Radia Extensions for Windows Installer

Radia Extensions for Windows Installer Guide

Software Version: 4.0

for the Windows operating system

Manufacturing Part Number: T3424-90052

August 2004

© Copyright 2004 Hewlett-Packard Development Company, L.P.

Legal Notices

Warranty

Hewlett-Packard makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company
United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Copyright Notices

© Copyright 1998-2004 Hewlett-Packard Development Company, L.P.

No part of this document may be copied, reproduced, or translated into another language without the prior written consent of Hewlett-Packard Company. The information contained in this material is subject to change without notice.

Trademark Notices

Linux is a registered trademark of Linus Torvalds.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Acknowledgements

PREBOOT EXECUTION ENVIRONMENT (PXE) SERVER
Copyright © 1996-1999 Intel Corporation.

TFTP SERVER
Copyright © 1983, 1993
The Regents of the University of California.

OpenLDAP
Copyright 1999-2001 The OpenLDAP Foundation, Redwood City, California, USA.
Portions Copyright © 1992-1996 Regents of the University of Michigan.

OpenSSL License
Copyright © 1998-2001 The OpenSSLProject.

Original SSLeay License
Copyright © 1995-1998 Eric Young (eay@cryptsoft.com)

DHTML Calendar
Copyright Mihai Bazon, 2002, 2003

Technical Support

Please select Support & Services from the following web site:
<http://www.hp.com/managementsoftware/services>

There you will find contact information and details about the products, services, and support that HP OpenView offers.

The support site includes:

- Downloadable documentation
- Troubleshooting information
- Patches and updates
- Problem reporting
- Training information
- Support program information

Preface

About this Guide

Who this Guide is for

The Radia Extensions for Windows Installer assists everyone in your organization responsible for packaging Windows-based applications, as well as testing, integrating, deploying, troubleshooting, and managing them.

Administrators or others responsible for application integration testing greatly benefit from the Windows Installer Packaging features as well as the Install Analysis and Profile State Analysis features. End users and business units that rely on rapid application turnaround and deployment benefit by the shortened time to deployment, eliminating the time it takes to resolve conflicts between applications that are frequently the bottlenecks to deployment. These conflicts are easily isolated and resolved using the Radia Extensions for Windows Installer in conjunction with the Radia Knowledge Base Manager and Radia Configuration Analyzer.

What this Guide is about

The *Radia Extensions for Windows Installer Guide* contains step by step instructions for each Radia Extensions for Windows Installer component, including how to:

- create, install, upgrade, and manage a package using the Radia Packager for Windows Installer
- use the optional wizards, including the State Wizard, the Library Wizard, the Package Wizard, and the Install Wizard
- create and use filters
- use the Advanced Options of the Package Wizard
- create a Merge Module

I-Lab and Radia Extensions for Windows Installer Publications

Radia Extensions I-Lab users should use the Radia Extensions for Windows Installer publications for any Windows Installer packaging information they may require. Please note, however, that this document contains information for the Radia Extensions for Windows Installer, which licenses more functionality than the Radia Extensions I-Lab. The following Radia Extensions for Windows Installer features covered in this document are only licensed in the Radia Extensions I-Lab product for testing purposes within a laboratory environment:

- Advanced Features
- Application Insulation
- Application Insight

Summary of Changes

This printing of the *Radia Extensions for Windows Installer Guide* contains the following changes to product names, information, and procedures for the following chapters.

Product Name Changes

- The Radia Advanced Publisher is now called the **Radia Packager for Windows Installer**, to reflect its main packaging function.

> **Note**
>
> See the *Radia Publisher Guide* for details on the new Radia component used to publish Windows Installer based applications to the Radia Database.

- The Radia Application Analyzer is now called the **Radia Configuration Analyzer**. See the *Radia Configuration Analyzer Guide* for installation and usage information.
- The Radia Application Knowledge Base (AKB) is now called the **Radia Knowledge Base** of the **Radia Knowledge Base Manager**. See the *Radia Knowledge Base Manager Guide* for installation and usage information.

Chapter 1: Introduction

4.0 Page 19, *About Packaging and Publishing with Radia*: new topic summarizing new or renamed products for packaging and publishing in Radia 4.0.

4.0 Page 21, *Starting Radia Packager for Windows Installer*: new topic explains how to start the product from the desktop icon or program group, and shows the new initial menu.

4.0 Page 21, *About the Unified Package Creation Process*: new topic discusses when to use the Typical option on the initial menu of the Radia Packager for windows Installer.

4.0 Page 22, *About the Custom Session Options*: new topic discusses when to use the Custom options from the initial menu of the Radia Packager for windows Installer.

4.0 Page 24, *About the Radia Packager for Windows Installer Process Menu*: new topic explains how to access the process-based Radia Packager for WI Menu.

Chapter 2: Creating a Package

4.0 Page 25, *Publishing a Package to Radia*: new topic refers users to the Radia Publisher as the tool needed to promote Windows Installer packages to the Radia Database. See the *Radia Publisher Guide*.

4.0 Page 28, *Creating a Package Using the Typical Session Option*: new topic and procedure to create a package. Creates a package in a single, streamlined session.

Chapter 3: Managing an Installation Package and Transforms

4.0 Chapter title was renamed from "*Installing a Package"* to "*Managing an Installation Package and Transforms"* to reflect the changed role of the Radia Install Wizard in relation to the new Radia Publisher.

4.0 Page 49, *Managing an Installation Package and Transforms*: deleted references to Radia Bundles and Package Signing Certificates. The Manage Radia Bundles and Manage Package Signing Certificate actions were removed from the Install Wizard due to diminished demand.

4.0 Page 50, *Radia Install Wizard*: deleted references to Radia Bundles and Package Signing Certificates. The Manage Radia Bundles and Manage Package Signing Certificate actions were removed from the Install Wizard due to diminished demand.

4.0 Page 75, deleted previous topics related to *Bundling* and *Package Signing*. These actions have been dropped from the Install Wizard.

Relocation Note

The *Using the Radia Application Analyzer* chapter has been relocated. See the *Using the Radia Configuration Analyzer* chapter in the *Radia Configuration Analyzer Guide*.

