrators, using wizards and a checklist interface to guide the process.

Using InstallShield AdminStudio, you can:

- **Repackage Traditional Setups**—Create new setups from traditional, non-Windows Installer setups. System snapshots taken before and after the application is installed and configured are used to create a new application image.
- **Author Setups/Enhance Repackaged Setups**—Create or change a Windows Installer setup directly. This option is recommended for authoring your own setups, or for making additional adjustments after the Repackaging process and prior to customization for various user group needs.
- **Customize Setups**—Create one or more custom installations from the same Windows Installer package, without altering the underlying setup (per Microsoft guidelines).
- **Identify and Resolve Conflicts**—Check for potential conflicts between Windows Installer setup packages and resolve issues that could affect proper application behavior.

Multimedia Authoring

InstallShield Software Corporation pioneered dynamic, software-specific demonstration technology. Multimedia Authoring, Technical Publications, Development, Sales, or Marketing departments can now target their audience and show-not just tell-what their software can do.

DemoShield

Capture your audience's attention with stylish demos that inform, entertain and sell! DemoShield lets any PC user rapidly develop professional multimedia demos, sales presentations, CD browsers, tutorials, and more.

Use DemoShield to guide your customers through the key benefits and powerful features of your product. DemoShield's easy-to-use, point-and-click authoring environment helps you add creativity to your content and clarify your user's next steps. Plus, DemoShield's visual impact greatly enhances all of your strategic messages, from marketing information and setup, to documentation and training. What's more, only DemoShield allows you to create professional demos without having to learn complex and unwieldy media tools.



Technical Support Resources

InstallShield's Product Support Services team hosts a wealth of options for you to find your solution. Online resources such as our Knowledge Base provide quick access to years of service at no cost, while service plans give you more robust support with our expert engineers. Take a look at your options below and see which solution is right for you.

Online Help

When you have questions about InstallShield, first consult the online InstallShield Help Library. Help is available both from the installation development environment's (IDE) Help menu and directly from certain individual IDE elements.

The help resources listed below are available from within InstallShield AdminStudio.

Help Library

The Help Library is your complete user's guide for InstallShield AdminStudio. This library contains reference materials that cover every aspect of your setup development.

Context-Sensitive Help

Online help is available for every dialog box, wizard panel, and view in the InstallShield IDE. Press the F1 key or click the Help button to view the IDE Reference help for the current dialog box.

Web-Based Help

Consult the InstallShield Web site (<http://www.installshield.com>) and click on Support for immediate access to an extensive selection of useful information. The site is a comprehensive resource of free support materials including the following:

- Technical information and product news
- Sample projects
- Frequently asked questions
- White papers
- Documentation updates
- Maintenance releases and updates for registered software
- Information about consulting, support, and professional services

This information is available to you 24 hours a day, 7 days a week.

InstallShield Knowledge Base

The InstallShield Knowledge Base is located in the Support section of our Web site (<http://www.installshield.com>). It contains answers to many commonly asked questions and includes new information about InstallShield that may not appear in the documentation. You can use the Knowledge Base search engine to search articles by phrases, numbers, platforms, and version.



InstallShield Newsgroups

The newsgroups are another excellent resource when you have questions about InstallShield AdminStudio. In these newsgroups, developers like you share tips and ideas and help each other get the most out of InstallShield AdminStudio. Visit the InstallShield and InstallShield AdminStudio newsgroups in the Support section of our Web site, available at by clicking on the Support link at <http://www.installshield.com>.

InstallShield Customer Care—Technical Support

Take advantage of InstallShield's comprehensive resources that help you find the answers you need. Connect to <http://www.installshield.com> and select the Support link for information about our support options if you have not already contracted for support.

Support Programs

In addition to the award-winning resources of our support Web site, InstallShield offers a number of support plans that are fine-tuned to specifically meet your needs.

- **Public Resources**—Any time, day or night, access our knowledge base, newsgroups, and the sample gallery for outstanding support resources.
- **Bronze Support Plan**—Unlimited support for requests submitted through our Web site.
- **Silver Support Plan**—Unlimited support for requests submitted by phone and through our Web site.
- **Gold Support Plan**—A dedicated phone number and Web-form give you a priority response and preferential queuing to dedicated engineers. Additionally, this plan allows you to participate in our Beta Program.
- **Platinum Corporate Support Plan**—Get personalized service that is fine-tuned to meet the needs of your organization. In addition to the services provided under the Gold service plan, Platinum support assigns a Customer Care manager to oversee all of your technical issues. The Customer Care manager is responsible for all issues submitted by your organization and holds optional conference calls every month regarding status and resolution open issues (if any). This plan also provides the option of having Web-based demonstrations and product assistance through the WebEx online meeting technology.



Services	Public	Bronze	Silver	Gold	Platinum
24 x 7 Knowledge Base Support	×	×	×	×	×
24 x 7 Newsgroup Access	×	×	×	×	×
Optimized Code and Sample Gallery	×	×	×	×	×
Direct Engineer Contact		×	×	×	×
Web-Based Email Support		×	×	×	×
Telephone Support			×	×	×
Priority Involvement in Beta Programs				×	×
Priority Queuing Through Dedicated Channels				×	×
Product Assistance Using WebEx				×	×
Access to Pre-Beta Products					×
Status Reports Option					×
Status Conference Calls Option					×
Personalized Escalation Contact Option					×

Contacting InstallShield Customer Care

Whenever you contact technical support, our Support Engineers will need the following information:

- Your full name and the full name of the registered user, if different.
- Your street address, telephone number, and email address.
- Your InstallShield AdminStudio serial number.
- The full InstallShield AdminStudio version number. To find this number, select About InstallShield AdminStudio from the Help menu.
- The system information for the computer on which the problem occurred, including the make and model of the machine and video driver, the operating system, the amount of memory and system resources, any relevant devices or peripherals, and any system DLLs you are using.
- A detailed description of the problem. Describe any error messages exactly as they appear. Please list all of the steps and conditions that led to the problem.



Our Support Engineers are available via the World Wide Web or phone:

Phone: (Mon–Fri 8:00am–5:30pm CST)	1-847-413-2896
WWW:	http://www.installshield.com

Support is limited to installation assistance and answers to general InstallShield usage questions. Responses to questions relating to operating system functionality, development environments, and the operation of various Windows technologies are not included.

InstallShield Professional Services

InstallShield Professional Services is made up of InstallShield Consulting Services and InstallShield Training.

InstallShield Consulting Services

No One Knows More about Installations than We Do

Bottom line: we've solved more than half-a-million man-hours (250 years) worth of difficult installation challenges.

As the industry's recognized experts, we have the unique capabilities necessary to ensure a successful first experience for your customers, keep difficult projects from impacting your ship dates, and lower your technical support costs.

We know the technology challenges and we own the solutions. Our teams, with the support of InstallShield's larger technical base of product developers, will help you determine and implement the best solution to any and every installation and deployment challenge.

Since 1990, we've been helping the industry with important projects, offering only the expertise an InstallShield Consulting Services can provide. Many of our consultants are former InstallShield developers, and all have thorough training and expertise. So with InstallShield Consulting Services, you can be sure your projects are being handled by the best possible resource. Contact us in the following ways:

North America, Central and South America, Asia (Except Japan), and the Pacific Regions

Phone: (Mon–Fri 8:00am–5:00pm CST)	1-847-619-7028
WWW:	http://www.installshield.com

Europe, Middle East, and Africa

Phone: (Mon–Fri 8:00am–6:00pm GMT)	+44 (0) 117 903 0650
WWW:	http://www.installshield.com



InstallShield Training

It's More than a Training Class

After participating in our courses, you'll understand the advanced features of your product and know how to create solutions for complex software distribution and deployment requirements in less time.

The time you spend in an InstallShield Training course is focused on helping you learn every technique and feature available in your InstallShield software solution so that you deploy faster, author with more custom skill and control, and meet the specific, strategic requirements of your organization.

We want to help your company deliver a more robust installation with more efficient development and quicker time-to-market for your important applications.

And we want to help you advance your skill set, learn valuable techniques from an industry leader, and share your experiences with fellow attendees and technical InstallShield staff.

Our goal is to provide educational excellence and “tell-a-friend” value. Every time.

For more information about InstallShield Training, contact us in the following ways:

North America, Central and South America, Asia (Except Japan), and the Pacific Regions

Phone: 1-888-560-6933
(Mon–Fri 8:00am–5:00pm CST)

WWW: <http://www.installshield.com>

Europe, Middle East, and Africa

Phone: +44 (0) 117 903 0650
(Mon–Fri 8:00am–6:00pm GMT)

WWW: <http://www.installshield.com>

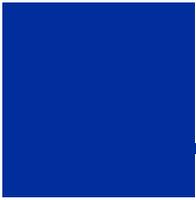


Installing AdminStudio

Table of Contents

Document Conventions	1-3
System Requirements	1-3
Installation Procedure	1-4
Uninstallation Information	1-4
Register Your Software	1-4

This chapter contains system requirements for InstallShield AdminStudio and information about the installation and uninstallation procedures. It instructs you on how to register InstallShield AdminStudio, and provides you with documentation conventions for this Getting Started Guide.





Document Conventions

To enhance readability, certain text elements are specially identified as follows:

Italic text denotes points of emphasis.

Bold text denotes commands, menu items, selectable options, Internet and email addresses.

Monospace text denotes command line input, file listings, scripts, and path specifications. It can appear in a shaded area, depending on its length.

Procedures will be numbered and be accompanied by the step graphic in the margin. For example:

To open a file, do the following:

1. Select Open from the File menu.
2. Navigate the file you want to open and select it.
3. Click OK.



Certain pieces of information should stand out in the endless sea of documentation. The note image appears in the margin for this type of information, and the note text is set off by horizontal bars.



Tips are also set off by horizontal bars, and are accompanied by the tip graphic in the margin. Tips are used when bits of information, such as time saving suggestions, are presented.



Warnings about potential pitfalls that you should avoid appear within lines and accompanied by the warning graphic in the margin. These warnings are set off by horizontal bars from the surrounding text.



System Requirements

- Microsoft Windows NT 4.0 (Intel) operating system with SP3 installed, Windows 2000, Windows Me, Windows 98, or Windows 95
- Administrative privileges on the system if using Windows NT 4.0 or Windows 2000
- Microsoft Internet Explorer (IE) 4.01 (SP2) or higher if running on NT 4.0, Windows 98, or Windows 95
- Pentium-class PC (300 MHz or higher recommended)
- 64 megabytes (MB) of RAM (128 MB recommended)
- Super VGA monitor with the following settings:



- Screen resolution set to at least 800 x 600
- Small fonts selected
- Disk (free) space required:
 - CD installation: 400 MB before installation and 200 MB after install is complete
 - Web download: 600 MB before installation and 400 MB after install is complete
 - If you have a version of InstallShield Professional—Windows Installer Edition already installed on your system, you must uninstall it prior to AdminStudio installation.

Installation Procedure



To install InstallShield AdminStudio, do the following:

1. Launch the AdminStudio.exe from the CD.
2. When the InstallShield wizard launches, provide the information required to install the application to your hard drive.

Uninstallation Information



To uninstall InstallShield AdminStudio, do the following:

1. Open the Add/Remove Programs dialog from the Windows Control Panel.
2. Select InstallShield AdminStudio and click Remove.

Register Your Software

To be eligible for our Support Programs, be sure to register your software. Registering also ensures that you receive the latest information about product releases and upgrades.

Register your InstallShield software in one of the following ways:

- Following installation, use the Registration form to immediately register your product with InstallShield.
- Complete a registration form on InstallShield's registration Web page at <http://support.installshield.com/register>.
- Complete and send in the registration card included in the InstallShield AdminStudio box.

Following registration, record your serial number in the space provided on the second page of this guide. If you need to contact InstallShield Customer Care, they will need this serial number to verify your product is registered.



InstallShield AdminStudio Concepts

Table of Contents

InstallShield AdminStudio is designed to assist administrators in the predeployment process. With this goal in mind, AdminStudio is separated into four features that each cover a large segment of predeployment, enabling repackaging, authoring, customization, and conflict identification & resolution. This chapter discusses these issues, in addition to providing information about the Windows Installer service.

Windows Installer	2-3
Traditional Setups vs.	
Windows Installer Setups	2-3
Benefits of Windows Installer	2-3
Feature Advertisement	2-4
User-Accessible Information	2-4
Componentized Installation	2-4
Rollback	2-4
Single File	2-4
Merge Modules	2-4
Open Architecture	2-5
Native to Current Windows Platforms	2-5
Windows 2000 Logo Compliance	2-5
Self-Repair	2-5
Clean Uninstallation	2-5
Predeployment Process	2-5
AdminStudio Interface	2-6
Repackaging	2-7
Authoring and Modifying Installations	2-7
Typical Elements of a Setup	2-7
Perform File Transfer	2-7
Display a User Interface	2-8
Create Shortcuts	2-8
Register File Associations	2-8
Register COM, COM+, and DCOM Files	2-8
Register Product for Uninstallation	2-8
Installation Customization	2-9
Installation Package	2-10
MSI and MST Files	2-10



- Transform Files 2-10
- Validation 2-11
- Customization Steps 2-11
- Conflict Identification and Resolution 2-12
 - Application Conflict Evaluators (ACEs) 2-13
 - ACE Types 2-13
 - ACE Descriptions 2-13



Windows Installer

The new Windows® Installer service introduces a faster, more reliable method of application installation. Standardized application installation, enhanced management, and easy application customization are some of the many benefits Windows Installer provides. Another important functionality of Windows Installer is its ability to diagnose and repair configuration problems when the application is opened. An integral part of the Windows 2000 operating system, Windows Installer provides a standard format for component management as well as an interface for managing applications and system tools. Windows Installer is available as a service pack upgrade for Windows 95/98 and Windows NT 4.0 operating systems.

Windows Installer delivers this enhanced performance through a combination of application advertisement, on-demand installation, and related technologies. Advertising applications enables software to be potentially available to many users. On-demand installation allows the seamless activation of an advertised feature or product without requiring the user to exit from the current program in order to run a separate setup program, a process that often involves restarting your computer.

This service gathers installation information from .msi files. An .msi file is actually a database that contains several tables. Each of these tables contains instructions and setup information. Windows Installer creates a script of actions from this relational database. These actions are then executed in the order in which the installer encounters them. An .msi file is the compiled output of the authoring tool project (.ism) files.

Traditional Setups vs. Windows Installer Setups

Although both traditional setups and Windows Installer setups achieve the same goal, the means to the end are very different. Traditional setups rely upon a custom-built engine. Some companies buy those engines from software vendors like InstallShield, promoting a semblance of a standard. Other companies build their own engines, which interact with the operating system in an entirely different way. With all of these different setup engines floating around, it is difficult for the operating system to maintain a reliable reference to all the files that are installed and how they must be uninstalled. With Windows Installer, Microsoft has created an industry standard for installation and uninstallation. According to Windows 2000 logo requirements, all installations must make use of Windows Installer. This standardization allows for greater stability, better maintenance, and improved management of applications installed on the system.

Benefits of Windows Installer

Besides being a standardized method of handling installation and uninstallation, Windows Installer brings many other benefits, from advertisement to rollback. The following section outlines a few of the more exciting features that Windows Installer offers.



Feature Advertisement

Windows Installer allows you to advertise features. Advertised features can appear installed, but the actual files are not installed until an application or a user requests the feature—for example, by clicking an advertised shortcut. For example, you may advertise your help system so that no hard disk space is taken up until the help is first accessed by the user. At that time the installer installs the necessary files and launches the help system.

User-Accessible Information

The Add/Remove Programs applet in Windows 2000 provides information on installed products. This information includes links to Web sites that offer support for the product and sites that host product updates. The software publisher and product version number are also displayed here. InstallShield provides full support of these tables, making it easier for you to pass information on to your end users.

Componentized Installation

Your product can be installed on a feature-by-feature basis. For example, if you have split your setup into two features, `Program_Files` and `Help_Files`, your end users can choose to install only the program files and not the help files. Later, the customer can uninstall just `Program_Files` or install `Help_Files`. This aspect of Windows Installer is especially useful on systems with limited hard drive space.

Rollback

Traditional setups definitely get the job done. They usually work as expected, leaving the end user's system in a stable, if somewhat altered, state. However, when these setups fail, there is little done to protect the system from harm. In traditional setups, necessary registry entries or files can be deleted without maintaining a backup of the original configuration. The rollback capabilities in Windows Installer ensure that the system is returned to its previous state if the setup fails for any reason. During a setup, Windows Installer identifies which files will be changed and makes a backup copy of those files. If the setup fails, those files are returned to their previous state, leaving the system unchanged.

Single File

All of your setup files can be streamed into a single `.msi` file for easy distribution. You do not need to ship your files separate from the installer package, which makes deployment much less complicated. A further option available in InstallShield Professional is to compress all of the setup files into a single self-extracting executable, `Setup.exe`.

Merge Modules

Windows Installer supports the use of merge modules, which allow you to add redistributable components to your setup without having to recreate that aspect of the setup every time you want to use it. For example, if your application requires DAO 3.6 in order to function properly, you can add the DAO 3.6 merge module to your setup.



Open Architecture

Since the .msi file adheres to a standard format, any developer or network administrator can view the database and customize it for their enterprise's needs.

Native to Current Windows Platforms

The Windows Installer service is native to Windows 2000 and Windows Me. You can also install it on Windows 95, Windows 98, and Windows NT 4.0.

Windows 2000 Logo Compliance

Use of Windows Installer is required in order to achieve the Windows 2000 logo. This is just one factor that goes into the overall list of compliance items, but it is a necessary part. The authoring takes you one step closer with Setup Best Practices. Setup Best Practices are a list of guidelines for your setup. These Best Practices help you to comply with the logo requirements.

Self-Repair

By adding a few lines of code to your program, you can protect against missing and corrupted files. If, after installation, a file becomes corrupted or is accidentally deleted, the Windows Installer service can automatically reinstall that file.

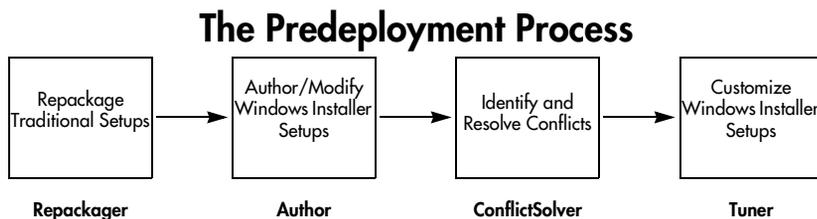
Clean Uninstallation

Since the Windows Installer service maintains a database of all the files that are installed and all the changes that are made by the service, uninstalling programs is much more reliable. Shared files, such as DLLs, are automatically kept track of and will not be removed if still required by another program.

Predeployment Process

The predeployment process can consist of up to four steps in which you prepare a Windows Installer installation for distribution. Depending on the source of Windows Installer package, and the specific needs of organizations, the inclusion and order of these steps may vary.

1. Repackaging traditional setups.
2. Authoring or modifying Windows Installer-based setups.
3. Resolving conflicts between Windows Installer setups.
4. Customizing Windows Installer setups.





InstallShield AdminStudio consists of four features to assist in the predeployment process:

- **Repackager:** used for repackaging traditional installations into a format which can be read by Author. You can also use it to combine multiple traditional installations into a single installation.
- **Author:** used to author new or modify existing Windows Installer packages. You can also use it to import traditional installations converted by Repackager and create Windows Installer packages from them.
- **Tuner:** used for customization of Windows Installer packages via a transform file. Also used for validating that the Windows Installer setup meets known guidelines to ensure all Windows Installer functionality works as expected. Customization includes the addition, modification, or removal of several key elements of the Windows Installer installation, including files, INI files, short-cuts, registry entries, ODBC resources, NT Services, and Features.
- **ConflictSolver:** used to identify potential conflicts between Windows Installer packages. In many cases, ConflictSolver can resolve these issues, ensuring interoperability between Windows Installer setups.

Each of these features is explained in detail in the following sections.

AdminStudio Interface

The InstallShield AdminStudio interface includes a collection of live Web pages that are updated as new information becomes available.

Figure 2-1.
The AdminStudio Interface.



When you launch AdminStudio, the DigitalWizard Update Service launches in the background to check the InstallShield Web site for updates to the pages in



AdminStudio. If it finds an update, the DigitalWizard Update Service downloads the update in the background, and then terminates. The next time AdminStudio is launched, the new Web pages replace the existing ones. If no Web connection is available, or if no updates exist, the download does not occur.

Repackaging

There are two primary reasons administrators need repackaging capabilities:

- To capture system changes as a Windows Installer package
- To combine multiple installations into a single installation, including some degree of customization
- To convert a traditional installation into a Windows Installer-based installation, to take advantage of Windows Installer functionality, such as application repair

The Repackager performs both these tasks, producing a Windows Installer package that contain multiple traditional installations.

The basic functionality behind Repackager is a baseline analysis of a target system, and then a post-analysis following one or more traditional installations. The Repackager determines the additions or changes made to files, INI files, shortcuts, and the registry, and then collects this data into a format that Author can open and convert into a Windows Installer package.

It is extremely important that repackaging is performed on a “clean” computer so all pertinent additions and changes are recorded.