Chapter 4: Upgrading a Package

4.0 Page 78, *Packaging a New Release*: revised procedure to show access from the **Package components** option of the Radia Packager for Windows Installer.

4.0 Page 88, *Packaging a New Build with a Patch Package*: revised procedure to show access from the **Package components** option of the Radia Packager for Windows Installer.

Chapter 5: Using Advanced Functions

4.0 Page 100, *Using Radia State Wizard Advanced Functions*: added text:

"To access these functions from the Radia Packager for Windows Installer, select the Custom option **Identify resources to package**."

Modified all procedures in this topic to include new access to the Radia State Wizard from the Radia Packager for Windows Installer selection **Identify resources to package**.

4.0 Page 117, Using Radia Library Wizard Advanced Functions: added text:

"To access these library functions from the Radia Packager for Windows Installer, select the Custom option **Refine selected components**."

Modified all procedures in this topic to include new access to the Radia Library Wizard from the Radia Packager for Windows Installer selection **Refine selected components**.

4.0 Page 123, *Figure 5.27 ~ Radia Library Wizard – Add Replacement Variable*: updated to show Radia variables are now a supported replacement variable type.

4.0 Page 167, Using Radia Package Wizard Advanced Functions: added text:

"To access these functions from the Radia Packager for Windows Explorer, select the custom option **Package components**."

Modified all procedures in this topic to include new access to the Radia Package Wizard from the Radia Packager for Windows Installer selection **Package components**.

Appendix A: About the Radia Packager for Windows Installer Settings

4.0 Page 227, *Appendix A: Radia Packager for Windows Installer Settings*: revised all figures to shown new product name. Added text:

"To access the configurable settings, select any **Custom** action from the Radia Packager for Windows Installer and click the **Settings** button located on the next dialog."

4.0 Page 232, *Paths (3) Tab*: the Bundling Path text box was removed from this Settings tab. Bundling is no longer supported.

Appendix B:
Radia Package Wizard Advanced Options

4.0 Page 247, *Appendix B: Radia Package Wizard Advanced Options,* added text:

"The Radia Packager Wizard includes an **Advanced Options** menu that gives you easy access to the common settings and various types of information placed in your Windows Installer packages, as shown in Figure B.1 on page 248.

These same settings are also available from the **Windows Installer Options** dialog of the Radia Packager for Windows Installer, as shown Figure B.2 on page 248; this dialog is presented at the end of a Unified packaging session.

The only difference between the Advanced Options menu and the Windows Installer Options dialog is the grouping of the options. "

4.0 Page 248: *Appendix B: Radia Packager Wizard Advanced Options*, added Figure B.2 ~ Radia Packager for Windows Installer, Windows Installer Options and Table B.1 Windows Installer Options, by Tab Location.

Editorial Improvements

In addition to the changes listed above, this version contains various editorial, organizational, and style updates to each chapter and section and the index.

Contents

1

Introduction

At the end of this chapter, you will:

- Understand the tools used to package and publish applications with Radia.
- Understand the components of the Radia Extensions for Windows Installer.
- Understand the flow of building packages with the Radia Packager for Windows Installer.
- Be able to start the Radia Packager for Windows Installer and select the appropriate Typical and Custom option according to your packaging needs.
- Be aware of the alternate Radia Packager for Windows Installer Process menu.

The Radia Extensions for Windows Installer Components

The Radia Extensions for Windows Installer is comprised of many different components that you can use to create a Windows Installer package.

- **Radia Packager for Windows Installer**
  Radia Packager for Windows Installer is a suite of components that give you the tools to easily create, modify, customize, and manage Windows Installer Packages. The components installed with this product are listed below.
  - **Radia Packager for Windows Installer Icon, Menus, and Help**
    The Radia Packager for WI icon opens a quick-start menu with Typical and Custom packaging session options. The Typical option from this menu takes you to a streamlined packaging session, the Custom options take you to the appropriate State, Library or Package Wizard, based on your selection.

    An alternative, process-based packaging menu is also available. To access it, select **Radia Packager for WI Menu** from the Radia Packager for Windows Installer program group.
  - **Radia State Wizard**
    Create and modify state files. Accessed from the Radia Packager for Windows Installer Custom option **Identify resources to package.**
  - **Radia Library Wizard**
    Create and modify library files. Accessed from the Radia Packager for Windows Installer Custom option **Refine selected components**.
  - **Radia Package Wizard**
    Build and modify the Windows Installer packages. Accessed from the Radia Packager for Windows Installer Custom option **Package Components.**
  - **Radia Install Wizard**
    Manage your installation packages and transforms. Accessed from the Radia Packager for Windows Installer program group.
  - **Radia Insulation Wizard**
    Use application specific resources that are isolated from other applications. Accessed from the Radia Packager for Windows Installer program group.
  - **Radia Merge Module Wizard**
    Create, modify, view, and delete merge module components (in Windows Installer .MSM format) and Merge Module Groups. Accessed from the Radia Packager for Windows Installer program group.
  - **Radia MSI Editor – Radia MSIEdit**
    Allows you direct access to the Windows Installer database tables through an easy to use interface. Accessed from the Radia Packager for Windows Installer program group.
- **Radia Extensions for Windows Installer Client Components –** Radia Extensions for Windows Installer requires particular components be installed on the client computer if you are going to use certain features. The client components consist of:

- **Radia Advanced Features** (AFSetup.exe/AFSetup.ini) – The component enables Radia Machine/User support, as well as the use of the advanced permissions available when creating Windows Installer packages.

> **Note**
>
> Radia Machine/User support is not supported in Radia 2.x.

- **Radia Packager for Windows Installer Client** (APSetup.exe/APSetup.ini) – Enables application tracing and profiling when the Radia Packager for Windows Installer is being accessed remotely.
- **Radia Application Insight** (AISetup.exe/AISetup.ini) – In a lab environment, this component monitors installed applications and assists with the troubleshooting process.

About Packaging and Publishing with Radia

Packaging is the process of identifying resources, editing those resource's installation attributes, defining how they are to be installed, and saving the resources and installation instructions in a machine-readable file format. A package typically contains one or more files and configuration settings.

Publishing is the process of importing a package and its imbedded information to the Radia Database (AKA the Radia Configuration Server Database). A package must be published before its content can be distributed and deployed into your environment.

Radia offers many different packaging and publishing options through multiple tools. Make note of the different types of resource packages you will create and publish, then choose your Radia packaging and publishing products as required.

- If you're **packaging** Windows Installer files, use this guide.
- If you're **publishing** Windows Installer files, use the **Radia Publisher**. (See the *Radia Publisher Guide).*
- To **package** and **publish** all other types of files and applications, use the **Radia Packager**. (See the *Radia Application Manager Guide* or the *Radia Software Manager Guide*).

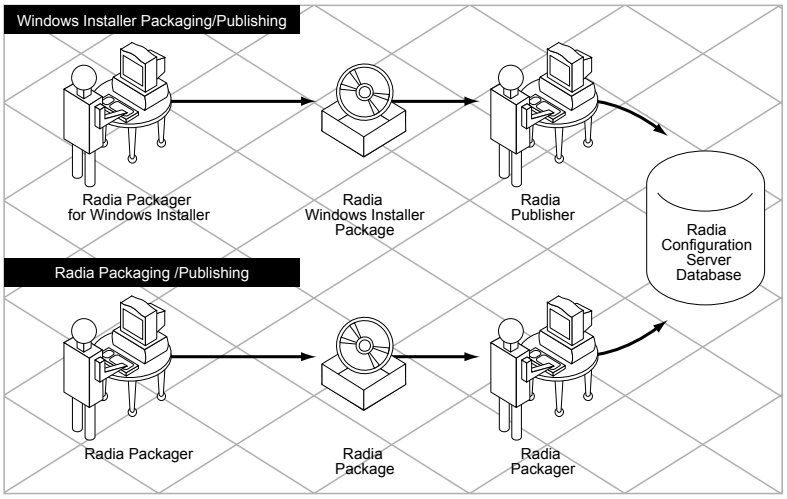

Figure 1.1 ~ Radia Packaging/Publishing options.

The Radia Extensions for Windows Installer Getting Started Guide

This guide contains specific information regarding the installation and use of each tool available with the Radia Extensions for Windows Installer.

Starting Radia Packager for Windows Installer

Start Radia Packager for Windows Installer from the **Radia Packager for WI** desktop icon or by selecting **Radia Packager for WI** from the Radia Packager for Windows Installer program group.

This opens the Radia Packager for Windows Installer quick-start menu shown in Figure 1.2.

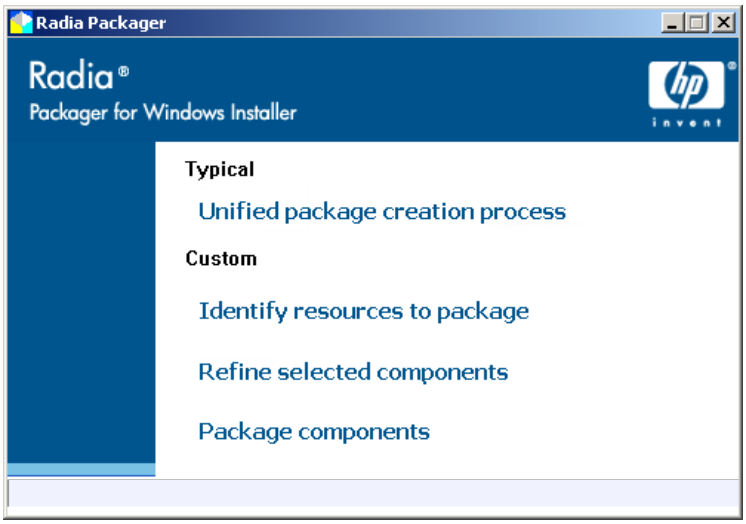

Figure 1.2 ~ Radia Packager for Windows Installer, Initial Menu.

The Radia Packager for Windows Installer provides Typical and Custom options for the package creation process.

About the Unified Package Creation Process

To build a package in a single, streamlined session, select the **Unified package creation process**. This mode supports both Installation Analysis and Resources Selection Methods.

- To use the Installation Analysis Method:

  Start by scanning the current system, install the application, and continue with a post-install after you install an application.

  OR

  Start with a saved system state (such as a baseline state) and compare it to the current state with the application currently installed.

- To use the Resource Selection Method:

  Start by selecting each resource to be included in the package.

Regardless of which method is used, the unified package creation process first produces a **State** file of the resources needed by the application, as shown in Figure 1.3. From this State file, the unified package creation process builds a **Library** file, and then uses the Library to create the **Windows Installer Package** for the application. The *system state* file contains resource pointers and detailed information about the resources; the *library* contains the actual binary files in addition to this information.

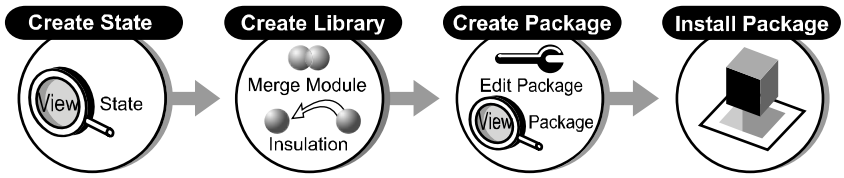

Figure 1.3 ~ Windows Installer package creation process.

After setting up the appropriate filters and your template MSI (both of which are detailed in later chapters in this *Radia Extensions for Windows Installer Guide*), your packaging process may not differ much from the steps outlined in this chapter when creating a new package.

About the Custom Session Options

Use the **Custom** session options of the Radia Packager for Windows Installer to create or modify state files, library files, or package files. The custom options allow access to the settings and advanced features of Radia Packager for Windows Installer discussed in the remaining chapters of this guide.