Authoring and Modifying Installations

Author performs three important tasks:

- Converting repackaged traditional installations into Windows Installer packages
- Modifying existing Windows Installer packages
- Authoring new Windows Installer packages

In most cases, administrators do not author packages from scratch. Rather, they modify them to meet their needs, whether they originate as a result of a repackaged traditional installation, or have been authored as a new installation.

Typical Elements of a Setup

Perform File Transfer

File transfer involves copying files from the source medium, such as a CD-ROM or a floppy disk, to a local drive on the end user’s machine. Depending on the configuration the end user chooses, all or only some of the files may be transferred to the



local disk. During file transfer, a setup can display billboards that provide product information such as new features or usability tips. A status bar may also be displayed to show the progress of the file transfer process.

Display a User Interface

The user interface of a setup provides information and setup configuration choices to the end user. Through the user interface an end user can choose to install only part of a product, choose to leave some files on the source medium, view a license agreement, or provide information to the installer that may be necessary to ensure the proper configuration of the installation. The user interface can be customized to meet whatever needs you may have. For example, you can prompt a user for a serial number before starting the installation if you would like to protect your software against illegal use.

Create Shortcuts

Shortcuts are links to files and applications that you can create on the end user's system during a setup. Shortcuts are often placed on the desktop or the Start menu of the target machine to provide quick and easy access to a program or its files.

Register File Associations

If your product supports distinct file types, you must register those file types on the end user's system. For example, InstallShield project files have the extension .ism. In order for you to double-click an .ism file and have it opened in the IDE, we had to create a file association for .ism files in your registry.

Register COM, COM+, and DCOM Files

COM servers (such as ActiveX, COM, and COM+ files) require special registration so that applications can access the files' interfaces. Traditionally, these EXEs, DLLs, and OCXs contained self-registration functions that could be invoked to register the files during installation. However, relying on self-registration can cause some problems: the end user cannot always be certain of what information is being registered or that the registry entries will be fully cleaned up when the file is uninstalled.

The solution provided by the Windows Installer service is to write the necessary registry entries during setup and then remove them when the COM component is uninstalled. This method helps to ensure that all the COM servers are registered in the appropriate way.

Register Product for Uninstallation

In order to uninstall a product, the operating system must know that the product is present. Therefore, a setup registers a product with the operating system so that it can be easily uninstalled. This registration is required for Windows logo compliance. Much of the information registered in this process is available to the end user through the Add/Remove Programs applet. For example, technical support contact information, product update information, product version, and product publisher information are all registered in this process.

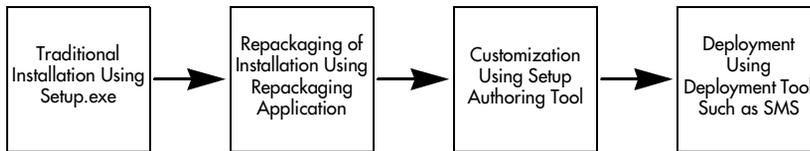


In addition to uninstalling, you can also repair or remove a product through Add/Remove Programs, assuming the setup author included the necessary functionality.

Installation Customization

Most companies which deploy software to large numbers of employees must perform customization of some degree to each program. This may be as simple as including a template with the company logo on it with a word processing application, or as complex as determining a specific feature set of a suite of applications. For traditional installations, this amounts to a four-step process:

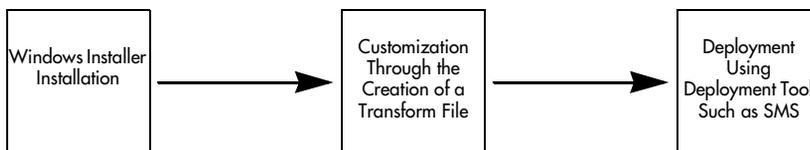
1. Acquire an off-the-shelf setup, created with InstallShield or other setup authoring tool.
2. Repackage the setup using a repackaging tool.
3. Customize the repackaged application, using a setup authoring tool.
4. Deploy the repackaged and customized setup using a deployment tool.



This process has to be performed in its entirety for each installation that needs customization, and is often a time consuming and laborious task.

For this, and other reasons, Microsoft has formulated a new installation model, using its new Windows Installer technology. Application data is stored in a database, along with installation instructions. The Windows Installer engine then reads the instructions and performs the installation. However, this poses a problem with the traditional customization path. First of all, Windows Installer-based installations cannot be repackaged. Therefore, Microsoft designed the concept of transforms, files that contain only customization information, leaving the original Windows Installer package untouched. The transform and the initial package together become the installation.

So the new model for installation customization is the following:



One of the main benefits is that the creation of these customizations is much simpler and less time consuming. You do not need a clean machine to capture all installation information for a repackager, nor do you have to repeat the same tasks over and over again for similar customizations. You merely create an initial transform



with the things common to each customization, and then use that as the basis for the things that are different from installation to installation. Best of all, since you never touch the initial installation package, all information stored within it is intact for customizations down the road. Tuner helps you create these transform (.MST) files.

Installation Package

An installation package contains all the information Windows Installer needs to run an installation or uninstallation routine for an application, including setup user interface information. Each package contains an .msi file containing an installation database, a summary information stream, and data streams for various installation parts. In addition, the .msi file can also contain transforms, internal source files, and external source files or cabinet files required by the installation.

MSI and MST Files

An MSI file is a Windows Installer setup package in its completed state. It includes setup resource files, and can have compressed within it all of the application's data files. The MSI file is the one that is distributed to the end users and will interact with the Windows Installer service in order to install the application on their machines.

The customization projects you create with Tuner that contain any changes to the base MSI file are called MST files or transforms.

Transform Files

When you need to customize a Windows Installer installation, you do so by creating a transform file. This file actually allows you to limit the features available during installation, as well as add or modify files, shortcuts, and registry entries. For example, you may need to customize an installation for different departments in your company. Typical business productivity suites come with a spreadsheet program, a word processor, and a presentation tool. Your accounting department may only need the spreadsheet and the presentation programs. On the other hand, the writing department may need only the word processor and the spreadsheet. A third department may need the entire suite of applications.

Instead of manually setting up every person in the company, you can take the original setup of the entire suite, and create a customization project in the form of a transform to meet the needs of each department. A transform would need to be created for every configuration that you plan to use.

Once you have created your transforms, you can apply them at run-time, depending on whose machine the application is being installed on. For example, you can check if the target machine is in the accounting department. If it is, the accounting transform will be applied to the original setup, and only the spreadsheet and presentation programs will be installed.

Transform files are designated by an .mst extension.



Validation

Validation is the process of comparing an MSI-based installation to a known set of guidelines (an evaluation file) to ensure it has been created to those guidelines. Tuner can perform two types of validations: prevalidation and postvalidation.

Prevalidation compares only the base MSI file to the evaluation file. This ensures that, when starting a customization project, the initial file was created using those guidelines. If it does not pass prevalidation, then the installation may work fine, but may not be able to use the full features of Windows Installer.

Postvalidation, on the other hand, compares the base MSI file and the changes made in a customization project against the evaluation file. In this case, the combination of the initial file and the subsequent modifications of the transform can produce different results than a comparison of just the base MSI file. If the initial file was valid on its own, then if postvalidation fails, the problems exist in the customization project. In some situations, advanced users may be able to use a transform file to make an initially invalid MSI file valid in conjunction with the transform.

Tuner provides two files that you can validate against: the Windows 2000 Logo Program Suite and the Full MSI Validation Suite. The Windows 2000 Logo Program Suite is a subset of the Full MSI Validation Suite, and is used to certify that the installation meets the Microsoft standards for the Windows logo. The Full MSI Validation Suite is used to ensure that the installation meets all MSI standards.

In practice, you should prevalidate the base MSI file to ensure compliance to MSI standards before you begin creating a customization project. After you have finished your project, postvalidate it to make sure it is still compliant. If the base installation was compliant and the postvalidation fails, go back to the changes you made to determine what caused the validation problems. For full details, consult the MSI help file, available from the Help menu.

Customization Steps

To assist you with your customization, Tuner provides you with a set of steps that cover all parts of the MSI file that can be customized in your transform project. You do not have to follow these steps in sequential order, or even complete all the steps. Below is a brief description of each step, and how they fit into the customization workflow.

1. **Prevalidation**—This step allows you to pre-check the base Windows Installer package to ensure it is valid according to MSI standards before you take the time to create the transform file. If it is invalid, there may be unexpected results with your installation. Tuner allows you to copy the results of the prevalidation to the clipboard, where you can paste them into a message to the application's vendor. This is one of the most important steps in the Tuner workflow. Without it, all of your work might be wasted.
2. **Setup Organization**—This allows you to specify both the default destination that the installation will suggest, as well as the default company name it will suggest. You also can modify some properties of individual features in the product.



3. **Target System Configuration**—This is where Tuner shows you its true customization power. You can add files and registry entries to the installation, allowing you to add company-specific files, templates, etc. to the installation. For example, if you have a set of word processing templates your company uses, you can include them in the installation, and even send them to the correct folder during installation. Further, you can modify or remove existing shortcuts, or add your own as necessary.
4. **Additional Servers**—If you intend to provide a network-based installation of the application, and want to specify the location of the base installation files for advertised features or repairing damaged features, you can specify additional server locations in this step.
5. **Application Configuration**—This step allows you to change functionality for the Add/Remove applet in the Control Panel for Windows 2000. You can also see the full spectrum of MSI properties as they exist in the base Windows Installer package, and add, modify, or remove them as necessary.
6. **Postvalidation**—This should be the last step you perform, which allows you to verify that your project, consisting of the original Windows Installer package and the transform file you have created, meets MSI-validation standards.

Conflict Identification and Resolution

Administrators may have to maintain hundreds or thousands of applications in their day-to-day activities. These may result from an equal number of installations. Windows Installer functionality provides much assistance to beleaguered administrators. However, even if an installation works correctly independently, there is no certainty that it will work well with other applications. And this may ultimately reduce the effectiveness of the additional Windows Installer functionality.

ConflictSolver provides administrators a means of locating and reporting different types of conflicts that may exist within Windows Installer packages, as well as between known packages. Using this information, administrators can make the appropriate modifications (using both built-in solutions in ConflictSolver as well as manual solutions using Tuner or Author) to resolve the issues.

ConflictSolver identifies and resolves problems that can arise between different Windows Installer-based installations. This includes ensuring the correct product is uninstalled when necessary, or verifying that duplicate components and files are not installed multiple times.

There are two main functions of InstallShield ConflictSolver:

- **Validation**—allows administrators to verify Windows Installer packages are built to a specific criteria (usually Microsoft guidelines). If a package is valid, conflict errors can be isolated to those between packages, rather than conflicts resulting from incorrectly built packages. If packages are invalid, you can usually fix these errors using InstallShield Tuner.
- **Conflict Identification and Resolution**—reports conflicts between a given Windows Installer package and those already imported into the Conflict Database. InstallShield ConflictSolver uses Application Conflict Evaluators (ACEs)



as rules to determine if there are conflicts between different packages, and reports them as they are discovered. In many cases, InstallShield Conflict-Solver can fix these issues automatically; in some cases, issues may require user intervention.

Application Conflict Evaluators (ACEs)

Application Conflict Evaluators (ACEs) are the rules that allow the ConflictSolver application to identify conflicts between Windows Installer-based setups. The ConflictSolver application can automatically solve some of the conflicts spelled out in these rules. When an automatic remedy is available, the associated rule used to fix the conflict is spelled out in a Conflict Application Resolution Definition (CARD).

ACE Types

ConflictSolver uses ACEs to identify the following conflict types:

- Component (ACE02, ACE04, ACE05, ACE06, ACE09)
- Files (ACE03, ACE07, ACE08, ACE12)
- Registry (ACE10)
- Shortcuts (ACE13)
- INI File (ACE14)
- ODBC Resources (ACE15)
- NT Services (ACE16)
- File Extensions (ACE17)
- Product Codes (ACE19, ACE20)

ACE Descriptions

The following table provides information about the various ACEs.

ACE	Conflict Type	Description
ACE01	None	Validates that the ACE mechanism is working. This ACE uses the Time property to get the time and returns either the system time or None. ACE01 should never fail.
ACE02	Component	Checks to see if identical Component GUIDs have identical destination paths.
ACE03	File	If component GUIDs are identical, the same files must be contained in the GUID.
ACE04	File	Ensures each component contains a key file.
ACE05	Component	Checks for the existence of more than one executable (EXE, DLL, OCX, CHM, or HLP) per component in a Windows Installer package.



ACE	Conflict Type	Description
ACE06	Component	Checks whether or not the EXE or a DLL file within a component is the key file.
ACE07	Component	Checks components to ensure that those that contain identical files (name, version, and destination) have identical GUIDs.
ACE08	File	Checks identical component GUIDs to ensure the versions of files match.
ACE09	File	Checks to ensure that Merge Modules with identical GUIDs have identical versions.
ACE10	Registry	Checks for the existence of the same Root/Key/Value registry combination within different component GUIDs.
ACE12	File	Checks for components containing files that already exist in Merge Modules.
ACE13	Shortcut	Checks for the existence of the same shortcut within different component GUIDs.
ACE14	INI File	Checks for the existence of the same INI file within different component GUIDs.
ACE15	ODBC Resources	Checks for the existence of identical ODBC entries within different component GUIDs.
ACE16	NT Services	Checks for the existence of identical NT Services within different component GUIDs.
ACE17	File Extension	Checks for identical file extensions within different component GUIDs
ACE19	Product GUID	Checks the Product Code to ensure it is unique.
ACE20	Product GUID	Checks the Upgrade Code to ensure it is unique. If it is not, checks other product information.



InstallShield AdminStudio Tutorial

Table of Contents

This tutorial is designed to take you through a typical pre-deployment process, starting with a traditional setup. Included with the AdminStudio installation is the InstallShield PackageForTheWeb setup. The goal is to turn this traditional setup into a Windows Installer package, add some customization to it, and check it against a database of known Windows Installer packages for conflicts. The sections in this chapter walk you through each of the phases of this process.

Preparation	3-3
Launching Repackager Remotely	3-3
Repackaging a Traditional Installation	3-4
Modifying and Building	
a Windows Installer Installation	3-11
Opening the .inc File	3-11
Modifying the Project	3-15
Building the Release	3-16
Customizing an Installation Using Tuner	3-18
Opening the Package and Creating	
a Transform	3-18
Prevalidation	3-19
Customization	3-20
Specifying Default Destination	
and Organization Information	3-20
Configuring Features	3-21
Adding Files to the Installation	3-22
Creating Shortcuts	3-23
Configuring Windows 2000	
Add/Remove Programs Settings	3-24
Postvalidation	3-25
Packaging	3-26
Further Customizations	3-29
Identifying and Resolving Conflicts	3-30
Opening ConflictSolver and	
Accessing the Database	3-30
Configuring ConflictSolver Options	3-31
Validating the Package and Transform	3-32
Identifying Conflicts	3-34



Resolving Conflicts	3-37
Checking for Conflicts a Second Time . . .	3-40
Importing the File	3-42
Conclusion	3-45



Preparation

Prior to running this tutorial, you need to install AdminStudio on a development machine. You also need to prepare a clean machine on which you will repackage the setup used in this tutorial.

The tutorial is broken up into four steps: repackaging, converting to a Windows Installer package, tuning, and identifying and resolving conflicts. During each step, you save your output to existing folders included with the AdminStudio installation.

For purposes of this tutorial, <AdminStudio Directory> represents the directory in which AdminStudio is installed.



Launching Repackager Remotely

The first step in this tutorial is to launch the Repackager remotely from a clean system. A clean system is essential so that all changes are captured correctly and ultimately converted into a Windows Installer package.

To access the Repackager remotely:

1. From your clean machine, connect to the machine on which AdminStudio is installed.
2. Navigate to the <AdminStudio Directory>\Repackager folder and launch isrepackager.exe.

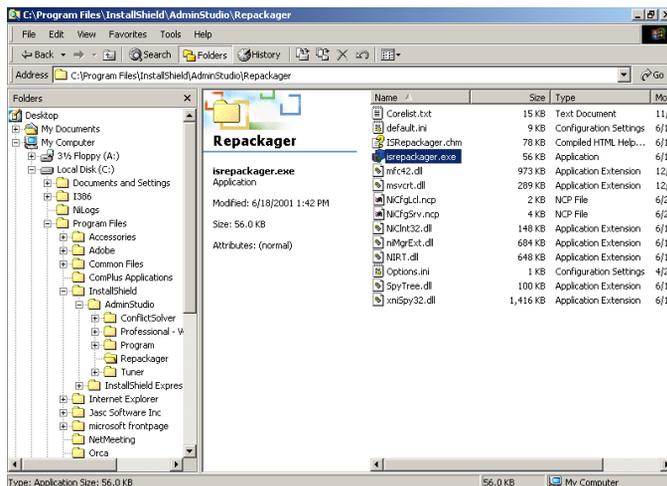


Figure 3-1.
Selecting the isrepackager.exe from the AdminStudio\Repackager directory.



Repackaging a Traditional Installation

The goal of this step is to repackage the setup into a Repackager script file, which Author can then import.

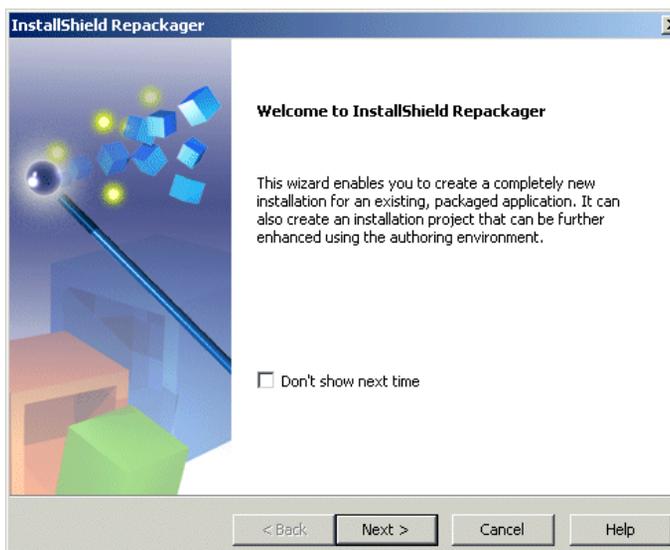
With the Repackager launched as described in the section above, do the following:

1. From the Welcome panel, click Next.



Figure 3-2.

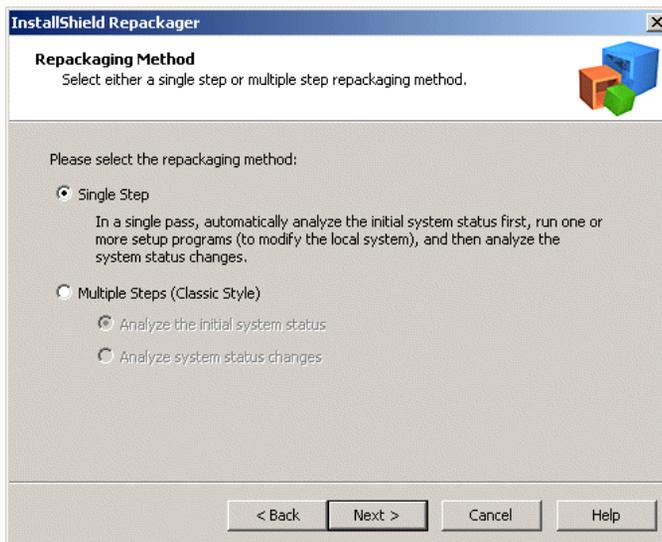
Welcome to InstallShield Repackager panel.



2. In the Repackaging Method panel, select Single Step. Click Next.

Figure 3-3.

Repackaging Method panel.



3. In the Collect Product Information panel, provide product information as follows:



- a. Click the Browse button (...) to the right of the Program File field and browse over the network to <AdminStudio Directory>\Tutorial\Repackager\PackageForTheWeb 3 directory and select PackageForTheWeb3.exe. Click Open
- b. Ensure the Product Name field reads **PackageForTheWeb 3** and the Version No. field is **3.00**.
- c. Enter **http://www.installshield.com** as the value for both the Product URL and the Support URL.
- d. Change the Company Name to **InstallShield Software Corporation**.
- e. Change the Install Directory to **[ProgramFiles-Folder] \ InstallShield \ PackageForTheWeb 3**.
- f. Click Next.

InstallShield Repackager

Collect Product Information
Provide product information for repackaging. Items with asterisks (*) are required.

Setup Programs

*Program File: C:\Program Files\InstallShield\AdminStudio\Tutorial\Repackager\...
Command-line Argument(s):
Additional Setup Programs

Product Information

*Product Name: PackageForTheWeb 3 Version No.: 3.00
Product URL: http://www.installshield. Support URL: http://www.installshield.
Company Name: InstallShield Software Corporation
Install Directory: ProgramFilesFolder)\InstallShield\PackageForTheWeb 3

< Back Next > Cancel Help

Figure 3-4.
Collect Product Information panel.



Figure 3-5.

Set Target Project Information panel.

4. In the Set Target Project Information panel, click the Browse button (...) and navigate to <AdminStudio Directory>\Tutorial\Repackager\Output. Click Next.