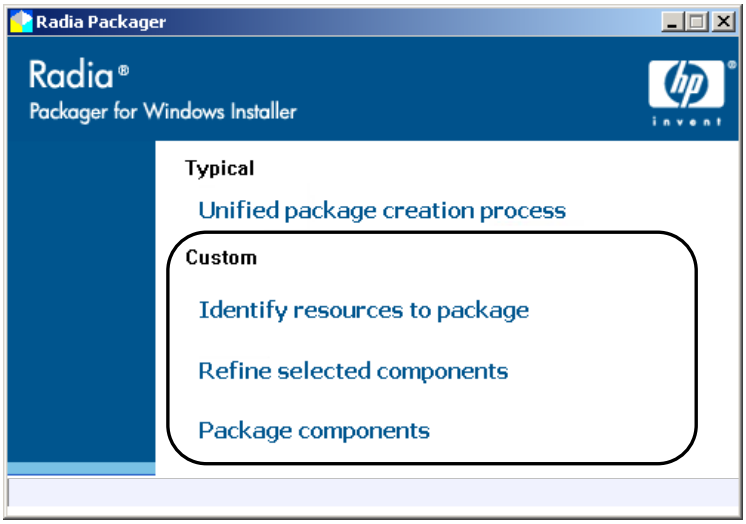

Figure 1.4 ~ Radia Packager for Windows Installer, Custom options.

Each Custom option is described below:

- Select **Identify resources to package** to access the Radia State Wizard, which is used to create, combine, or manage state files. For example, use the Radia State Wizard to create a baseline machine state or change a state file description. See *Using Radia State Wizard Advanced Functions* on page 100 for more information.
- Select **Refine selected components** to access the Radia Library Wizard, which is used to create or modify library files. Many important packaging and management application functions are performed within libraries. For example, use the Radia Library Wizard to modify the target path of an installation, make file or variable substitutions, apply permissions, or compress or decompress the library files. See *Using Radia Library Wizard Advanced Functions* on page 117 for more information.
- Select **Package components** to access the Radia Package Wizard, which is used to create, modify, or upgrade packages, as well as apply Advanced Options available for Windows Installer packages.
  - See the chapter *Upgrading a Package* on page 77 for information on how to to package a new release or package a new build from a patch package.
  - See the topic *Using Radia Package Wizard Advanced Functions* on page 167 for information on rebuilding a package, upgrading multiple products, editing a package template, or setting any of the Advanced Options available for Windows Installer packages.

About the Radia Packager for Windows Installer Process Menu

Radia Packager for Windows Installer includes an alternate process-based menu, shown in Figure 1.5 below. Use the icons and buttons on this process menu to access any components of the Radia Packager for Windows Installer, as well as the Radia Configuration Analyzer, if installed.

To access this menu, select **Radia Packager for WI Menu** from the Radia Packager for Windows Installer program group.

To obtain more information on how to use this menu, click the **Help** button to access online help.

Figure 1.5 ~ Radia Packager for Windows Installer Menu with Packaging Process Flow.

To access this menu, select **Radia Packager for WI Menu** from the Radia Packager for Windows Installer program group.

Publishing a Package to Radia

After building and testing a package, use the Radia Publisher to promote your Windows Installer packages into the Radia Database. The Radia Publisher is one of the tools installed with the Radia Administrator Workstation. See the *Radia Publisher Guide* for more information.

Summary

- The Radia Extensions for Windows Installer is comprised of many components.
- Most components are part of the Radia Packager for Windows Installer.
- Additional Client Components are installed separately.
- Typical and Custom packaging sessions can be selected from the initial Radia Packager for Windows Installer menu.
- An alternate, processed based menu is also available for quick access to individual components of the Radia Packager for Windows Installer and the Radia Configuration Analyzer, if installed.

2

Creating a Package

At the end of this chapter, you will:

- Create a Windows Installer Package using the unified package creation process. This Typical option creates a package in a single, streamlined packaging session.
- Be able to use the Radia Packager to:
  - Apply filters to a state file or package.
  - Modify Library Installation Settings.
  - Apply Advanced Options to the Package.
- Know about the features available through the Custom options of the Radia Packager menu.

Creating a Package Using the Typical Session Option

The best way to become acquainted with the Radia Extensions for Windows Installer is to build your own Windows Installer package. This topic takes you through the steps using the Radia Packager for Windows Installer in a streamlined session.

Use the **Typical—Unified package creation process** of the Radia Packager for Windows Installer to create a package in a single, unified packaging session. This option requires minimal interaction.

Note

To complete the example below, an application must be installed after the initial scan by the Radia Packager for Windows Installer is complete. For this example, we will install the **ORCA** tool, available from Microsoft, at www.microsoft.com**.** ORCA is a database table editor for creating and editing Windows Installer packages and merge modules.

To create a package using the Typical Session option

1. Double-click the **Radia Packager for WI shortcut** on your desktop

   OR

   Go to **Start**, Programs, **Radia Packager for Windows Installer**, and select **Radia Packager for WI**.

Figure 2.1 ~ Radia Packager for Windows Installer, desktop shortcut.

The **Radia Packager for Windows Installer** menu opens.

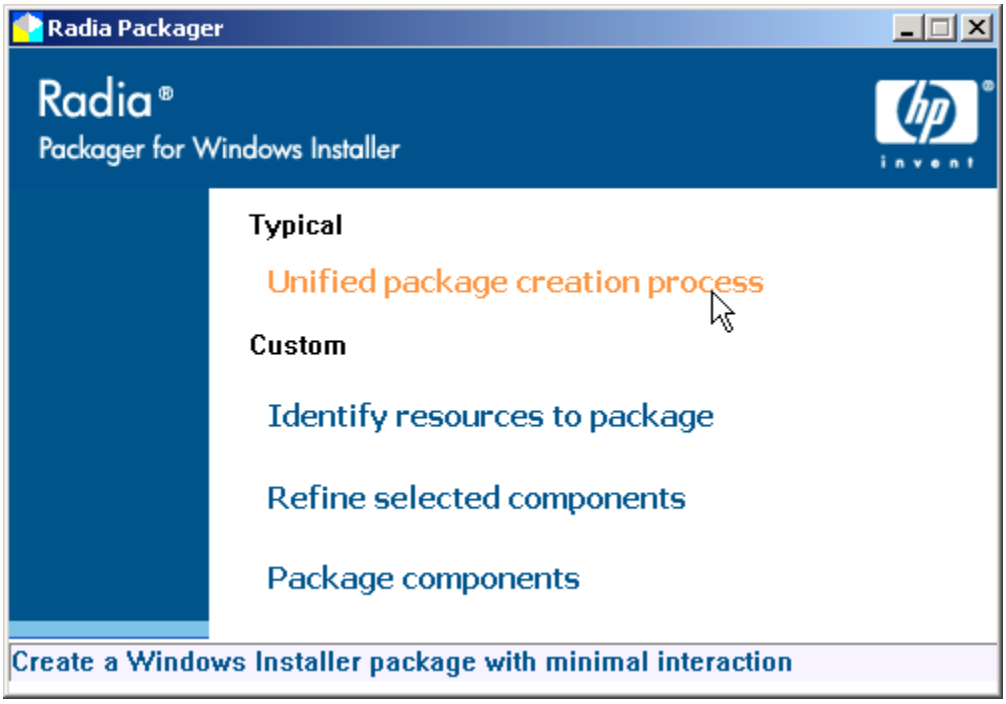

Figure 2.2 ~ Radia Packager for Windows Installer, start menu.