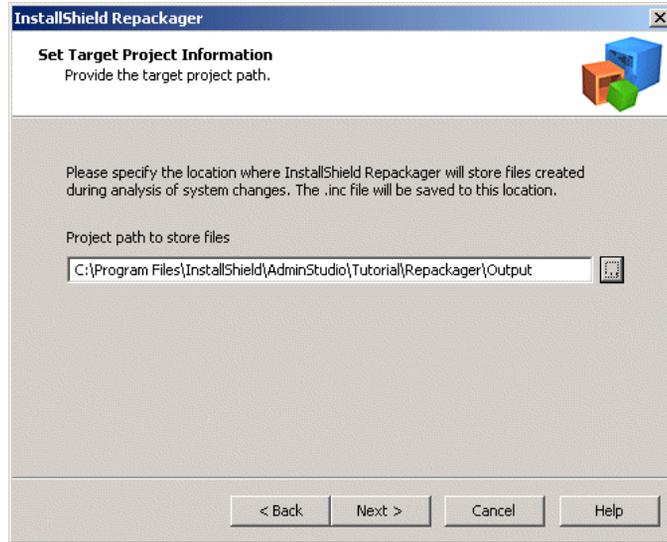
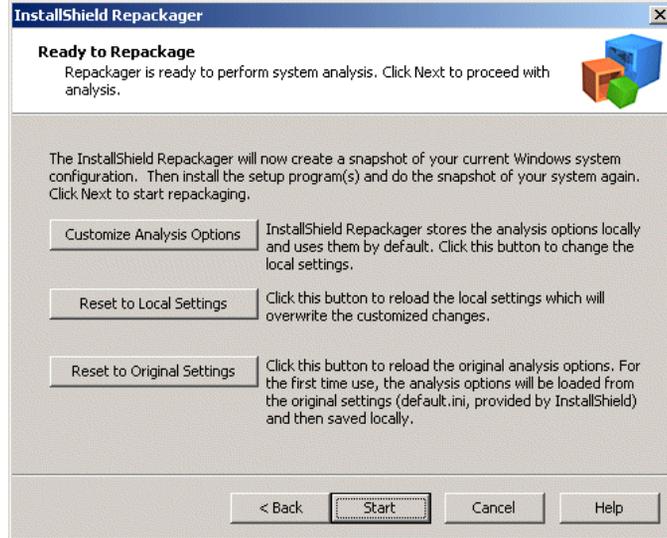


Figure 3-6.

Ready to Repackage panel.

5. In the Ready to Repackage panel, click Start. You could also select to customize analysis options, but the default options are sufficient for this tutorial.





- When the initial system status analysis completes, you are prompted to run the setup. Click OK. The InstallShield Wizard launches to extract PackageForTheWeb 3 for installation.



Figure 3-7.
The Initial Scan Complete message box.

- When the PackageForTheWeb3 extraction Welcome panel appears, click Next to extract the program.

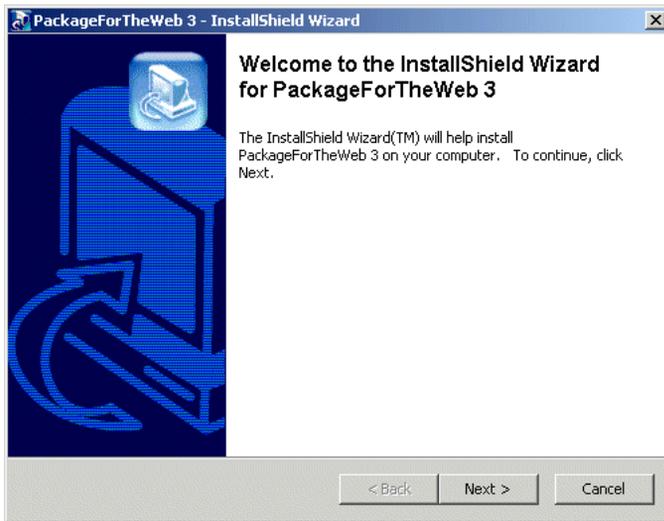


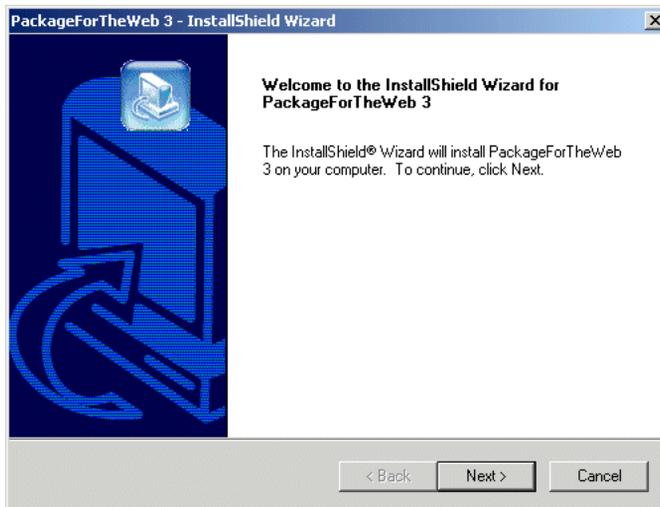
Figure 3-8.
PackageForTheWeb 3 extraction Welcome panel.



8. After extracting PackageForTheWeb, the PackageForTheWeb 3 Installation Welcome panel appears. Click Next.

Figure 3-9.

PackageForTheWeb 3 installation Welcome panel.



9. Read the license agreement. Accept the terms and click Next.

Figure 3-10.

PackageForTheWeb License Agreement panel.





10. Enter your name as the User Name and your organization as the Company Name. Click Next.



Figure 3-11.
PackageForTheWeb Customer Information panel.

11. Click Next on the Destination Folder panel. For this installation, allow PackageForTheWeb to install in its default location.

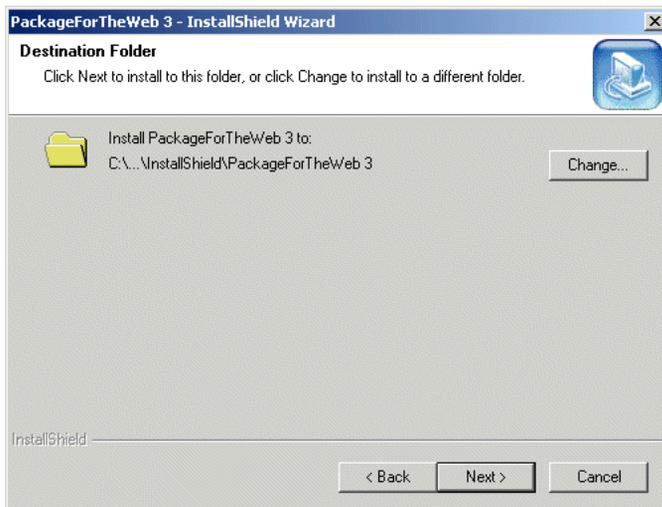


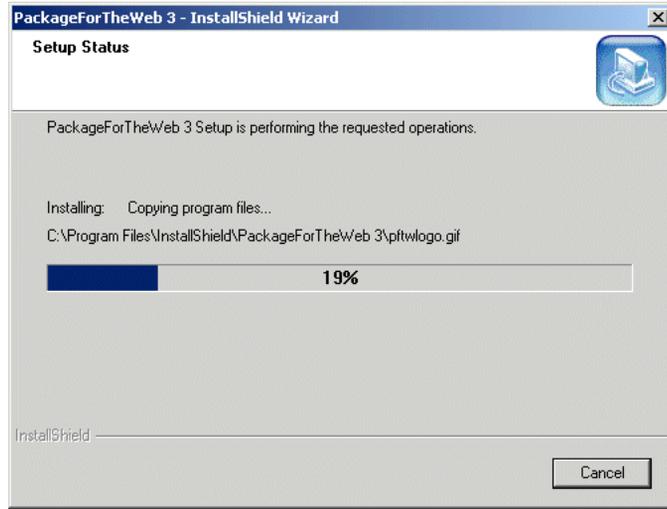
Figure 3-12.
PackageForTheWeb Destination Folder panel.



Figure 3-13.

PackageForTheWeb Setup Status panel.

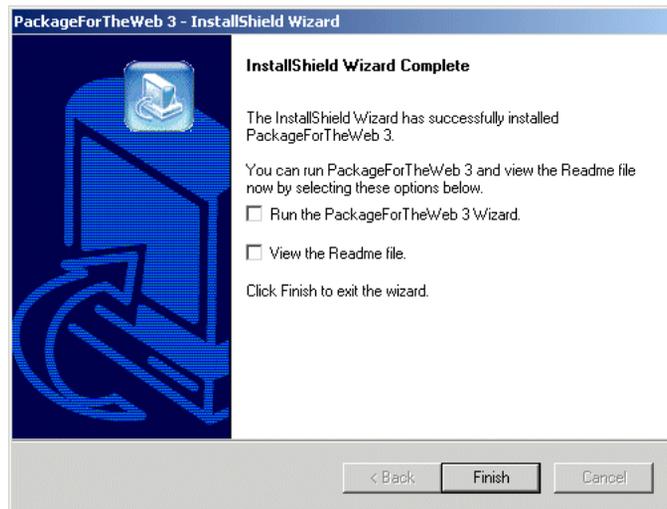
12. From the Ready to Install the Program panel, click Install.



13. From the InstallShield Wizard Complete panel, click Finish.

Figure 3-14.

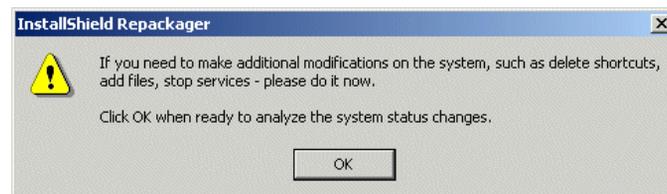
PackageForTheWeb 3 Installation Completed panel.



14. When prompted to analyze system status changes, click OK. Analysis of system status changes begins.

Figure 3-15.

The Ready to Analyze System Status Changes message box.





15. When the Repackager finishes, click OK.

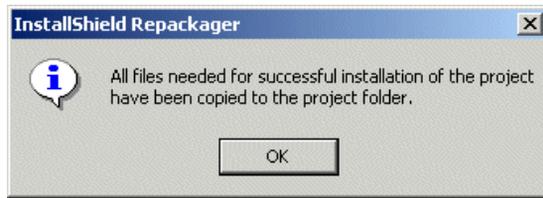


Figure 3-16.
The Repackaging Completed message box.

16. Read the summary and click Finish. The Repackager exits.

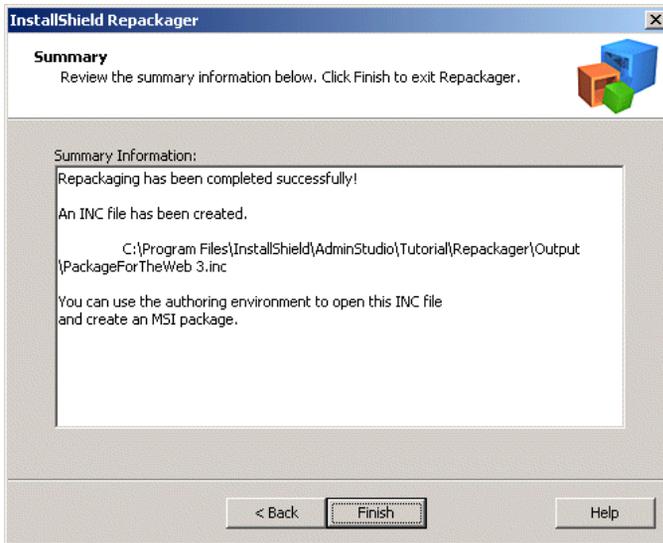


Figure 3-17.
Repackager Summary panel.

After repackaging, all files necessary for conversion to a Windows Installer package are located in the folder specified in the Set Target Project Information panel (in <AdminStudio Directory>\Tutorial\Repackager\Output).

Modifying and Building a Windows Installer Installation

Author, although primarily used in AdminStudio for converting Repackager output into Windows Installer packages, is a full-featured authoring tool. Using it, you can create new Windows Installer packages, or open existing Windows Installer packages and modify them, in addition to the workflow described in this tutorial. Consult Author's online help for comprehensive authoring information.

Opening the .inc File

Repackager created an .inc file during repackaging, which Author reads and converts to an Author project (.ism). Once the repackaged data is stored within an .ism, you



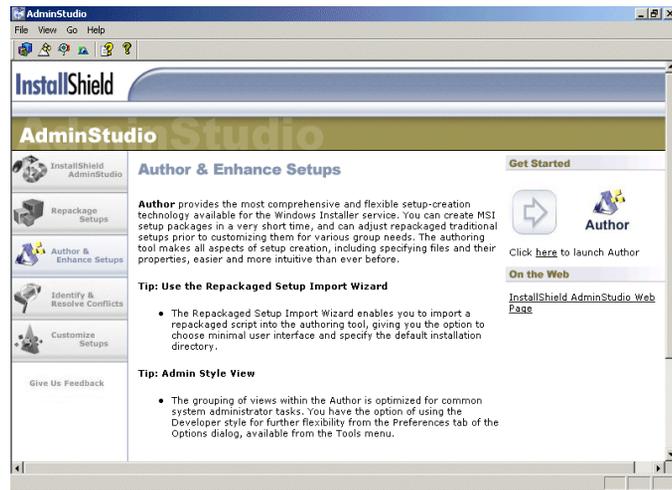
can make modifications as necessary, and finally build the .ism into a Windows Installer package (.msi).

To open and convert the .inc file, do the following:

1. Launch AdminStudio and then launch Author.

Figure 3-18.

The Author & Enhance Setups page in AdminStudio.



2. Select Open from the File menu.
3. Browse to the PackageForTheWeb 3.inc file created by InstallShield Repackager. This is at the location specified during the Repackager portion of the tutorial (<AdminStudio Directory>\Tutorial\Repackager\Output). The Import Repackaged Setup wizard launches.
4. On the Welcome panel, click Next.

Figure 3-19.

Import Repackaged Setup wizard Welcome panel.





- From the Project Information panel click the Browse button (...) to the right of the Project Name field. From the resulting dialog box, browse to <AdminStudio Directory>\Tutorial\Author\Output and name the file PackageForTheWeb 3.ism. Click Save. When the dialog is dismissed, click Next on the Project Information panel.

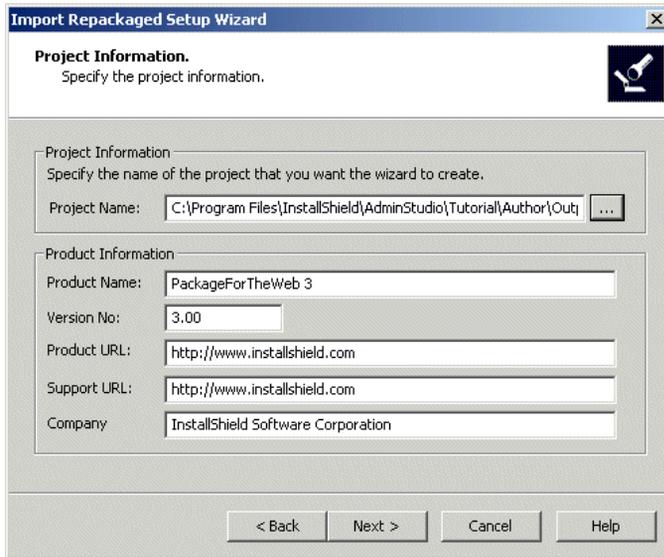


Figure 3-20.

Import Repackaged Setup wizard Project Information panel.

- Provide your name as the author and PackageForTheWeb 3 as the subject in the Summary Stream dialog. Click Next.

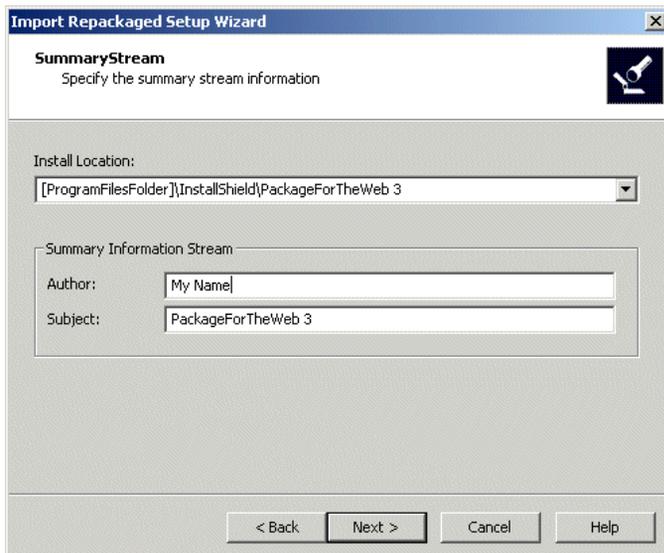


Figure 3-21.

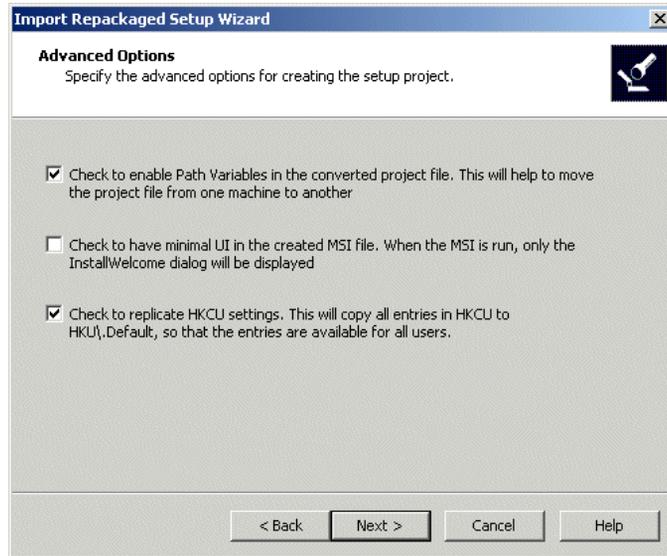
Import Repackaged Setup wizard Summary Stream panel.



Figure 3-22.

*Import Repackaged Setup wizard
Advanced Options panel.*

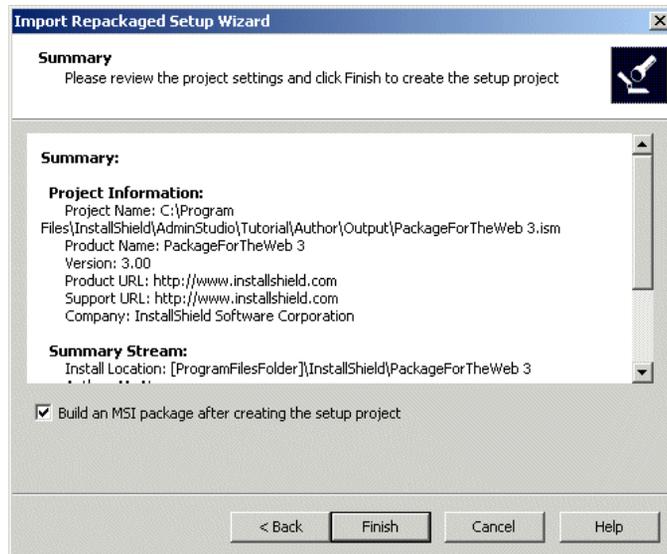
7. In the Advanced Options panel, ensure all boxes are checked except for the middle box (minimal user interface). Click Next.



8. Review the summary. Check the Build an MSI package after creating the setup project box.

Figure 3-23.

*Import Repackaged Setup wizard Sum-
mary panel.*





9. Click Finish to convert the script file to an Author project and MSI.

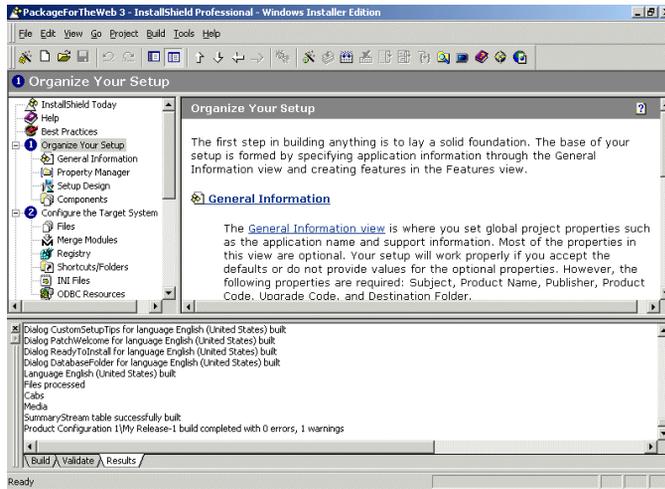


Figure 3-24.
The Results tab of the Output window

Modifying the Project

Once converted to an .ism project, you can make customizations as necessary. For this tutorial, you are going to add your name as the setup author, and add some authoring comments. You are also going to add a file to a component and make it the key file.

Setting Project Properties

To modify the project, do the following:

1. Select the General Information view from the checklist.
2. Select Project Properties from the General Information tree.
3. Click in the Setup Author Name field and enter your name.
4. Click in the Authoring Comments field and enter AdminStudio Tutorial.

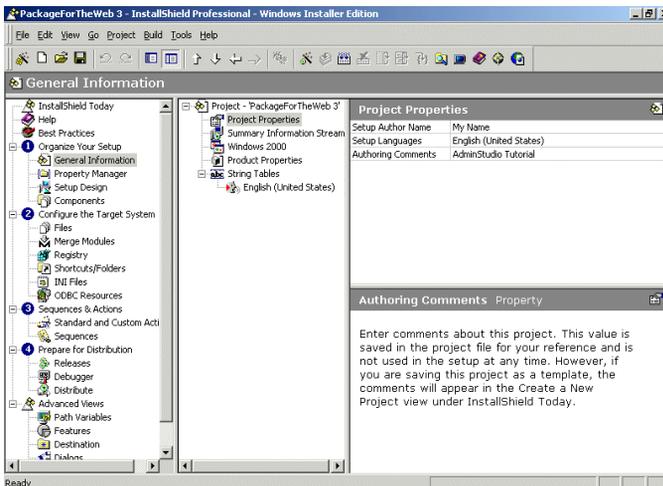


Figure 3-25.
Project Properties view.

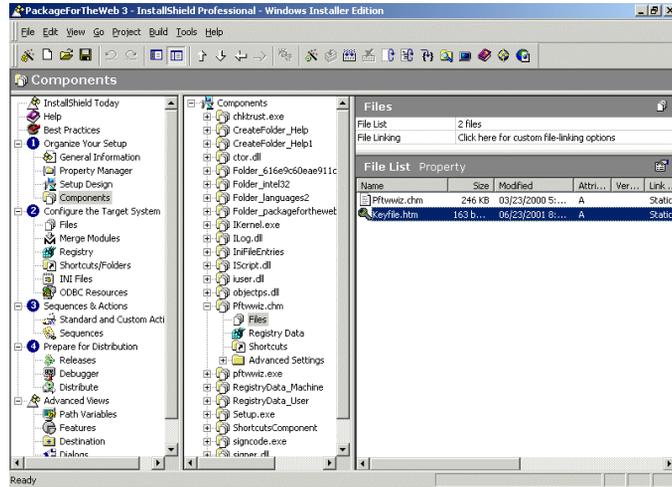


Adding a File and Setting the Key File

To add a file to an existing component and make it the key file, do the following:

1. Select Components from the Checklist.
2. From the Components tree, click and expand the Pftwwiz.chm component.
3. Click on the Files node below Pftwwiz.chm.
4. In the File list, right-click and select Add.
5. Navigate to the <AdminStudio Directory>\Tutorial\Author\Files and select Keyfile.htm. Click Open. Keyfile.htm appears in the File list.
6. Right-click Keyfile.htm and select Set Key File.
7. When prompted for confirmation, click Yes.

Figure 3-26.
Components view.



Building the Release

Once you have made modifications to the Author project, you need to build it into a Windows Installer package again to reflect those changes.

To build the release, do the following:





1. From the Build menu, select Build.

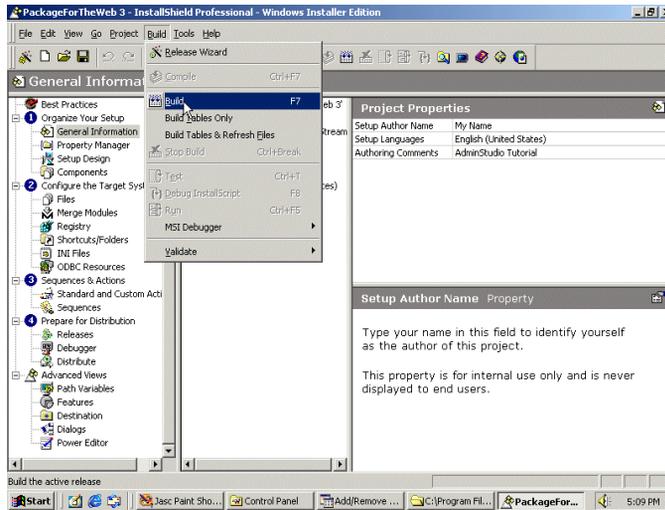


Figure 3-27.
Selecting Build from the Build menu.

2. Verify the build completes successfully by monitoring the information in the Build tab of the Output window.

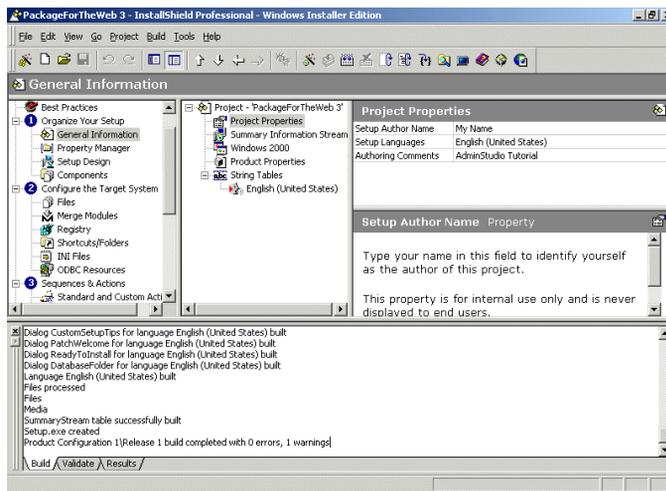


Figure 3-28.
The Build tab of the Output window.

When completed, the PackageForTheWeb 3.msi is located at <AdminStudio Directory>\Tutorial\Author\Output\PackageForTheWeb 3\Product Configuration 1\Release1\DiskImages\DISK1.



Customizing an Installation Using Tuner



Although this tutorial is designed as a complete predeployment process, if you have not completed the other phases of the process (Repackaging and Modifying/Building), the output to begin this section can be found at <AdminStudio Directory>\Tutorial\Tuner\Source.

The Microsoft-recommended method of customizing Windows Installer-based installations is via a transform file. This ensures that modifications are made outside of the base package. If you intend to deploy the same installation to different sets of users with modifications of the installation, it is much easier to make the changes necessary in transform files, rather than in the original Windows Installer package. This also protects the software vendor's original installation from being modified incorrectly.

Opening the Package and Creating a Transform

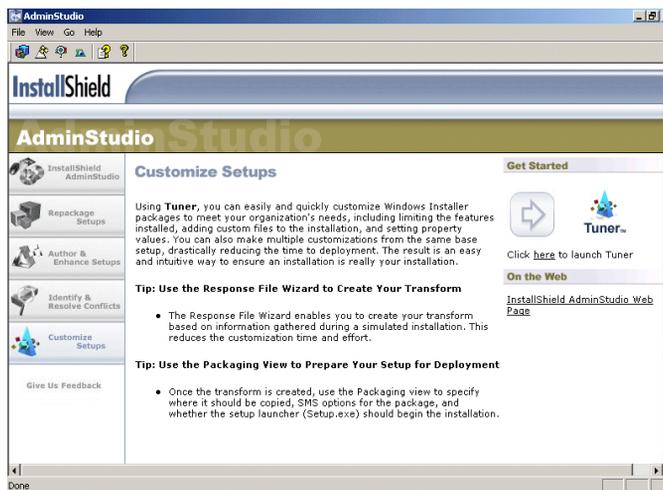
To open the base Windows Installer package and create a new transform, do the following:

1. From AdminStudio, launch Tuner.



Figure 3-29.

The Customize Setups page in AdminStudio.



2. Select Create a New Transform File.
3. In the Base Windows Installer Package area, click Browse and navigate to the <AdminStudio Directory>\Tutorial\Author\Output directory. Select the PackageForTheWeb 3.msi file and click Open.
4. In the Windows Installer Transforms area, click Browse and navigate to the <AdminStudio Directory>\Tutorial\Tuner\Output directory and create a file called PFTW.mst.



5. Click Create Transform File.

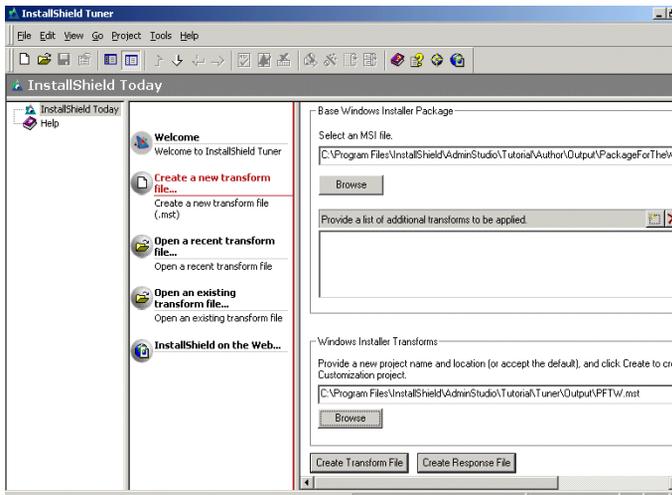


Figure 3-30.
Creating a new transform in Tuner.

6. Click the Save button.

Prevalidation

The first step after creating the transform file is to run prevalidation on the original Windows Installer package. InstallShield Tuner's prevalidation process ensures the Windows Installer package you are customizing is valid according to a predetermined set of guidelines. In most cases you will want to run prevalidation using Microsoft's Full MSI Validation Suite, a set of over 70 guidelines that ensure the package was created to Microsoft standards.

To begin prevalidation on the base Windows Installer package, do the following:

1. Click Prevalidation from the Checklist.

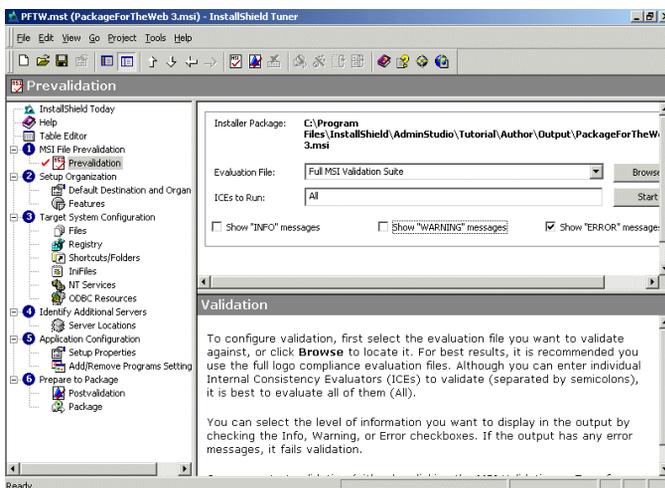


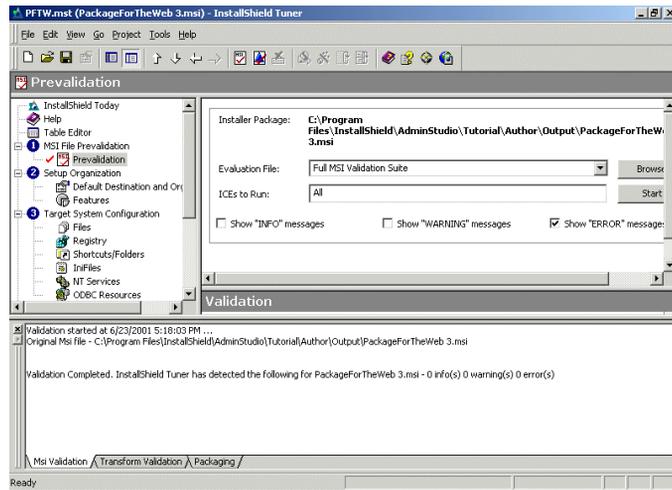
Figure 3-31.
Prevalidation view.



Figure 3-32.

MSI Validation tab of the Output window.

2. From the top-right pane, select Full Validation Suite as the evaluation file, and ensure the ICEs to Run field is set to All.
3. Check the Show “ERROR” messages box, and uncheck both Show “INFO” messages and Show “Warning” messages. If any errors occur during prevalidation, then the package fails the validation process.
4. Click Start. InstallShield Tuner begins the validation process, displaying its progress in the MSI Validation tab of the Output window.



When completed, a message appears indicating that the PackageForTheWeb 3.msi file has no validation errors.



In the event that a Windows Installer package did have validation errors, it means that the package was not created to Microsoft's guidelines. Consult the online help for information about handling invalid packages.

Customization

Tuner can be used for a wide variety of customizations, including changing default destination and organization information, feature display and installation options, addition, modifications, and removal of files, INI files, registry information, and shortcuts, and configuration of ODBC resources and NT services. Some of these modifications are more typical than others, so this tutorial concentrates on those modifications.

Specifying Default Destination and Organization Information

To specify default destination and organization information, do the following:

1. From the Checklist, click Default Destination and Organization.
2. Click on the value field for the Default Destination Path. Change the value to ProgramFilesFolder\InstallShield\AdminStudio\PFTW. This becomes the default destination directory when you run the setup.





3. In the Company Name value field, provide the name of your organization (up to 30 characters). This is the default suggested organization when you run the setup.

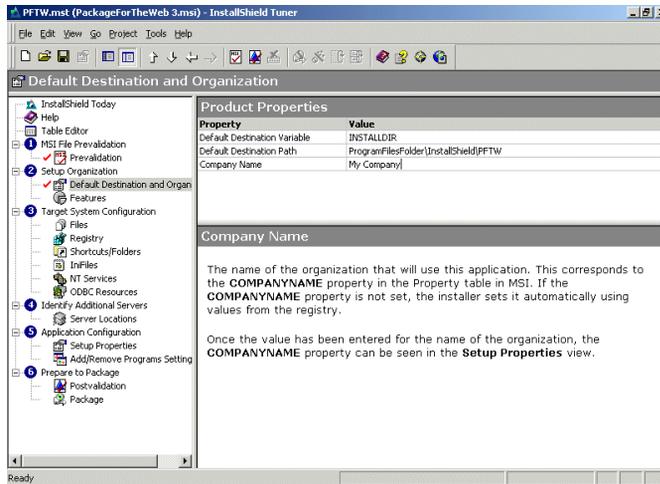


Figure 3-33.
Default Destination and Organization view.

The Default Destination Variable has been selected by the application vendor (or the InstallShield Repackager) for specific reasons. Unless you have a substantial need to change it, you should avoid reassigning this value. Any changes have the potential to seriously hamper the installation. In the event you do inadvertently change this value, select it from the grid and click Reset to revert to the original value from the base Windows Installer package.



Configuring Features

During customization, you can also change how features are presented during setup. For example, you may want a specific feature installed by default, without giving a user the option to not install it from the Custom Setup panel during installation.

To configure this feature to work as described above, do the following.

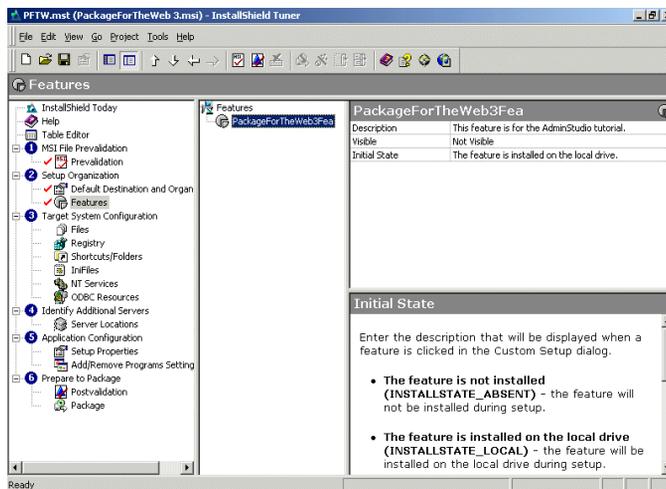
1. Select Features from the checklist.
2. Select PackageForTheWebFeature from the Features tree.
3. Enter “This feature is for the AdminStudio tutorial.” in the Description field.
4. Click on the Visible property’s value cell and change it to Not Visible.





Figure 3-34.
Features view.

5. Click on the Initial State property's value and change it to The feature is installed on the local drive.



With these configurations, this feature and all its associated files will always be installed on the target computer. The end-user will not have the opportunity to prevent the feature from installing because it will not be displayed in the Custom Setup panel during installation.

Adding Files to the Installation

One of the more common customizations to a setup is adding files. An example of when you might want to do this is if you have a series of custom templates to include with the setup. You can configure these templates to be installed to the correct location when the installation is performed.



To add files to the installation, do the following:

1. Select Files from the checklist.
2. In the Source computer's directory tree pane, navigate to <AdminStudio Directory>\Tutorial\Tuner\Files.
3. In the Destination computer's folders pane, right click on Destination Computer and select ProgramFilesFolder from the Show Folder submenu. Expand this folder until you see the PackageForTheWeb 3 folder. Select this folder.



4. Select the AdminStudioTutorial.htm and AdminStudioTutorial.txt files from the Source computer's files pane and drag them to the Destination computer's files pane.

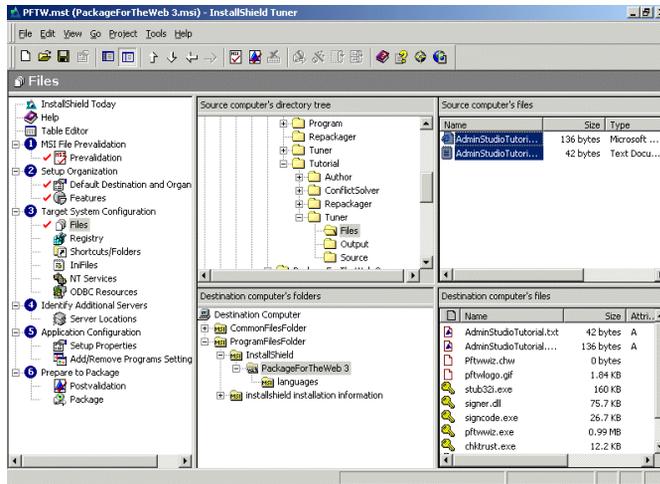


Figure 3-35.
Files view.

These files, which are not part of the original PackageForTheWeb installation, will be installed when this transform is used with the base Windows Installer package.

Creating Shortcuts

Another common customization to make with InstallShield Tuner is to create additional shortcuts. These can be to files in the original Windows Installer package, to files added via your customization, or to files already on the target system. In this example, you are going to create a shortcut to PackageForTheWeb on the desktop.

To create the shortcut, do the following:

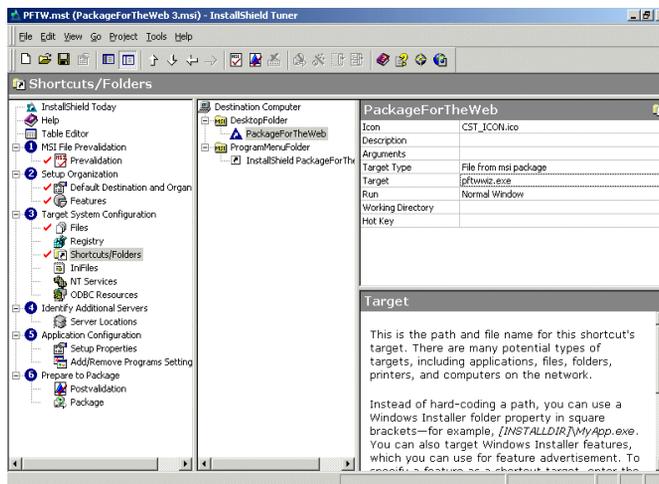
1. Click the Shortcuts/Folders view from the checklist.
2. Click on the folder where you want to add the shortcut. In this case, you are going to place the shortcut on the Desktop. However, since the DesktopFolder is not displayed in the Shortcut/Folder view, right-click on the top-level Shortcuts folder and select DesktopFolder from the Show Folder submenu.
3. Right-click the DesktopFolder and select New Shortcut. Name this shortcut PackageForTheWeb.
4. When InstallShield Tuner displays the properties for the new shortcut, perform the following configurations:
 - a. Select the Icon property and click the Change Icon button in the pane below the properties grid.
 - b. Navigate to <AdminStudio Directory>\Tutorial\Repackages\PackageForTheWeb 3 and select PackageForTheWeb.exe. Click Open.
 - c. Select the icon in this file and click OK.
 - d. Change the Target Type to File from MSI Package. Click in this view again and, when the Select File dialog appears, select pftwiz.exe and click OK.





Figure 3-36.
Shortcuts/Folders view.

e. Double-click the Run value and select Normal window from the pull-down menu.



When the installation is installed, this shortcut will be placed on the desktop of the target system.

Configuring Windows 2000 Add/Remove Programs Settings

Windows 2000 provides additional functionality to the Add/Remove Programs control panel. InstallShield Tuner enables you to modify settings pertaining to this in your customization. In this example, you are going to disable the control panel setting that allows users to Modify a setup after it has been installed.



To set the Modify property in the Add/Remove Programs control panel, do the following:

1. Click the Add/Remove Programs Setting view from the checklist.
2. Double-click the value field to the right of the Disable Modify Button property.