2. Select **Unified package creation process** under the Typical option.
   The **Select the Package Creation Method** dialog box opens.

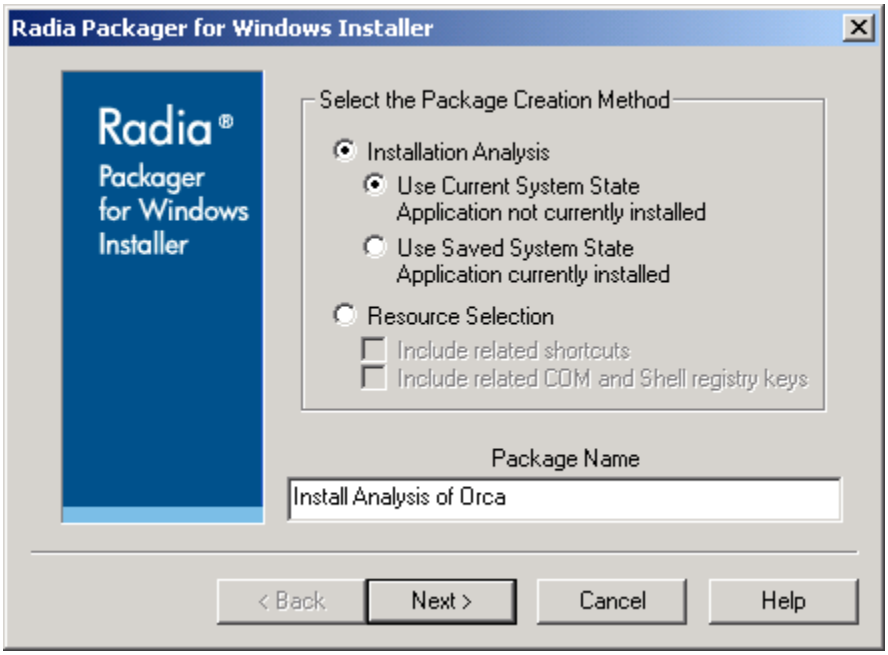

Figure 2.3 ~ Radia Packager for Windows Installer – Select the Package Creation Method.

3. Select **Installation Analysis** and **Use Current System State** to begin creating a package with a scan of the machine's resources before you install a program. For this exercise, we are going to install the ORCA tool, available at www.microsoft.com.

About the Package Creation Methods

When you select an **Installation Analysis using Current System State Files**, the Radia Packager for WI performs the same functions as the Radia State Wizard to create a before installation state file of the machine. The before state file represents the state of the machine before installing the application. It will be compared to the state of the machine after you install an application to create the package for the application.

Select **Installation Analysis, Use Saved System State** to have the Radia Packager for WI compare current state of the machine with the application installed to a previously saved baseline state of the machine. The analysis of the differences between the current and baseline states will be used to create the application package.

Select **Resource Selection** to create a package from a set of known resources.

4. Enter a **Package Name**, such as **Install Analysis of ORCA**.

Note

This package name will also be applied to the state and library files created during the packaging process.

5. Click **Next**.

   The **Select the Scope of Analysis** dialog box opens.

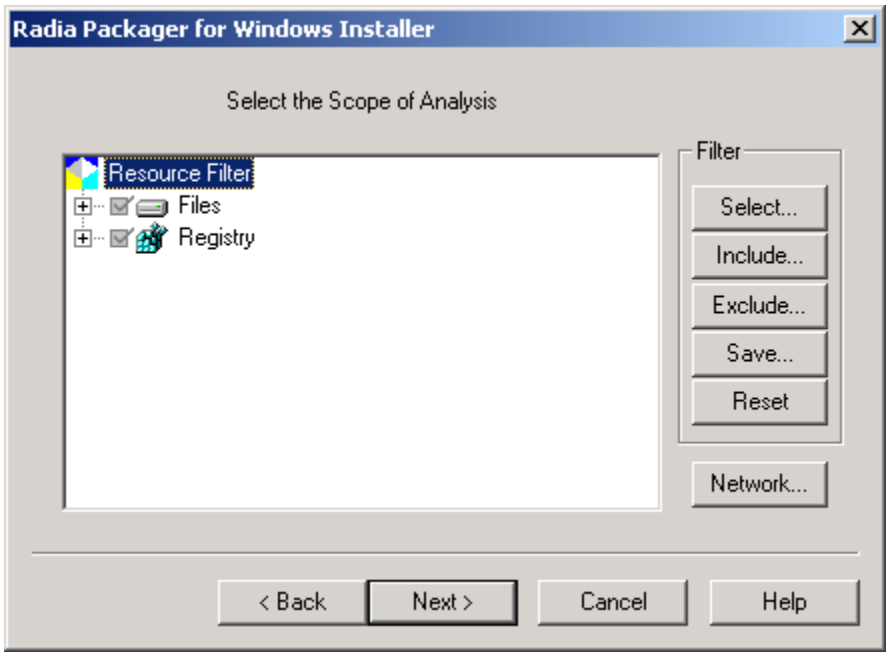

Figure 2.4 ~ Radia Packager for Windows Installer – Select the Scope of Analysis.

6. Use the Resource Filter tree to select the files and registry information to include in your scan. For the purpose of this exercise, select the C drive only. Clear the check boxes for all other selected drives, as shown in Figure 2.5.