3. From the pull-down menu, select Yes.

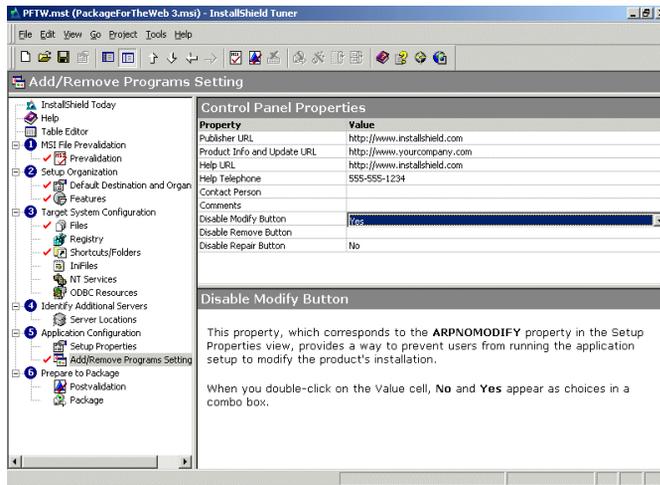


Figure 3-37.
Add/Remove Programs Setting view.

When this installation and transform are run on a Windows 2000 operating system, users will not be permitted to use the Modify function for this setup in the Add/Remove Programs dialog.

Postvalidation

After you have performed customizations, and prior to packaging the base Windows Installer package and transform, you need to postvalidate it. This ensures that your transform did not introduce validation issues that were not present in the base Windows Installer package. In some circumstances, you can use a transform to correct validation errors identified during prevalidation. However, in this example, you are only concerned that you did not introduce errors via the transform.

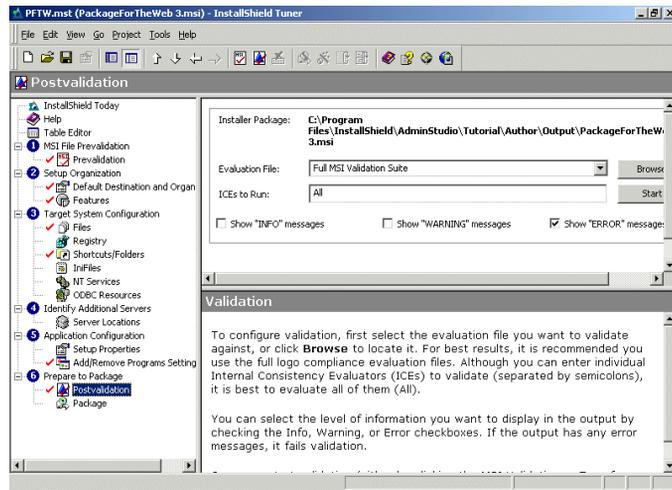
To postvalidate the combination of the base Windows Installer package and transform file, do the following:





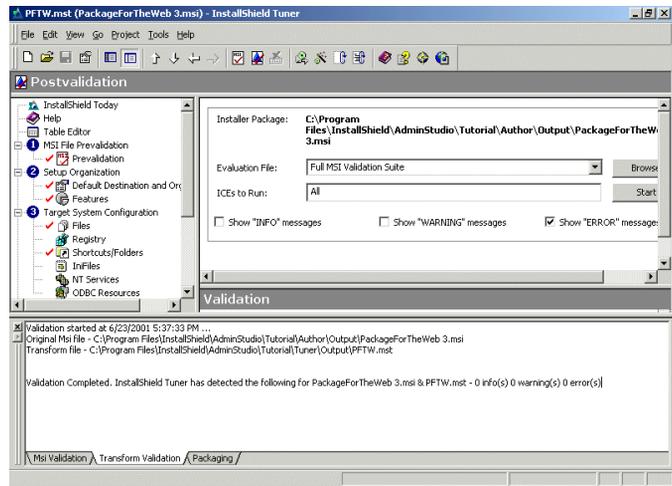
Figure 3-38.
Postvalidation view.

1. Select Postvalidation from the checklist.



2. Ensure that the Error box is checked, but the Warning and Info boxes are unchecked.
3. Select the Full MSI Validation Suite as the evaluation file.
4. Click Start. Messages will appear in the Transform Validation pane of the Output window.

Figure 3-39.
Transform Validation tab of the Output window.



When postvalidation completes, you are informed that the combined Windows Installer package and transform are valid according to Microsoft guidelines.

Packaging

In the final part of this tutorial, you will package the original Windows Installer package and transform into a single executable file (Setup.exe). This file will begin the installation, including all modifications you made.



To package the Windows Installer installation and transform, do the following:

1. Select the Packaging Wizard from the Project menu.

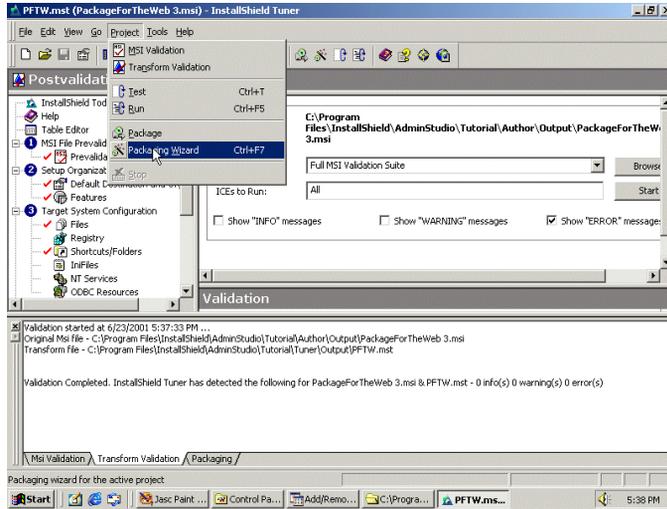


Figure 3-40.
Selecting the Packaging wizard.

2. Click the Network Location option and set the location to <AdminStudio Directory>\Tutorial\Tuner\Output\Packaged.Click OK and then click Next.

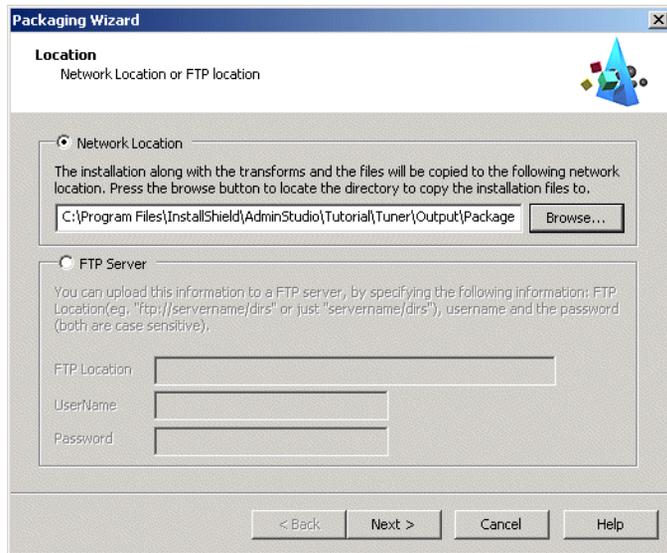


Figure 3-41.
The Packaging wizard Location panel.

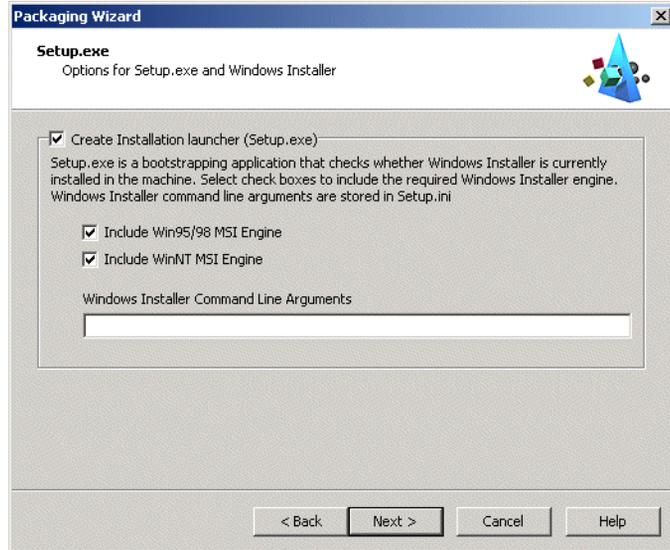
3. Click the Create Installation Launcher box, and check both the include engine boxes. This ensures that, should the Windows Installer engine not be installed



Figure 3-42.

The Packaging wizard Setup.exe panel.

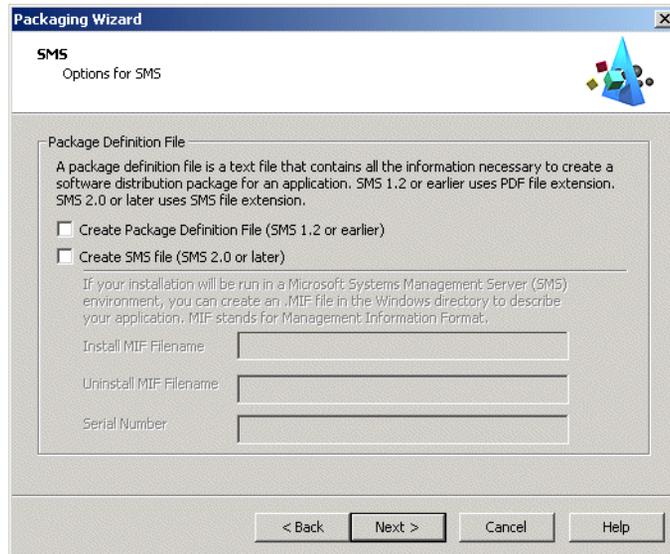
on the target system, that it will be installed so that installation of your package can occur. Click Next.



4. In the SMS panel, ensure all boxes are unchecked and click Next.

Figure 3-43.

The Packaging wizard SMS panel.





5. Review the selected options and click Finish.

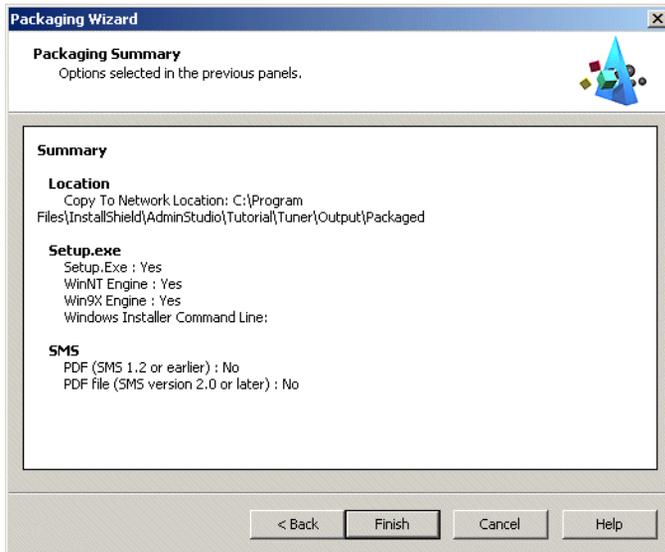


Figure 3-44.
The Packaging wizard Packaging Summary panel.

The Packaging pane in the Output window displays the packaging progress.

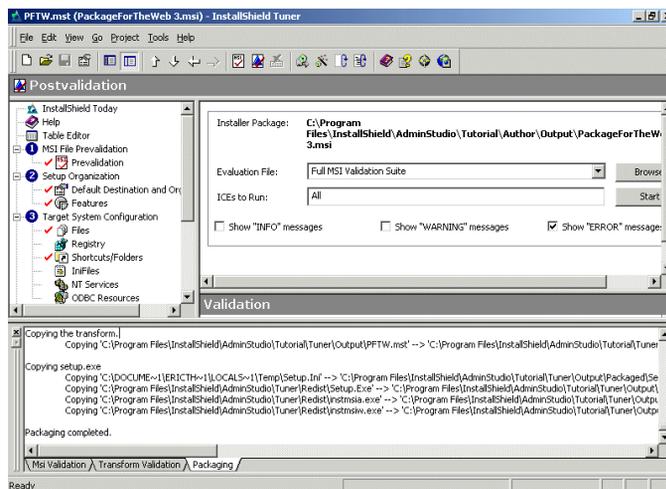


Figure 3-45.
The Packaging tab of the Output window.

The executable, and all necessary files, are now stored at <AdminStudio Directory>\Tutorial\Tuner\Output\Packaged.

Further Customizations

InstallShield Tuner is capable of many customizations not included in this tutorial. For example, your package may contain ODBC resources, NT Services, or INI files that need configuration. Additionally, you may want to modify the files installed, or even change registry settings. InstallShield Tuner provides flexibility to create these



types of customizations depending on your individual needs. Please review the InstallShield Tuner online help system for further details.

Identifying and Resolving Conflicts



Although this tutorial is designed as a complete predeployment process, if you have not completed the other phases of the process (Repackaging, Modifying/Building, and Customizing), the output to begin this section can be found in the <AdminStudio Directory>\Tutorial\ConflictSolver\Source directory.

ConflictSolver can identify and resolve conflicts that exist between multiple Windows Installer packages, ensuring proper functioning of installations.

Opening ConflictSolver and Accessing the Database

The first step in using ConflictSolver is to launch it and connect to a Conflict Database. For purposes of this tutorial, use the sample database included with ConflictSolver. This database contains information about common Windows Installer packages.

To open InstallShield ConflictSolver and access the proper database, do the following:

1. From AdminStudio, launch ConflictSolver.



Figure 3-46.

The Identify & Resolve Conflicts page in AdminStudio.





2. Connect to the Sample Database. This Conflict Database is populated with information about several common Windows Installer packages.

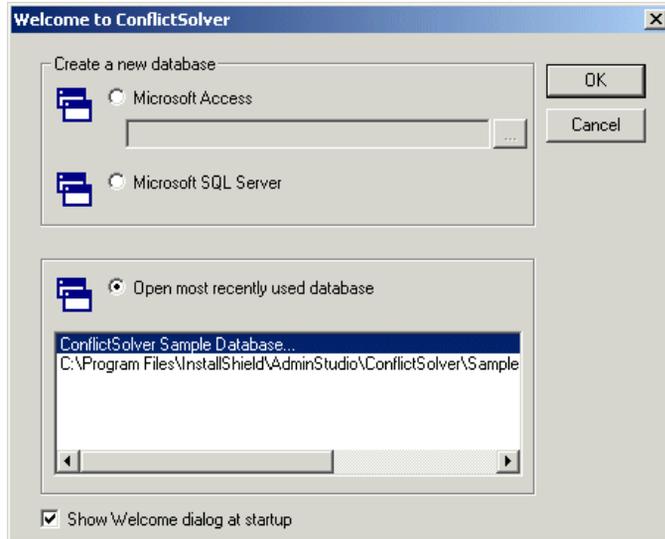


Figure 3-47.
Welcome to ConflictSolver dialog box.

Configuring ConflictSolver Options

Before importing the package into the database and identifying conflicts, you need to configure the ConflictSolver options pertaining to import, validation, and conflict identification and resolution.

To configure these options, do the following:

1. Select Options from the Tools menu.
2. Click General to display the General tab.

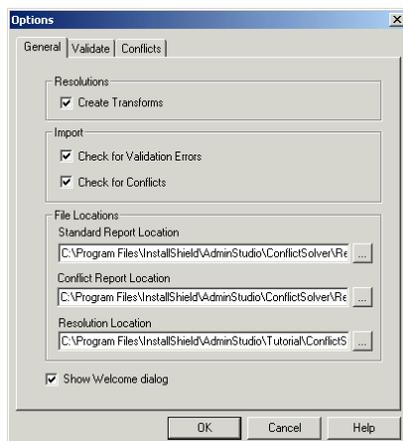


Figure 3-48.
The General tab of the ConflictSolver Options dialog.

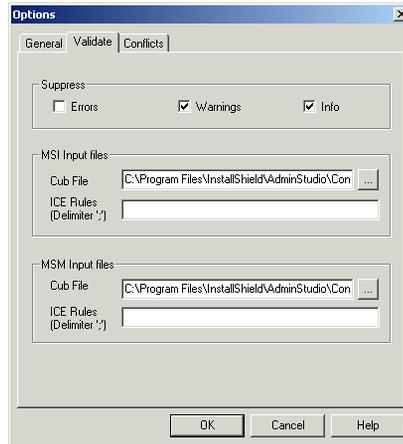
3. Ensure the Create Transforms, Check for Validation Errors, and Check for Conflicts options are checked.



- Click the Browse button (...) to the right of the Resolution Location field and navigate to <AdminStudio Directory>\Tutorial\ConflictSolver\Output. Click OK.
- Click the Validate tab. Ensure in the Suppress area that Warnings and Info are checked, and Errors is unchecked. As with validation in Tuner, if any errors occur during validation, it fails validation.

Figure 3-49.

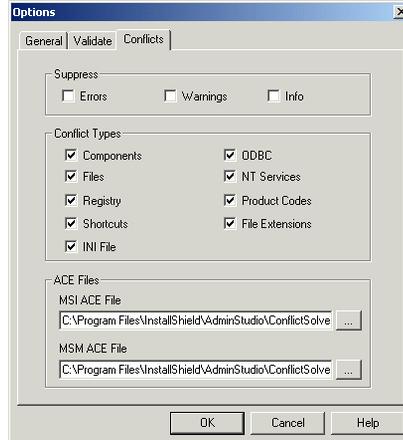
The Validate tab of the ConflictSolver Options dialog.



- Click the Conflicts tab. Ensure that all three Suppress options are unchecked. This enables you to see all messages about conflicts.

Figure 3-50.

The Conflicts tab of the ConflictSolver Options dialog.



- Under Conflict Types, check all of the options.
- Click OK to close the Options dialog.

Validating the Package and Transform



To validate the package and transform, do the following:

- Select Validate from the Conflicts menu. The Validation wizard launches.
- On the MSI Source Information panel, click Browse and navigate to the <AdminStudio Directory>\Tutorial\Author\Output\PackageForTheWeb



3\Product Configuration 1\Release 1\DiskImages\DISK1 directory and select PackageForTheWeb 3.msi as the Windows Installer package. Click Open.

3. Click the New button above the Transforms area.
4. Click in the Transforms area.
5. Click the Browse button (...) to the right of the transform, and browse to the <AdminStudio Directory>\Tutorial\Tuner\Output directory and select PFTW.mst. Click Open.

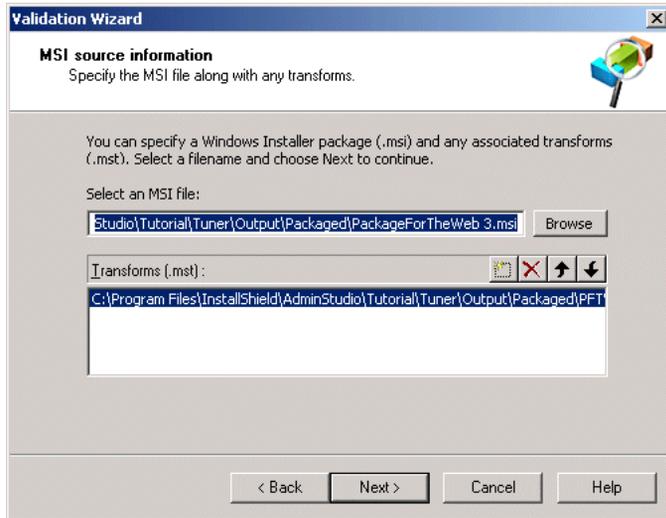


Figure 3-51.
Validation wizard MSI Source Information panel.

6. Click Next.
7. Review the information presented in the Summary panel. If these are correct, click Finish to begin validation. Progress appears in the Validate tab of the Output window.

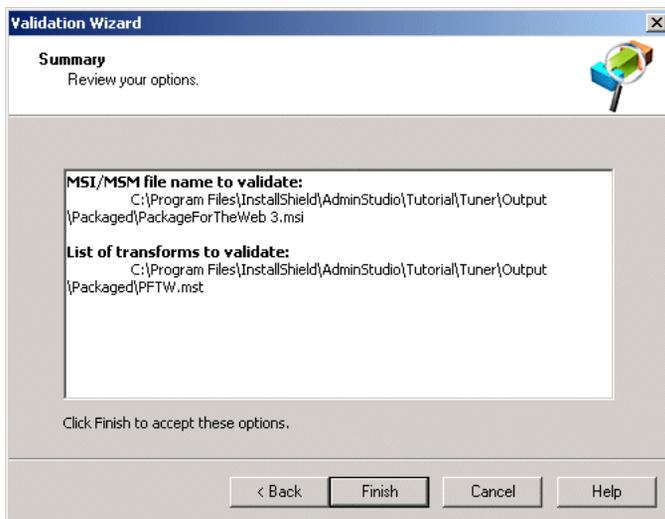


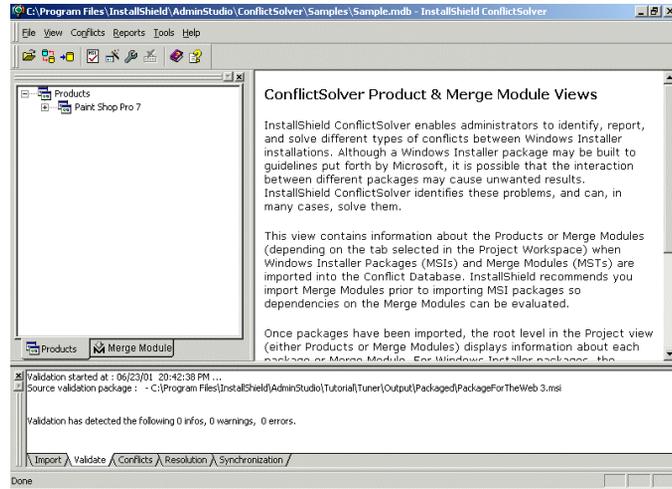
Figure 3-52.
Validation wizard Summary panel.