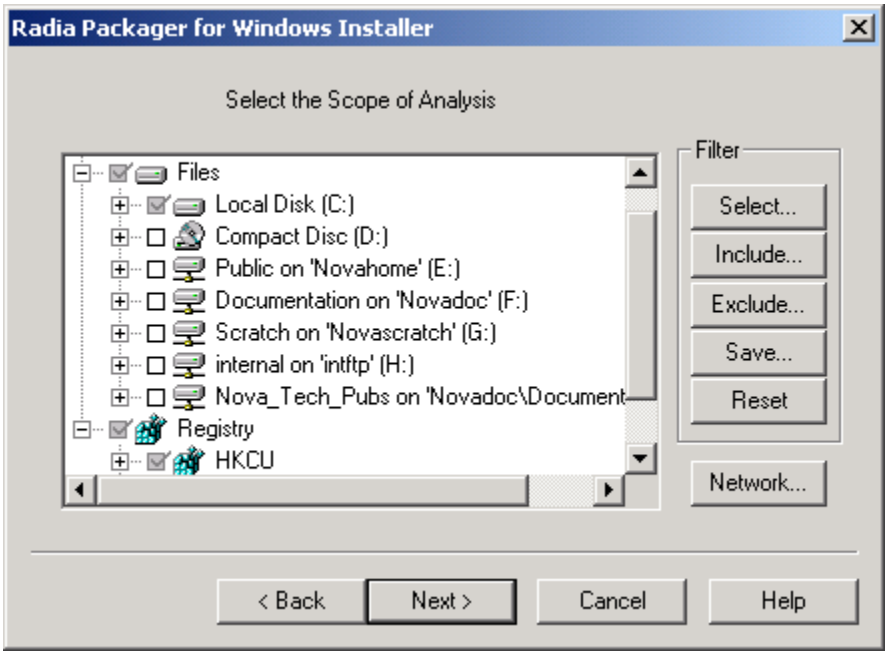

Figure 2.5 ~ Filtering Files for Radia Packager for Windows Installer – Select the Scope of Analysis.

Note

Do not *over-filter* the initial scan. You may not realize that an application places files in various directories, and if you filter that directory out, the changes will not be reflected in the final package.

If necessary, additional filtering can be applied after you run a Typical packaging session. Use the Custom menu option: **Refine selected components** to apply additional filtering to the captured State files, create a new Library, and then "repackage" the Library. For more information, see the *Filtering* chapter on page 203.

Tips

- Press **F10** on your keyboard to maximize the dialog box for easier filtering of paths, files, and registry entries. To return to the initial size, press **F10** again, or close the dialog box.
- Click the **Save** button to save this filter selection. You can select and apply any saved filters in later packaging sessions.

7. Click **Next**.

The **Installation Analysis in Progress** dialog box opens and performs the initial scan of the machine.

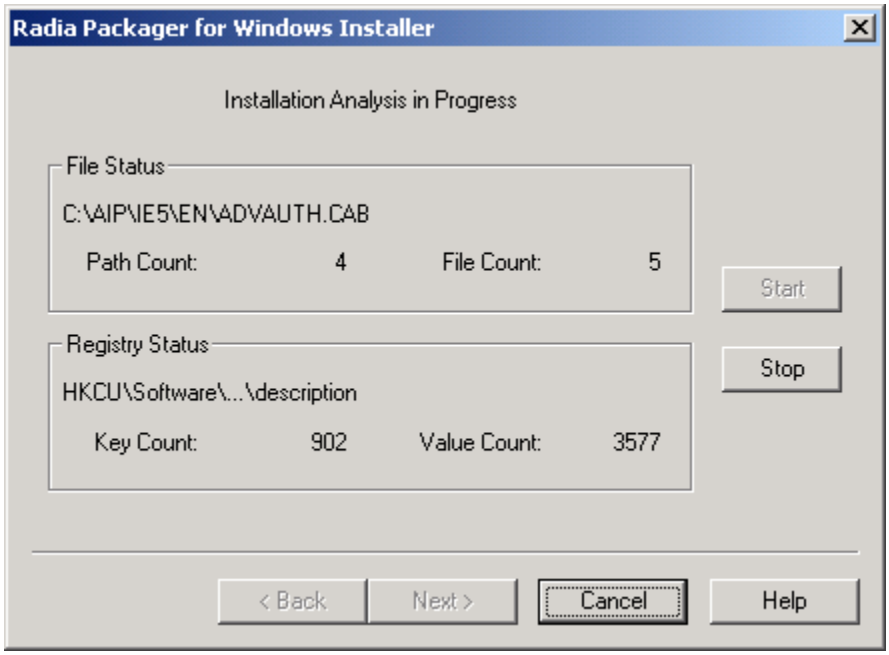

Figure 2.6 ~ Radia Packager for Windows Installer – Installation Analysis in Progress.

Note

Click **Stop** to halt the scan. This allows you to use **Back** and revise the scan filter, or use **Start** to restart the scan.

Click **Cancel** only if you want to end your current Radia Packager session altogether.

8. When the scan is finished, the **Current System State Saved** dialog box opens. This dialog box indicates it's time to install the application.

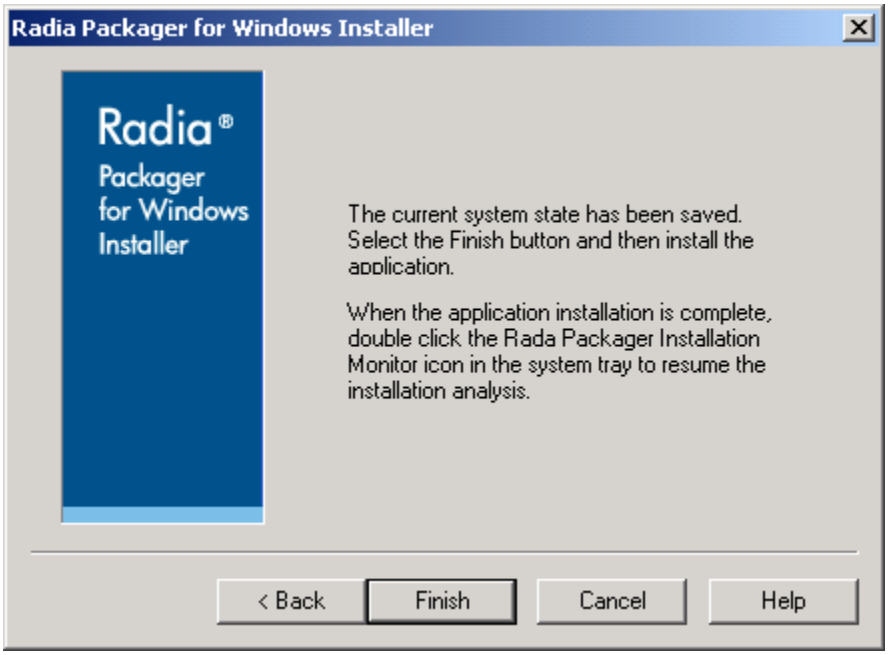

Figure 2.7 ~ Radia Packager for Windows Installer – Current System State Saved, Time to Install Application.

Important Note

Remember to install a sample application at this point.

9. Click **Finish**.

While the Radia Packager waits for you to install the application, a Radia Packager for Windows Installer icon is added to the Windows System Tray.

Figure 2.8 ~ Icon in Windows System Tray to Restart Radia Packager Installation Monitor.

10. Install the application to be packaged. For this exercise, we are using ORCA, available at www.microsoft.com.

About the After and Delta State Files

After you install the application, the Radia Packager runs a post-installation scan to create **after** and **delta** state files. This phase of the installation analysis obtains a full and accurate view of the resources required and used by the application.

11. After you install the application, double-click the  icon in the system tray to restart the packaging session.

A Radia Packager dialog box asks if the installation finished successfully.

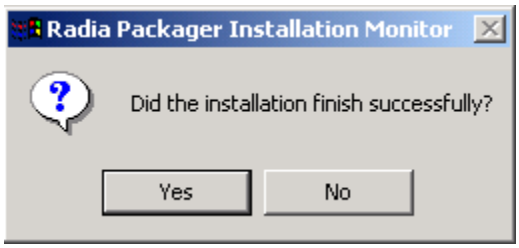

Figure 2.9 ~ Radia Packager prompt after double-clicking the system tray icon.

12. If the installation finished successfully, click **Yes** to confirm and begin the post-install analysis of the system state files.

Note

If the installation did not finish successfully, click **No** when prompted. Selecting No restores the Radia Packager for WI icon to the system tray so you can finish the installation.

The Radia Packager continues with a post-installation scan of the machine. It will also perform a *Combination Analysis* of the scans captured before and after the application was installed.

Figure 2.10 ~ Radia Packager for Windows Installer – (After) Installation Analysis in Progress, below, shows File Status of "Combination Analysis Complete", but the analysis of the registry keys is still in progress.

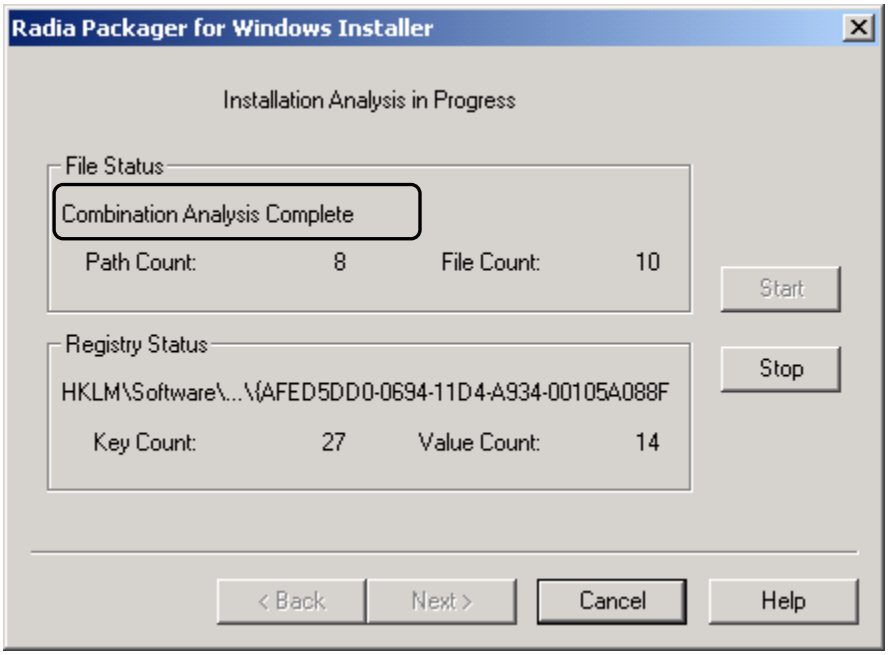

Figure 2.10 ~ Radia Packager for Windows Installer – (After) Installation Analysis in Progress.

When the scans and analyses are complete, the Radia Packager for Windows Installer **Select the Paths, Files and Registry Entries to Include** dialog box opens.

The Radia Packager is ready to create a library from the state file. The system state file contains resource pointers and detailed information about the resources; the library contains the actual binary files in addition to this information.

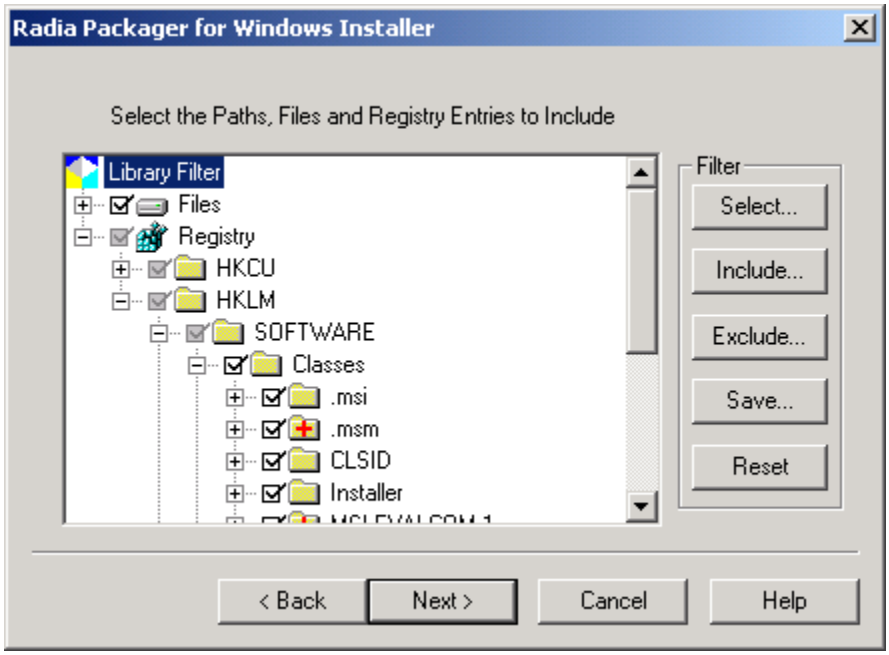

Figure 2.11 ~ Radia Packager for Windows Installer – Select the Library Filter Paths, Files and Registry Entries to Include.