When validation completes, you are informed that the package and transform are valid.

Figure 3-53.

The Validate tab of the Output window.



Identifying Conflicts

Once validation is complete, you should identify conflicts that may exist between the package you intend to import and packages already in the Conflict Database. The conflict types checked by default is determined by the options set in when configuring ConflictSolver options, but these can be changed within the Conflict wizard.



To identify conflicts, do the following:

1. Select Conflict Wizard from the Conflicts menu.
2. On the Welcome panel, click Next.

Figure 3-54.

Conflict wizard Welcome panel.





3. On the Source Type panel, select Windows Installer Package and click Next.

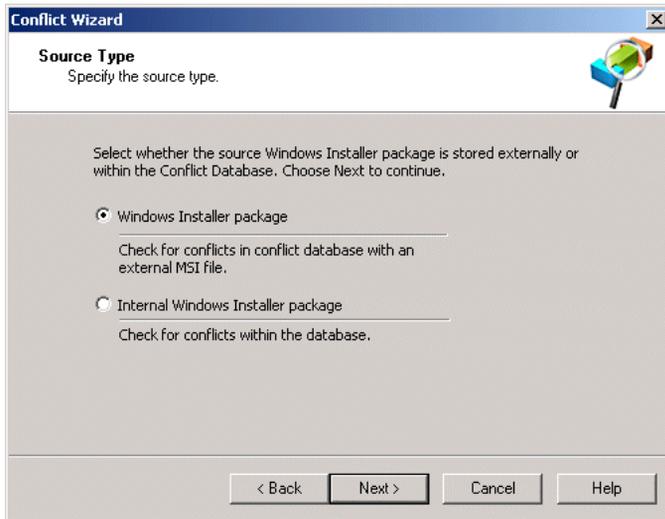


Figure 3-55.
Conflict wizard Source Type panel.

4. When the MSI Source Information panel appears, click Browse.
- From the Open dialog, navigate to <AdminStudio Directory>\Tutorial\Author\Output\PackageForTheWeb 3\Product Configuration 1\Release 1\DiskImages\DISK1 and select PackageForTheWeb 3.msi.
 - Click the New button in the Transforms area.
 - In the Transforms area, click Browse (...) and navigate to <AdminStudio Directory>\Tutorial\Tuner\Output and select PFTW.mst.

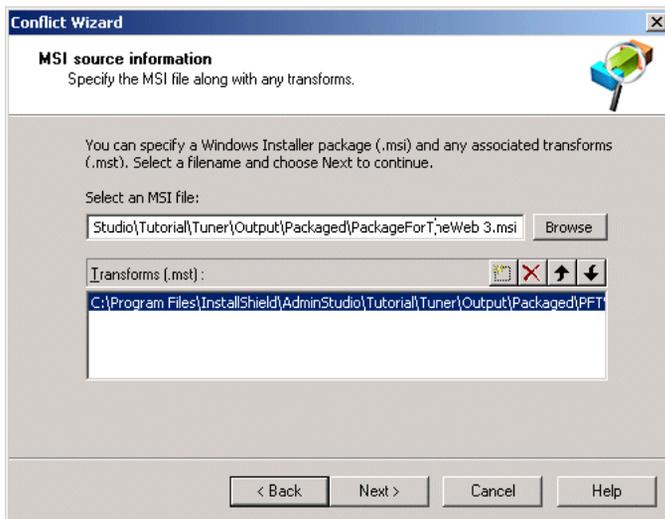


Figure 3-56.
Conflict wizard MSI Source Information panel.

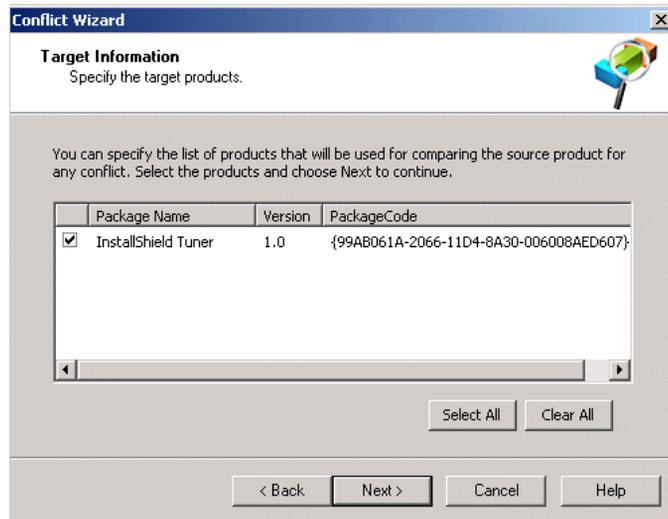
5. Click Next.



Figure 3-57.

Conflict wizard Target Information panel.

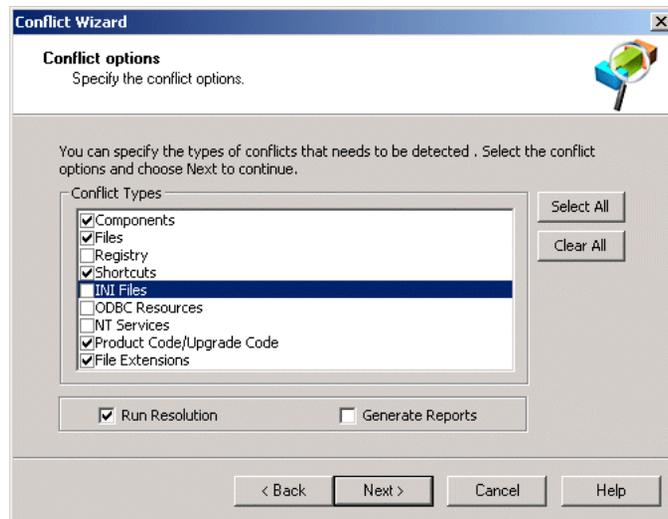
6. On the Target Information panel, click Select InstallShield Tuner to perform conflict comparison against InstallShield Tuner in the Conflict Database. Click Next.



7. On the Conflict Options panel, all conflict types should already be checked. Deselect Registry, INI Files, ODBC Resources, and NT Services.

Figure 3-58.

Conflict wizard Conflict Options panel.



8. Ensure Run Resolution is checked and Generate Reports is unchecked before clicking Next.



- In the Summary panel, review the options selected in the previous panels. Click Finish to begin conflict identification.

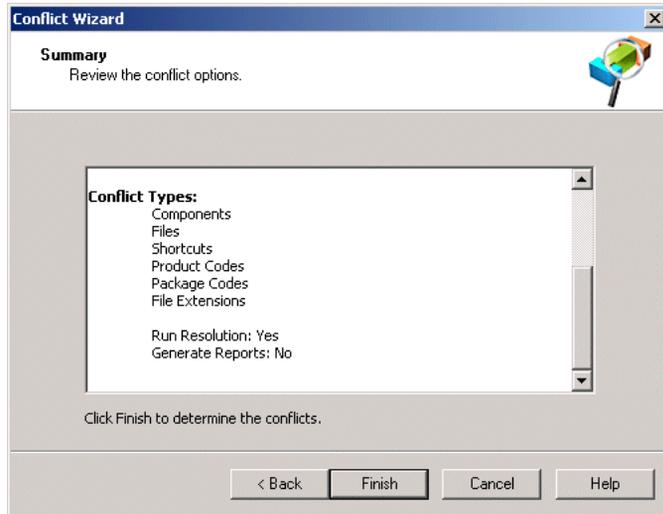


Figure 3-59.
Conflict wizard Summary panel.

The messages about the conflict identification process are displayed in the Conflicts tab of the Output window.

When finished, you find that there is a conflict that exists between the PackageForTheWeb 3.msi/PFTW.mst package and the InstallShield Tuner package already in the Conflict Database. Because you checked the Run Resolution option from the Conflict Options panel of the Conflict wizard, the Resolution wizard automatically launches following the end of the Conflict wizard.

Resolving Conflicts

Once conflicts have been identified, you can use the Resolution wizard to resolve some conflicts. Others may need manual resolution using Author or Tuner. For this tutorial, all conflicts can be resolved manually.

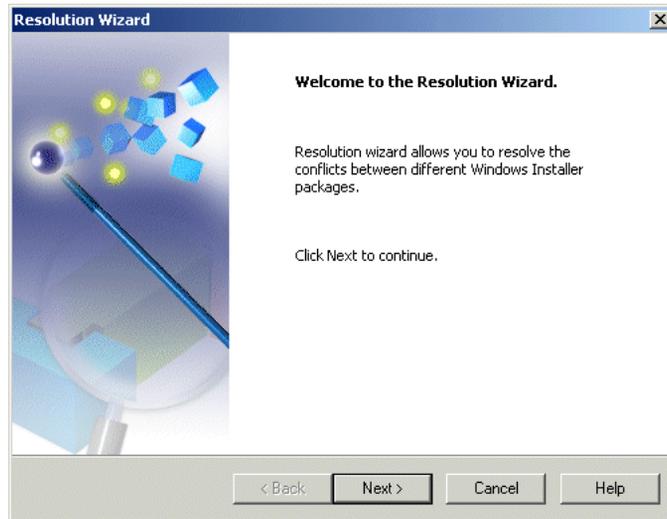
To use the Resolution wizard to correct resolvable conflicts, do the following:





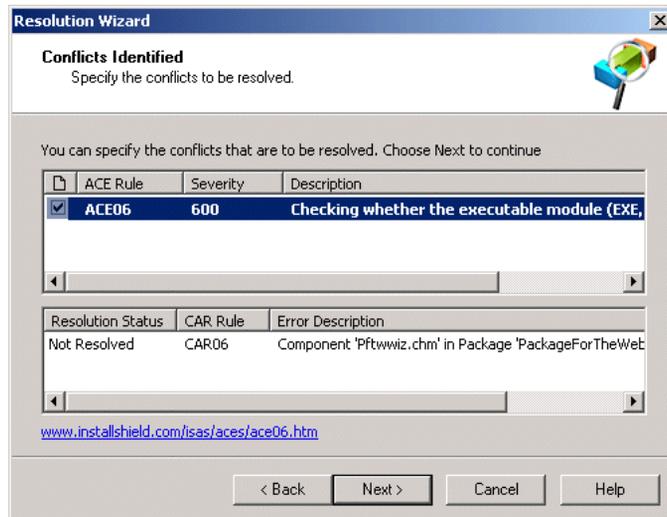
Figure 3-60.
Resolution wizard Welcome panel.

1. On the Welcome panel of the Resolution wizard, click Next.



2. From the Conflicts Identified panel, click the check box to the left of each conflict displayed in bold type. Conflicts listed in bold type can be automatically resolved by ConflictSolver.

Figure 3-61.
Resolution wizard Conflicts Identified panel.





3. Click Next. The Resolution Options panel appears. Select to save changes in a transform, and ensure the directory is set to <AdminStudio Directory>\Tutorial\ConflictSolver\PackageForTheWeb 3.mst.

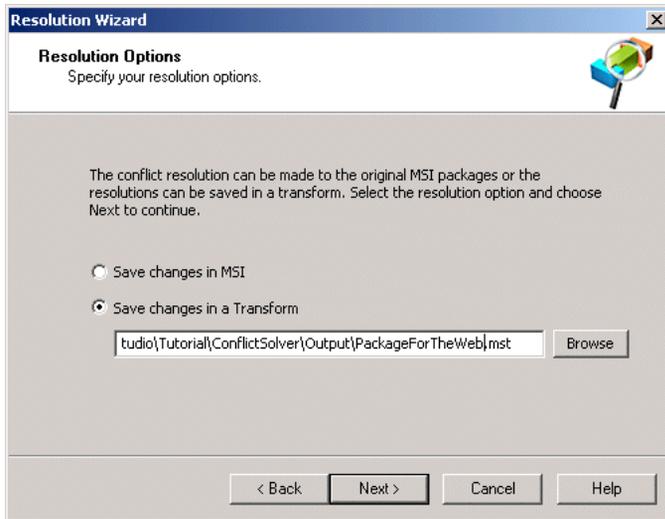


Figure 3-62.
Resolution wizard Resolution Options panel.

4. Click Next. The resolution progress appears in the Resolution tab of the Output window.

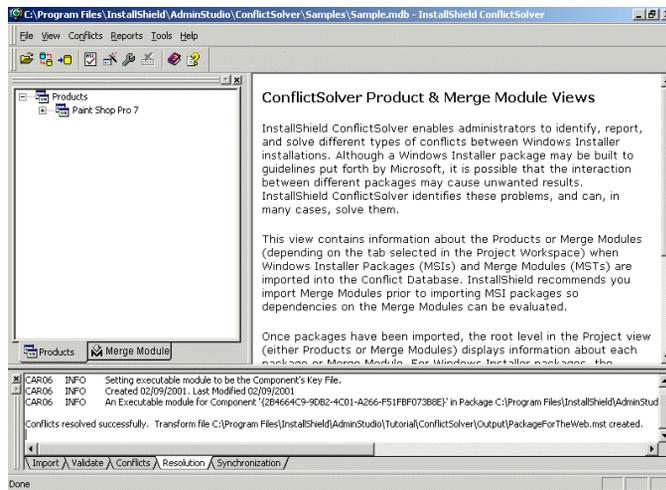


Figure 3-63.
Resolution tab of the Output window.

ConflictSolver automatically resolves the conflicts between the PackageForTheWeb 3.msi package and the packages in the database. The resolutions are encapsulated in a transform file called PackageForTheWeb 3.mst, stored in the <AdminStudio Directory>\Tutorial\ConflictSolver\Output directory (which you specified when configuring ConflictSolver options).



Checking for Conflicts a Second Time

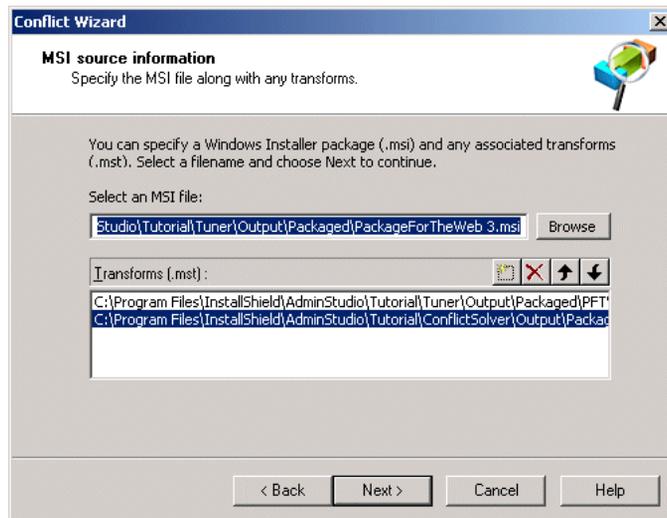
Although optional, it is generally a good practice to check that the conflicts you automatically resolved were resolved correctly before importing the package into the Conflict Database. Doing so involves running the Conflict Wizard a second time and checking the package and resolution transform for conflicts.

To verify conflicts have been resolved, do the following:

1. Select Conflict Wizard from the Conflicts menu.
2. On the Welcome panel, click Next.
3. On the Source Types panel, select Windows Installer Package and click Next.
4. When the MSI Source Information panel appears, click Browse.
 - a. From the Open dialog, navigate to <AdminStudio Directory>\Tutorial\Author\Output\PackageForTheWeb 3\Product Configuration 1\Release 1\DiskImages\DISK1 and select PackageForTheWeb 3.msi.
 - b. Click the New button in the Transforms area.
 - c. In the Transforms area, click Browse (...) and navigate to <AdminStudio Directory>\Tutorial\Tuner\Output and select PFTW.mst.
 - d. Click the New button in the Transforms area again.
 - e. In the new entry in the Transforms area, click Browse (...) and navigate to <AdminStudio Directory>\Tutorial\ConflictSolver\Output and select PackageForTheWeb 3.mst.

Figure 3-64.

Conflict wizard MSI Source Information panel.



5. Click Next.



- On the Target Information panel, click Select All to perform conflict comparison against all packages in the Conflict Database. Click Next.

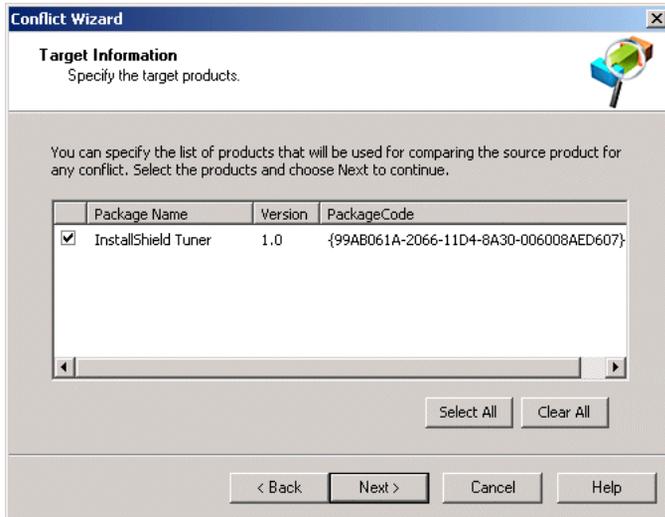


Figure 3-65.
Conflict wizard Target Information panel.

- On the Conflict Options panel, select InstallShield Tuner as the package against which you want to check your package and transforms.
- Ensure Run Resolution is checked and Generate Reports is unchecked before clicking Next.

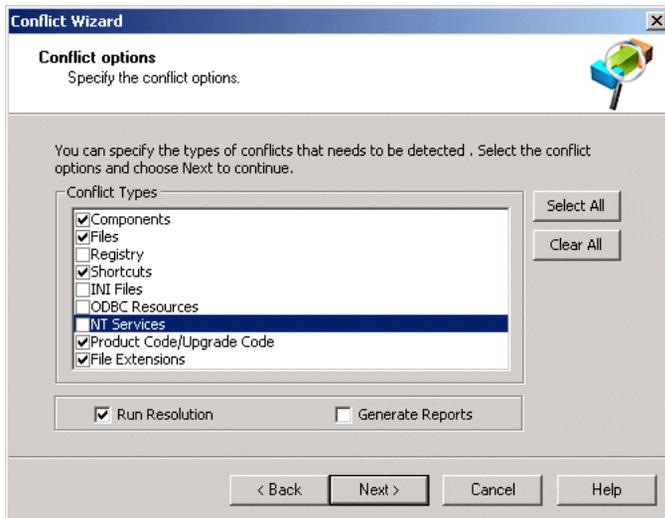


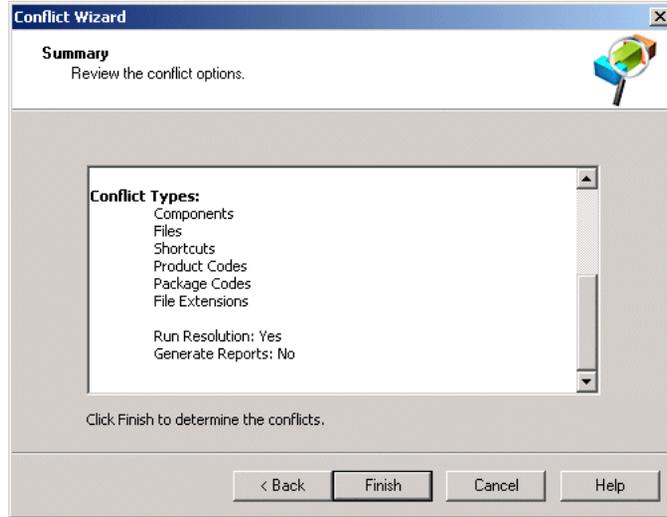
Figure 3-66.
Conflict wizard Conflict Options panel.



Figure 3-67.

Conflict wizard Summary panel.

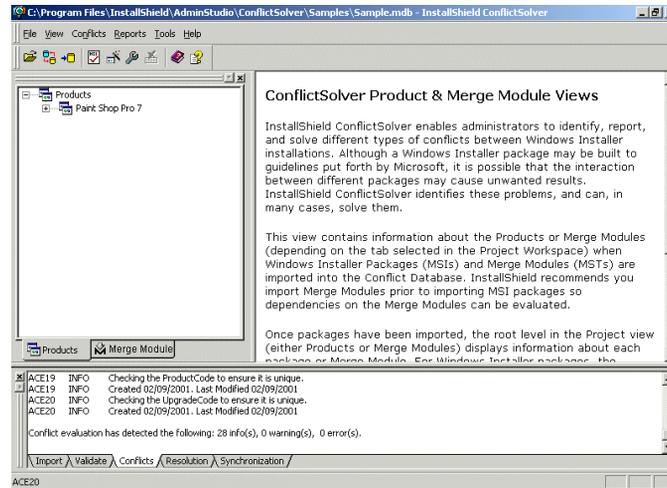
9. In the Summary panel, review the options selected in the previous panels. Click Finish to begin conflict identification.



The messages about the conflict identification process are displayed in the Conflicts tab of the Output window. When finished, you find that no conflicts remain after correcting them using the Resolution wizard.

Figure 3-68.

The Conflicts tab of the Output window.



Importing the File

Following validation, conflict identification, and conflict resolution, you can import the Windows Installer package, PFTW transform, and resolution transform into the Conflict Database so packages in the future can be compared against it.

To import the Windows Installer package and associated transforms, do the following:





1. On the Import wizard Welcome panel, click Next.

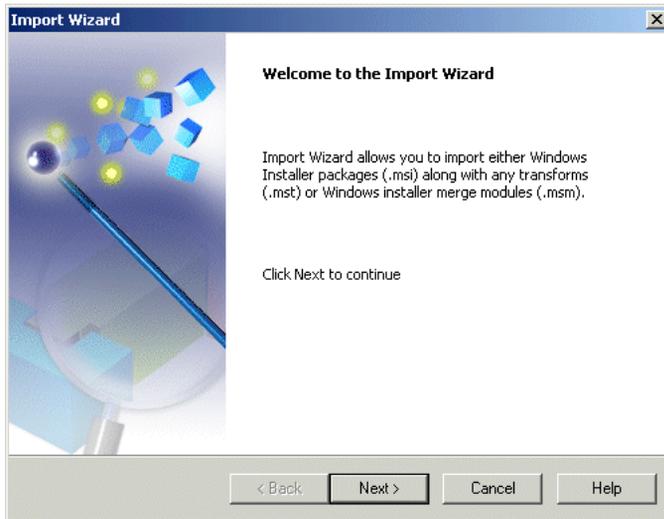


Figure 3-69.
Import wizard Welcome panel.

2. From the Import Options panel, select Import MSI/MST. Click Next.

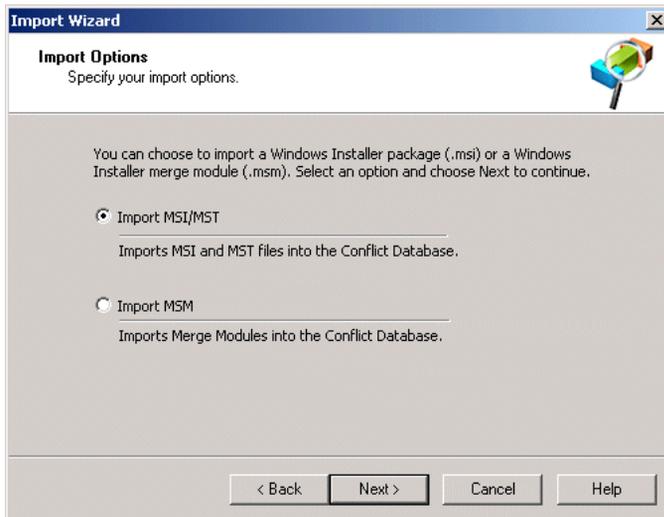


Figure 3-70.
Import wizard Import Options panel.

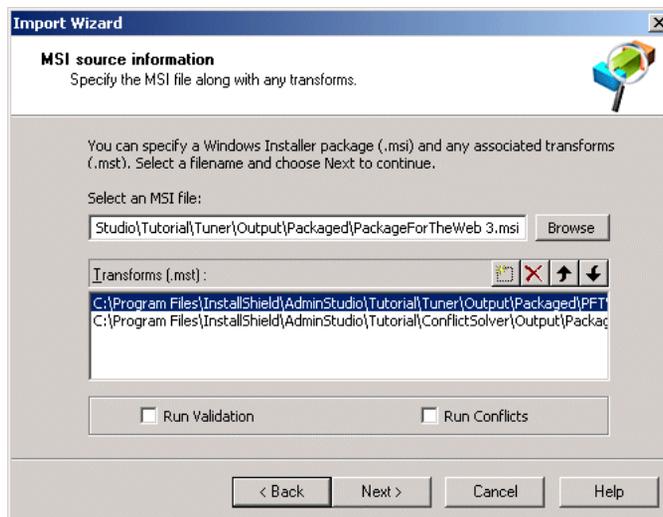
3. From the MSI Source Information panel, click Browse.
 - a. Navigate to <AdminStudio Directory>\Tutorial\Author\Output\PackageForTheWeb 3\Product Configuration 1\Release 1\DiskImages\DISK1 and select PackageForTheWeb 3.msi.
 - b. Click the New button in the Transforms area.
 - c. In the Transforms area, click Browse (...) and navigate to <AdminStudio Directory>\Tutorial\Tune\Output and select PFTW.mst.
 - d. In the Transforms area, click Browse (...) again and navigate to <AdminStudio Directory>\Tutorial\ConflictSolver\Output and select PackageForTheWeb 3.mst.



Figure 3-71.

Import wizard MSI Source Information panel.

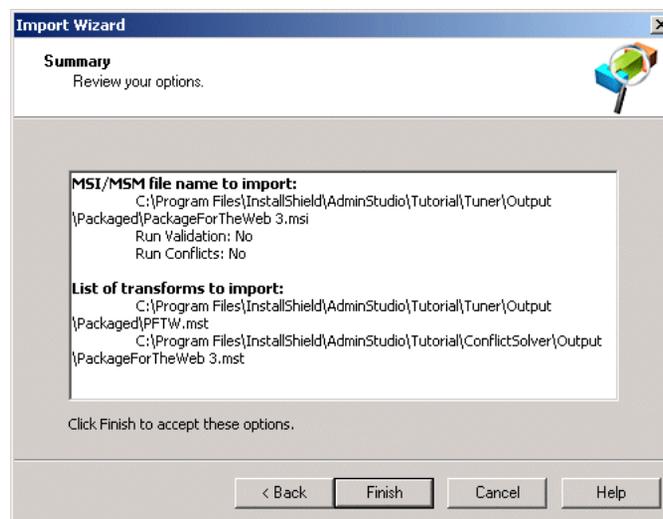
- e. Ensure the transform list has the PFTW.mst listed above the PackageForTheWeb 3.mst. If the ordering is incorrect, use the Move Up and Move Down buttons to change the order.



4. Click Next.
5. Review the options selected in the Summary panel and click Finish.

Figure 3-72.

Import wizard Summary panel.





The Import process displays its progress in the Output window's Import tab.

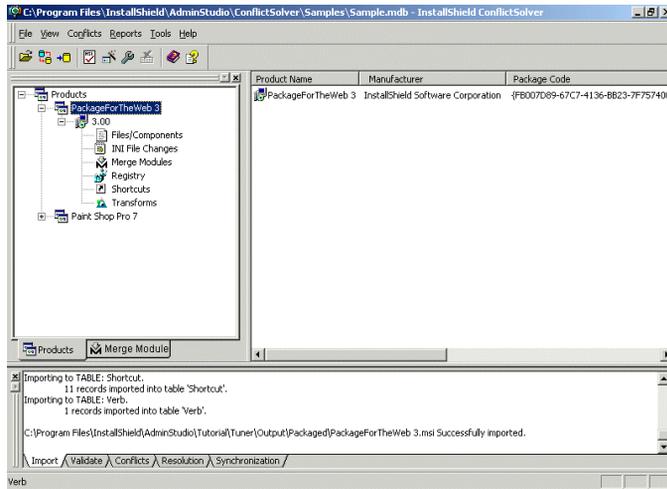


Figure 3-73.
The Import tab of the Output window.

Conclusion

This tutorial is by no means comprehensive; rather, it is designed as an introduction to the AdminStudio workflow for many administrators. It is entirely possible that some aspects of AdminStudio will be used daily, while others infrequently. The flexibility of AdminStudio ensures that—no matter what your needs are—you have the tools to address your predeployment needs.



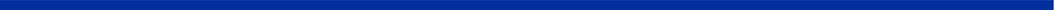


Frequently Asked Questions

Table of Contents

AdminStudio FAQ	4-3
How Do I Launch the AdminStudio Features?	4-3
How Do I Check for AdminStudio Updates? .	4-3
Why Does AdminStudio Look Different Than the Last Time I Used It?	4-3
Repackager FAQ.	4-4
Why Do People Use a Repackager?	4-4
What Can I Do with a Repackaged Setup? .	4-4
Why is a “Clean” System Important for Repackaging?	4-4
How Do I Repackage a Non-Windows Installer Setup?	4-4
How Do I Repackage a Windows Installer Setup?	4-4
How Can I Speed Up Repackaging?	4-5
How Can I Use the Same Repackager Settings Again?	4-5
Author FAQ	4-5
When Do I Need to Use Author?	4-5
How Do I Import and Customize a Repackaged Application?	4-5
How Do I Import and Customize an Existing MSI File?	4-6
How Can I Ensure the Target Machine Has the Windows Installer Service?	4-6
How Do I Specify Product or Component Default Destinations?	4-6
How Do I Make IDE Changes Take Effect When I Run My Setup?	4-7
How Do I Display a License Agreement? . . .	4-7
How Do I Create Shortcuts?	4-7

This chapter deals with some of the questions that InstallShield AdminStudio users most frequently have. If you have a question about InstallShield AdminStudio that is not answered here, check the InstallShield Knowledge Base, available on InstallShield’s support Web site. Your question may already be answered there.



- How Do I Associate File Types
with My Program? 4-8
- How Do I Launch an External Application
from My Setup Project? 4-8
- How Do I Conditionally Install My Setup
Based on the Target's Operating System? . . 4-8
- Tuner FAQ 4-9
 - When Do I Use Tuner vs. Author? 4-9
 - How Do I Add a File to a
Windows Installer Package? 4-9
 - How Do I Prevalidate a
Windows Installer Package? 4-9
 - What Should I Do If
MSI Prevalidation Fails? 4-10
 - How Do I Create a Setup.exe File
for My Package and Transform? 4-10
 - How Do I Prevent a Feature from
Displaying During Custom Installation? . . . 4-11
- ConflictSolver FAQ 4-11
 - What Types of Conflicts are Detected? 4-11
 - When Should I Import Merge Modules? . . . 4-11
 - How Do I Import a Package into
the Conflict Database? 4-11
 - How Do I Import a Merge Module into
the Conflict Database? 4-12
 - How Do I Identify Conflicts? 4-12
 - Identifying Conflicts with an
External Package 4-13
 - Identifying Conflicts Between
Existing Packages. 4-13
 - How Do I Validate the Package? 4-13
 - How Do I Change Which Conflicts
Are Checked? 4-14
 - How Do I Automatically Resolve Conflicts? . 4-14



AdminStudio FAQ

How Do I Launch the AdminStudio Features?

There are several different ways to launch the AdminStudio features:

- **From the Go menu**—Select the appropriate feature from the Go menu to launch it.
- **Via keyboard shortcuts**—Each feature has a keyboard shortcut to launch it. The shortcuts are:
 - Repackager: Ctrl+R
 - Authoring Environment: Ctrl+A
 - ConflictSolver: Ctrl+L
 - Customize: Ctrl+M
- **From the toolbar**—Each AdminStudio feature can be launched using a button on the AdminStudio toolbar.
- **From the Getting Started area on the main AdminStudio page**—Links to each AdminStudio feature are located under the Getting Started section on the right side of the main AdminStudio page.
- **From individual feature pages**—Each feature has a page associated with it in AdminStudio, from which you can get information and tips about that specific feature. These pages are accessed by clicking on the appropriate feature in the navigation bar to the left of the AdminStudio interface. On the right side of the page that appears is a shortcut to launch the appropriate feature.

How Do I Check for AdminStudio Updates?

InstallShield AdminStudio uses the InstallShield DigitalWizard Update Service to inform you of updates to the product. This happens automatically every 30 days. To manually check for updates, select Update AdminStudio from the Help menu.



Why Does AdminStudio Look Different Than the Last Time I Used It?

InstallShield AdminStudio contains functionality that allows it to update itself with the latest information from InstallShield, including tips about product use, or answers to frequently asked questions. When AdminStudio is launched, if there is an Internet connection, AdminStudio checks if there are any updates to the user interface. If there are, it downloads these in the background. The next time you launch AdminStudio, the new user interface is used.



Repackager FAQ

Why Do People Use a Repackager?

In a traditional environment, repackagers are used to customize an installation. For Windows Installer environments, repackagers such as InstallShield Repackager are used to convert traditional setups to Windows Installer packages.

What Can I Do with a Repackaged Setup?

Once you have repackaged a setup into a Windows Installer package, you can further configure it using Author. You can also create transforms from it, providing extensive customization for your specific needs. You can also check the package for conflicts that may exist between it and other known Windows Installer packages in the Conflict Database, ensuring the proper installation and functioning of your setups. Finally, once converted to a Windows Installer package, the setup can take advantage of Windows Installer functionality such as application repair and feature advertisement.

Why is a “Clean” System Important for Repackaging?

A clean system, which usually consists of the operating system only, is important when repackaging because you want to capture all additions and changes made by the installation. If you do not use a clean system, it is entirely possible that a previous installation may have placed files on your system that are also part of the installation you are repackaging. In that case, the installation may not install those files, and therefore Repackager never captures those files for the repackaged setup.

How Do I Repackage a Non-Windows Installer Setup?



To repackage a traditional, non-Windows Installer setup, do the following:

1. Install AdminStudio on a development machine.
2. On a clean system, launch isrepackager.exe over the network from the development machine.
3. Follow the directions in the Repackager wizard, storing the .inc file in a target directory on the development machine.
4. When the Repackager finishes, move to the development machine and launch AdminStudio.
5. Open Author, and select Open from the File menu.
6. Select the .inc file you created during repackaging. Follow the directions in the Import Repackaged Setup wizard. Ensure you create an MSI file at the end of the process.

How Do I Repackage a Windows Installer Setup?

Microsoft recommends that Windows Installer setups not be repackaged. Instead, use Tuner to create transforms.



How Can I Speed Up Repackaging?

Repackaging can be a time consuming process, because the way Repackager determines the changes made by an installation requires both an initial analysis and a final analysis. The difference between these constitutes the changes made by the installation. But there are ways to speed up repackaging. First, if you know what directories are modified by the installation, you can limit both initial and final analysis to those directories only. Also, you can set up a detailed exclusion list for files, INI files, and registry entries, instructing what Repackager should ignore during both analysis phases.

How Can I Use the Same Repackager Settings Again?

At times, it may be desirable to retain the same repackaging configuration for multiple repackaging projects. Repackager automatically saves the current configuration to the `isrepackager.ini` file, stored in the Windows directory. The next time Repackager launches, settings from this INI file are used by default.

You can also revert to the factory-installed settings for Repackager, stored in `default.ini` in the Repackager subdirectory of the AdminStudio directory. You can perform this reset from the Ready to Repackage panel.

If you customize the `default.ini` file, you can use it as your “default” settings for repackaging.

InstallShield strongly recommends you keep a backup copy of `default.ini` if you intend to make changes to it.



Author FAQ

When Do I Need to Use Author?

Most system administrators use Author to import repackaged setups and convert them into Windows Installer packages. Author is also ideal for making changes to the package that you want reflected in all deployments of the package. However, InstallShield recommends you use Tuner to create transforms for changes that you only want to affect a particular deployment, rather than every installation.

How Do I Import and Customize a Repackaged Application?

To import a repackaged application, do the following:

1. Launch AdminStudio and open Author.
2. Select Open from the File menu.
3. Navigate to the directory containing the script file (`.inc`) for the repackaged application.
4. When the Import Repackaged Setup wizard launches, click Next.





5. Provide project information for the repackaged setup, including the name and location of the project file. Click Next.
6. Enter summary stream information. Click Next.
7. Configure advanced options as necessary. Click Next.
8. Review the summary. If you want to build a Windows Installer package immediately after import, click the appropriate check box. Click Finish to convert the script file to an Author project.

The progress of the import appears in the Results tab of the Output window at the bottom of the interface. After import, use Author to further customize the package. You can then build the Windows Installer package.

How Do I Import and Customize an Existing MSI File?

You can import any .msi file into Author and customize it to your needs.



To open an .msi file, do the following:

1. Select Open from the File menu. A browse dialog opens for you to locate the setup package.
2. Set the Files of type filter to Windows Installer Packages (*.msi).
3. Navigate to the project that you would like to open.

The Open MSI/MSM wizard launches and prompts you for information about the new setup project.

How Can I Ensure the Target Machine Has the Windows Installer Service?

Although the Windows Installer service is built into Windows 2000 and Windows Me, other Windows platforms require the service to be installed before any Windows Installer setups can run on those systems. In the Release wizard you have the option of including a traditional setup in your project that will check if the Windows Installer service is installed on the target machine. If it is not installed, or it is an older version, Setup.exe will install Windows Installer. When finished, Setup.exe will launch your setup project. If you are not sure which platforms your end users have, it is a good idea to include Setup.exe and the Windows Installer engine for each operating system with your setup package.

How Do I Specify Product or Component Default Destinations?

The Product's default destination can be set in the Product Properties, located in the Project view. All of your components default to [INSTALLDIR], which is a property that stores the path that you provide in the product property INSTALLDIR. Your component destinations can be changed in the component's property sheet, which is located in the Setup Design view. You can build upon INSTALLDIR, thereby installing your component to a subfolder of the default installation folder, or you can specify an entirely new path. If you would like to have your component



installed to a subfolder of INSTALLDIR called Files, enter [INSTALLDIR]\Files into the component's Destination property.

How Do I Make IDE Changes Take Effect When I Run My Setup?

After you make changes to your setup project, you must build your setup in order for the changes to take effect. To rebuild your project, you can either run through the Release wizard, or click the Build button. The Build button will rebuild your release, using the same settings that were selected the last time you used the Release wizard. If you have not built your release previously, your release will be built with default settings.

Alternately, you can perform a quick build if no files have been changed. The two commands available from the Build menu for quick builds are:

- **Build Tables Only**—Only MSI tables are built. If you have not built this setup previously, a new MSI file is created, but no files are added to the setup. If you have built your setup already, the MSI file is updated when all the tables are built, but no files are transferred.
- **Build Tables & Refresh Files**—Rebuilds the MSI file and updates the Files table, capturing all new or changed files in your setup. This type of build can only be run after a full build is performed. Setup files are not transferred to a new location after this type of build.

How Do I Display a License Agreement?

A license agreement dialog is included as part of the default user interface. All you need to do is associate a rich text (.rtf) file with this dialog.

To add your text to the license agreement, do the following:

1. Navigate to the Dialogs view under Step 4: Design the User Interface.
2. Double-click the License Agreement dialog in the User Interface explorer to expand it.
3. Click the language version of the license agreement that you would like to edit. In most cases this is English (United States).
4. The Dialog editor opens, displaying the License Agreement dialog. Click the text box on the dialog to display its properties to the right.
5. Click the File Name property and enter the path to your license text, or click the Browse button (...) to navigate to your file.



How Do I Create Shortcuts?

Shortcuts allow your end users quick access to your product.

To create a shortcut, do the following:

1. Navigate to the Shortcuts/Folders view under Step 3: Configure the Target System. The Shortcuts explorer opens to the right.
2. Right-click on the directory in the Shortcuts explorer where you would like your shortcut to be placed. Select New Shortcut from the list.





3. Give your new shortcut a meaningful name by selecting it, pressing the F2 key, and typing the new name. This name is for your reference only and is not displayed to the end users.
4. Set the shortcut's properties. These properties include the display name, the shortcut's target, and any command-line arguments that you would like to pass to the target.

How Do I Associate File Types with My Program?

If your program creates or relies upon custom file types, you can associate those file types with your program on the target machine. File types can be set up in the File Types advanced setting. Navigate to the Setup Design or Components view and expand the Advanced Settings folder.



To create a file association, do the following:

1. Right-click the ProgID folder in the Shortcuts explorer and select New ProgID.
2. Give your new progID a meaningful name by selecting it, pressing the F2 key, and typing the new name. An example of a good name for your progID would be to use HTMLFile for any file that should be opened with an HTML viewer.
3. Right-click the Extensions folder in the Shortcuts explorer and select New Extension.
4. Rename this extension to the actual extension of the file type, without the dot. For example, you should enter htm for HTML files.
5. Click the progID property in the property sheet to the right and select the progID that you would like to associate this file type with.

How Do I Launch an External Application from My Setup Project?

With the use of custom actions, you can launch executable files or call functions from DLLs, JScript, or VBScript. You must first create a custom action, then enter that action into a sequence. Custom actions can be created with the Custom Action wizard, located in the Actions/Scripts view. That action will then be available in the Sequences view, where you can insert it into one or more sequences.

How Do I Conditionally Install My Setup Based on the Target's Operating System?

If you would only like your setup to run on certain operating systems, say Windows 2000, you can set an install condition. If this condition evaluates to false, the product will not be installed.



To set the install condition, do the following:

1. Navigate to the General Information view under Step 1: Organize Your Setup.
2. Click the Product Properties icon in the Project explorer to view the property sheet.



3. Select Install Condition from the property sheet to view the Condition Builder. The Condition Builder helps you create your condition by providing the proper syntax for you.

Tuner FAQ

When Do I Use Tuner vs. Author?

Author is used when converting the .inc file resulting from repackaging into a Windows Installer package. Any modifications necessary to the base package can be made from within Author. It is also used to create new Windows Installer packages.

Tuner is used to create transforms from the base package, while keeping the base package intact. You can use Tuner to add files to a project, control the features installed, and modify information within the base package without affecting the package for other installations.

Put simply, Author is for creating and modifying the base package, and Tuner is for creating customized installations from the base package without altering the original package.

How Do I Add a File to a Windows Installer Package?

To add a file to a Windows Installer package, do the following:

1. With your transform project open, select Files from the checklist.
2. In the Destination computer's folders pane, select the folder in which you want to add the file. If the folder you want is not visible, you can either right click the Destination Computer node and show the appropriate folder, or you can right click the Destination Computer node and add the folder.
3. Once the correct destination folder is selected, locate the folder containing the file you want to add in the Source computer's directory tree.
4. Drag the file from the Source computer's files to the Destination computer's files.



How Do I Prevalidate a Windows Installer Package?

To prevalidate a Windows Installer package, do the following:

1. Within Tuner, select Prevalidation from the checklist.
2. Select the Evaluation file you want to use. In most cases, this is the Full MSI Validation Suite.
3. Select the Internal Consistency Evaluators (ICEs) you want to run. Typically, you will want to use all ICEs. Should you want to run particular ICEs, specify them separated by semicolons (;).
4. Select the messages you want to display. Info messages provide information about the ICEs as they are processed. Warnings are potential problems with the Windows Installer package, but are still acceptable in a valid Windows Installer package. If any Errors are found, the package fails validation. InstallShield recommends you always display Error messages.





5. Click Start to begin validation. Progress appears in the MSI Validation tab of the Output window.

What Should I Do If MSI Prevalidation Fails?

Sometimes, the Windows Installer package you plan on customizing fails Tuner's prevalidation test—meaning that Errors or Warnings appear in the MSI Validation results. If this happens, there are a few possible courses of action:

- **Reconsider installing the package**—You may decide you do not want the hassle of unexpected results in installing the base package, and you certainly do not want to waste your time customizing something that may not work even without the customization.
- **Continue with the customization**—Even if you know the base Windows Installer package is invalid, you are willing to take the chance that it will still work. And it might. The validation failure might be in an inconsequential area or one that doesn't affect the transform you are creating.
- **Fix the problems using Tuner and then customize the package**—Tuner provides you with the tools to fix issues within the source Windows Installer Package, whether it be through individual views or through the Table Editor. Once you have corrected problems in the existing package, you can continue customization for your own purposes.
- **Contact your software vendor**—You can inform the vendor of not only the fact that the Windows Installer package is invalid, but also which internal consistency evaluation(s) it failed. If you click the Copy Results button, you can copy the entire results file to the clipboard, from which it can easily be pasted into an email message.

How Do I Create a Setup.exe File for My Package and Transform?



To create a Setup.exe file to begin the installation of your base package and transform, do the following:

1. From the checklist, select Package.
2. From the Package tree, select Setup.
3. Select the Create Installation Launcher (Setup.exe) check box.
4. Specify whether you want to include the Windows 95/98 or Windows NT MSI engines.
5. Provide any command line arguments for your installation.
6. Save your transform. The Setup.exe file is stored in the directory specified in the Location panel of the Packaging wizard, or in the Location view within the Package view.



How Do I Prevent a Feature from Displaying During Custom Installation?

To prevent a feature from being displayed to your end users during a custom installation, do the following:

1. Select the Features view from the checklist.
2. From the Features tree, select the feature you want to hide.
3. Change the Visible property to Not Visible

The feature selected will not be visible during custom installation. Depending on the feature's Initial State, the feature may or may not be installed on the end user's system.



ConflictSolver FAQ

What Types of Conflicts are Detected?

ConflictSolver has rules to detect conflicts involving:

- Components
- Files
- Registry Entries
- Shortcuts
- INI Files
- ODBC Resources
- NT Services
- File Extensions
- Product Codes

When Should I Import Merge Modules?

For optimal performance, merge modules (.msm) should be imported into the Conflict Database prior to importing Windows Installer packages. This ensures that conflicts resulting from not using available merge modules are correctly identified.

How Do I Import a Package into the Conflict Database?

To import a Windows Installer package into the Conflict Database, do the following:

1. Select Import from the File menu. You can also right-click on the Products node of the Products tree and select Import from the context menu.
2. When the Import wizard launches, click Next.
3. From the Import Options panel, select Import MSI/MST. Click Next.
4. From the MSI Source Information panel, click Browse to locate the Windows Installer package (MSI) you want to import.





5. Should the package require transforms, click the New button above the Transforms area. Use the Browse button (...) to locate the transform. If the package requires multiple transforms, you can repeat the procedure as necessary. The order in which transforms are applied can be changed by selecting a transform and clicking the Move Up and Move Down buttons. If you need to delete a transform you have added, select the transform and click the Delete button.
6. Click Next.
7. Review the information in the Summary panel. If you are satisfied with the import options, click Finish to start the import.

A report of the import process appears on the MSI Import tab of the Output window. Depending on whether options have been set from the General tab of the Options dialog (available from the Tools menu), validation and conflict checking may be performed during import.

The Windows Installer package appears under the Products tab in the Project Workspace.



Although not required, InstallShield recommends you import only Windows Installer packages that have passed the Full MSI Validation Suite validation into the Conflict Database. This is the only way to ensure that conflicts identified between different packages do not result from issues resulting from incorrect installation creation.

How Do I Import a Merge Module into the Conflict Database?



To import a merge module into the Conflict Database, do the following:

1. Select Import from the File menu.
2. When the Import Wizard launches, click Next.
3. From the Import Options panel, select Import MSM. Click Next.
4. Click Add and browse to the location of the Merge Module file(s) you want to import. Select the Merge Module(s) and click Open.
5. Click Next.
6. Review the information in the Summary panel. If you are satisfied with the import options, click Finish to start the import.

A report of the import process appears on the MSI Import tab of the Output window. The Merge Module appears under the Merge Modules tab in the Project Workspace.

How Do I Identify Conflicts?

The Conflict Wizard can be used to identify conflicts between an external Windows Installer package and ones already in the Conflict Database, or between one or more packages in the Conflict Database.



Identifying Conflicts with an External Package

To identify conflicts between an external Windows Installer package and the Conflict Database, do the following:



1. Select Conflict Wizard from the Conflicts menu.
2. Click Next from the Welcome panel.
3. From the Source Types panel, select Windows Installer package. Click Next.
4. From the MSI Source Information panel, click Browse to locate the MSI file you want to check against packages in the Conflict Database.
5. Should that package require transforms, click the New button above the Transforms area. Use the Browse button (...) to locate the transform. If the package requires multiple transforms, you can repeat the procedure as necessary. The order in which transforms are applied can be changed by selecting a transform and clicking the Move Up and Move Down buttons. If you need to delete a transform you have added, select the transform and click the Delete button.
6. Click Next.
7. From the Target Information panel, select the packages in the Conflict Database you want to compare the package against. Click Next.
8. From the Conflict Options panel, select the types of conflicts you want to identify. If you want to run the Resolution Wizard or run reports when the Conflict Wizard finishes, select the appropriate options. Click Next.
9. Review the options selected in the Summary panel. Click Finish to begin the conflict identification process. Progress appears in the MSI Conflicts tab of the Output window.

Identifying Conflicts Between Existing Packages

To identify conflicts between packages already in the Conflict Database, do the following:



1. Select Conflict Wizard from the Conflicts menu.
2. Click Next from the Welcome panel.
3. From the Source Types panel, select Internal Windows Installer package. Click Next.
4. From the Source Package panel, select the package you want to compare against other packages in the Conflict Database. Click Next.
5. From the Target Information panel, check all the packages you want to compare the source package against. Click Next.
6. From the Conflict Options panel, select the types of conflicts you want to identify. If you want to run the Resolution Wizard or run reports when the Conflict Wizard finishes, select the appropriate options. Click Next.
7. Review the options selected in the Summary panel. Click Finish to begin the conflict identification process. Progress appears in the MSI Conflicts tab of the Output window.

How Do I Validate the Package?

To validate a Windows Installer package, do the following:





1. Select Validate from the Conflicts menu.
2. Browse to the Windows Installer package and click OK. This begins validation against the default validation file. Output appears in the MSI Validation tab of the Output window.

You can also perform validation when importing a package using the Import wizard.



Should the package contain validation errors, you may be able to correct them using InstallShield Tuner or Author.

How Do I Change Which Conflicts Are Checked?



To specify the default conflict types checked, do the following:

1. From the Tools menu, select Options.
2. In the Options dialog, click on the Conflicts tab.
3. In the Conflict Types area, select the conflict types you want ConflictSolver to identify by default.
4. Click OK.

When you run the Conflict wizard, the default types specified in the Options dialog are already checked in the Conflict Options panel. You can override these default types from this panel.

How Do I Automatically Resolve Conflicts?

After running the Conflict wizard, or after importing a package while conflict checking is enabled, ConflictSolver may identify conflicts. If it does, run the Resolution wizard to attempt to resolve conflicts automatically. You can only resolve conflicts between external Windows Installer packages and packages already in the ConflictSolver Database.



To use the Resolution wizard to resolve conflicts, do the following:

1. After identifying conflicts, select Resolution Wizard from the Conflicts menu.
2. When the Resolution wizard launches, the Welcome panel is displayed. Click Next.
3. From the Conflicts Identified panel, select the check boxes to the left of all conflicts listed in bold type. These conflicts can automatically be resolved by ConflictSolver.
4. Click Resolve to begin resolution.



If conflicts are discovered that cannot be resolved using ConflictSolver, use Author or Tuner to make corrections.



Glossary

Action

The Windows Installer service works through a series of specific executions called actions. There are basically two types of actions: built-in actions, such as file transfer or checking for disk space; and custom actions, such as launching an EXE or calling a function in a DLL.

Administration Sequence

The Administration sequence is the list of actions that are executed when you launch the setup with the /a command-line option. This is useful for network administrators who would want to provide a common installation point on the network and prepare different installation criteria for multiple users. Each Administration sequence consists of a User Interface sequence and an Execute sequence.

Advertisement

Advertisement is a type of “just-in-time” installation in which features are installed when they are requested from the installer and not installed immediately during setup.

Advertising

Advertising is one of the modes you can run your setup in. When you launch msixexec.exe with the /j <package> option, the components are advertised on the end user’s system, but not immediately installed.

Advertisement Sequence

The advertising sequence contains the list of actions that are executed when you launch the setup with the /j command-line option.

App Paths

The registry key that Windows uses to find your application and its DLLs if their locations are not already in the system’s path. App Paths can be set through the Advanced Settings folder.

Assigning

Assigned features have their shortcuts installed, and they can be installed from the Add/Remove Programs applet. But in fact, an assigned feature remains advertised until a user requests it.



COM Server

An executable file that exposes objects to other applications according to the component object model specification. The types of objects exposed by a COM server may include ActiveX controls, ActiveX documents, Automation objects, or MTS components.

COM File

A COM file contains reusable code that can be shared across products. A popular example of a COM file is an Excel spreadsheet when it is embedded in a Word document. In this example, Excel acts as the COM server, and Word acts as the client.

Context Menu

A context menu, also known as a right-click menu or a popup menu, displays when a user right-clicks on an item on the desktop, in Windows Explorer, or in an application.

Converted Projects

The Open MSI/MSM wizard converts .msi files to .ism files. To launch the wizard, select Open from the File menu and set the filter to *.msi files.

Dynamic Link Library (DLL)

A shared code-base file containing functions that can be called from other applications.

End User

The term end user, in InstallShield documentation, refers to the customer who will be installing your product onto his or her computer.

Environment Variables

Environment variables are a set of paths that are stored under the Environment tab of the System Properties dialog. To view these variables, right-click the My Computer icon on your shortcut and click on the Environment tab, or type “set” at a DOS prompt.

Feature

Parts of a setup from an end user’s perspective, such as a help file, a clip art package, or program files.



Globally Unique Identifier (GUID)

A globally unique identifier is a long string of numbers created to uniquely identify your product, component, package, COM class, and so on, from others. Enter string GUIDs throughout the IDE in the format {XXXXXXXX -XXXX-XXXX-XXXX-XXXXXXXXXXXX}.

Installation Sequence

The installation sequence runs by default when the setup is launched, for example, by double-clicking on an .msi file.

.inc File

An .inc file is the script file outputted by Repackager. It contains information about all changes made to the target computer system during repackaging. This file is necessary to generate a Windows Installer package following repackaging.

.ism File

The .ism file is the working file that Author uses to store your project information. When you build a release, Author uses the .ism file to create an .msi file for distribution.

Key File

A unique file for each component that Windows Installer uses to detect the component's presence. In order to create advanced component settings or shortcuts, a key file must be specified.

Key Path

A unique registry value for each component that Windows Installer uses to detect the component's presence. A component can have either a key file or a key path, but not both.

Load-Ordering Group

One service may require another service to already be running before it starts. For this reason, services need to be grouped and set to load in a specific order. Service load-ordering groups are listed under KEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Service-GroupOrder. The service's Start Type property determines when it loads within its group.



Merge Module

A merge module is a package containing all of the logic and files needed to install distinct pieces of functionality such as run-time DLLs and virtual machines. Merge modules are built once and can be added to any setup project. For detailed information on merge modules, see the online help under Merge Modules and Merge Module Authoring.

.msi File

The .msi file, created by Author, is the setup package in its finished state. It includes setup resources files and can have compressed within it all of the application's data files. The .msi file is the one that is distributed to the end users and will interact with the Windows Installer service in order to install your application on their machines.

.ocx File

An .ocx file is a cross-platform COM file that is called by an application to perform a desired function. Based on OLE and COM technologies.

Publishing

Publishing is a type of advertising (“just-in-time” installation) in which no user-interface elements are created for the component during installation, but the component can still be installed through the Add/Remove Programs applet of the Control Panel or when an installed component requests the published component from the installer.

Product

Refers to the actual application or collection of files that is to be installed.

Project

Refers to the entire collection of source files and the .ism file that make up your setup while it is under construction.

Reference Count

A reference count is a tally that is incremented each time a shared file is installed. It is maintained under HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\SharedDLLs.



Registry or REG file

A text file of a predefined format that contains keys and values that can be merged into a registry.

Reinstallation

When a product has already been installed on a machine and its setup is run again, the setup will reinstall the product by overwriting its existing files, shortcuts, and registry entries.

Rollback

The Windows Installer service keeps track of all changes that are made during the installation process so that, if an error occurs and the installation is aborted, the changes will be rolled back and the machine will be restored to its original state.

Run Time

The time during which the .msi file interacts with the Windows Installer service to install your application on the target machine.

Sequences

A collection of actions and dialogs that is executed sequentially. See also Installation Sequence, Advertise Sequence, and Administrator Sequence.

String Table

A database that maintains the string IDs, values, and comments for a specific language.

Uninstallation

The undoing of an installation. Uninstallation is the installation maintenance option that allows the end user to remove the product files and reverse any changes to the machine made during the original installation.



Index

Symbols

.inc file
opening 3-11

A

ACEs 2-13
descriptions 2-13
types 2-13
AdminStudio
checking for updates 4-3
launching features 4-3
user interface 4-3
what to do with repackaged setups 4-4
Application Conflict Evaluators 2-13
application repair 2-5
Author
conditional installations 4-8
creating shortcuts 4-7
file types 4-8
importing existing Windows Installer packages 4-6
importing repackaged application into 4-5
launching external application from setup 4-8
launching from AdminStudio 4-3
when to use 4-5
when to use vs. Tuner 4-9
authoring installations 2-7

B

build
quick 4-7

C

clean system 4-4
COM files
registering 2-8
COM+ files
registering 2-8
component
default destination 4-6

conditional installations 4-8
Conflict Database
importing merge modules 4-12
importing packages 4-11
conflict types 4-11
conflicts
determining which checked 4-14
identification and resolution 2-12
identifying 4-12
identifying between existing packages 4-13
identifying between external package
and Conflict Database 4-13
resolving automatically 4-14
types detected by ConflictSolver 4-11
ConflictSolver 4-12
conflict types detected 4-11
determining which conflicts are checked 4-14
identifying conflicts 4-12
identifying conflicts between existing packages 4-13
identifying conflicts with external package 4-13
importing packages 4-11
launching from AdminStudio 4-3
merge modules 4-11
resolving conflicts automatically 4-14
validation 4-13
conventions 1-3
note 1-3
procedure 1-3
tip 1-3
warning 1-3
customization
installations 2-9
steps 2-11

D

DCOM files
registering 2-8
default destination
specifying product or component 4-6
default.ini 4-5



DemoShield xxi
display user interface 2-8
document conventions 1-3

E

external package
 identifying conflicts between Conflict Database 4-13

F

feature advertisement 2-4
features
 launching in AdminStudio 4-3
features and benefits xv
file associations
 registering 2-8
file transfer 2-7
file types
 associating with programs 4-8
files
 adding to Windows Installer package in Tuner 4-9

H

help resources xxii

I

import
 merge modules into Conflict Database 4-12
importing merge modules into 4-12
installation
 customization 2-9
 package 2-10
installation customization 2-9
installation defined 2-7
installation package 2-10
installation procedure 1-4
installations
 conditional 4-8
InstallShield

about xviii
Consulting Services xxv
Customer Care xxiii
DemoShield xxi
Developer xix
Express xx
Knowledge Base xxii
Multi-Platform Edition xx
newsgroups xxiii
products xix
Professional xix
Professional Services xxv
Training xxvi

InstallShield Consulting Services xxv
InstallShield Developer xix
InstallShield Express xx
InstallShield Multi-Platform xx
InstallShield Professional xix
InstallShield Professional Services xxv
InstallShield Training xxvi
interface
 AdminStudio 2-6
isrepackager.ini 4-5

K

Knowledge Base xxii

L

license agreement
 displaying 4-7
live pages 4-3

M

Merge Modules 2-4
merge modules 2-4
 importing into Conflict Database 4-12
 when to import into ConflictSolver 4-11
modifying installations 2-7
MSI files 2-10

MST 2-10
relationship to MSI 2-10

N

newsgroups xxiii

O

online help xxii
help library xxii
operating system
installing conditionally based on version 4-8

P

package
identifying conflicts between external
and Conflict Database 4-13
importing into Conflict Database 4-11
installation 2-10
packages
identifying conflicts between existing 4-13
validating using ConflictSolver 4-13
Windows Installer 2-10
predeployment process 2-5
prevalidation 4-9
handling failure 4-10
product
default destination 4-6
products xix
DemoShield xxi
installation development xix
InstallShield Developer xix
InstallShield Express xx
InstallShield Multi-Platform xx
InstallShield Professional xix
Professional Services xxv
Consulting xxv
Training xxvi

Q

quick build 4-7
Build Tables & Refresh Files command 4-7
Build Tables Only command 4-7

R

register 1-4, 4
register product for uninstallation 2-8
repackaged setups 4-4
Repackager
increasing repackaging speed 4-5
launching from AdminStudio 4-3
reusing settings 4-5
repackagers
reasons to use 4-4
repackaging 2-7
clean system 4-4
increasing speed of 4-5
traditional setup 4-4
Windows Installer setup 4-4
requirements
system 1-3
rollback 2-4

S

self-repair 2-5
setup
launching external application from 4-8
repackaging traditional 4-4
repackaging Windows Installer 4-4
setup defined 2-7
setup elements 2-7
creating shortcuts 2-8
display user interface 2-8
file transfer 2-7
register file associations 2-8
registering COM, COM+, and DCOM files 2-8
registering product for uninstallation 2-8



setup.exe
 creating for package and transform 4-10

setups
 repackaged 4-4
 traditional vs. Windows Installer 2-3

shortcuts
 creating 2-8
 creating in Author 4-7

software registration 1-4

support
 contacting xxiv
 context-sensitive help xxii
 Customer Care xxiii
 help library xxii
 Knowledge Base xxii
 newsgroups xxiii
 online help xxii
 resources xxii
 Web-based help xxii

system requirements 1-3

T

target machine
 ensuring presence of Windows Installer 4-6

target system
 conditionally installing based on operating system 4-8

technical support xxii

traditional setup
 repackaging 4-4

transform 2-10

Tuner
 adding files to Windows Installer packages 4-9
 creating setup.exe for package and transform 4-10
 launching from AdminStudio 4-3
 prevalidation 4-9
 when to use vs. Author 4-9

tutorial
 adding files 3-22
 launching Repackager remotely 3-3

building the release 3-16

configuring ConflictSolver options 3-31

configuring Windows 2000
 Add/Remove Programs settings 3-24

creating a transform 3-18

creating shortcuts 3-23

customization 3-20

customizing an installation using Tuner 3-18

default destination and organization 3-20

identifying and resolving conflicts 3-30

identifying conflicts 3-34

importing the file 3-42

modifying and building
 a Windows Installer installation 3-11

modifying the project 3-15

opening ConflictSolver 3-30

opening the .inc file 3-11

packaging 3-26

postvalidation 3-25

preparation 3-3

prevalidation 3-19

repackaging a traditional installation 3-4

resolving conflicts 3-37

validating the package and transform 3-32

U

uninstallation
 registering product for 2-8

updates
 checking for AdminStudio 4-3

V

validation 2-11
 ConflictSolver 4-13

W

Windows Installer 2-3
 benefits of 2-3
 ensuring target machine has service 4-6

Windows Installer package



prevalidating 4-9
Windows Installer packages 2-10
 adding files to with Tuner 4-9
 identifying conflicts between existing 4-13
 importing into Conflict Database 4-11
 validating in ConflictSolver 4-13
Windows Installer setup
 repackaging 4-4