13. Use this dialog box to filter items from the library before creating the package. To help determine what items to filter, the directories, files, and registry keys are denoted as added, changed, or removed.

> **Note**
>
> If necessary, additional filtering can be applied after you run a Typical packaging session. Use the Custom menu options: **Refine selected components** and **Package Components** to apply additional filtering to the Library resources. For more information, see the *Filtering* chapter on page 203.

14. Click **Next**.

The Radia Packager performs a quick analysis of the files to include in the library. The **Modify Library Installation Settings** dialog box opens.

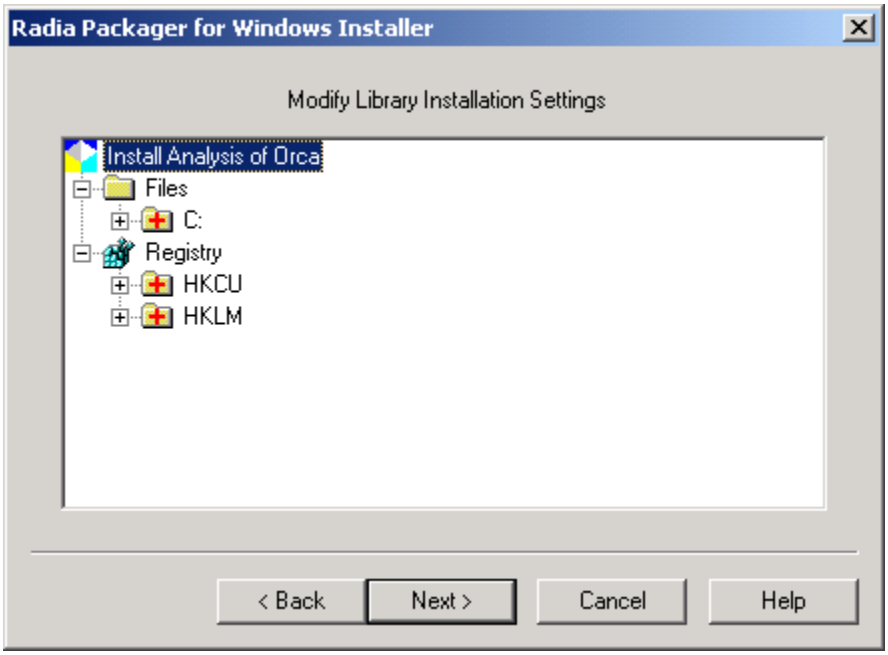

Figure 2.12 ~ Radia Packager for Windows Installer – Modify Library Installation Settings.

When packaging your own applications, use this dialog box to modify the Library Installation Settings. This is a very important and powerful dialog box. The information set here determines properties for individual files, directories, and registry settings upon installation from a Windows Installer package created with the Radia Packager or Radia Extensions for Windows Installer. You can also add and remove resources from the library here.

To obtain the menu of library modifications available, select a resource, then click right. Figure 2.13 shows a sample File resource menu.

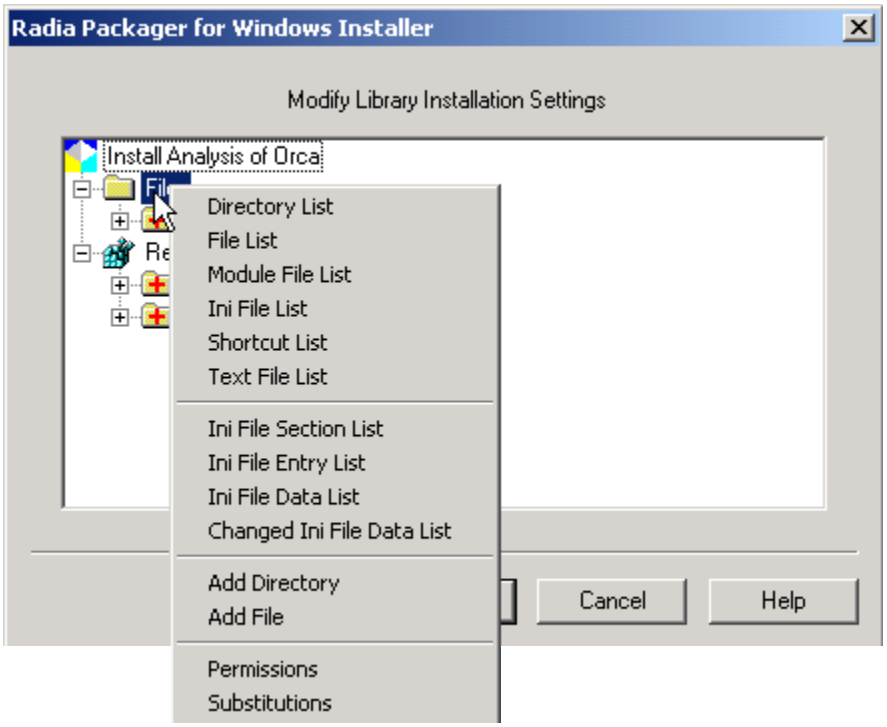

Figure 2.13 ~ Radia Packager for Windows Installer – Sample Modify Library Installation Settings menu for Files.

Click the online **Help** button for more information.

OR

See *Using Radia Library Wizard Advanced Functions* on page 117 for more information.

15. Click **Next**.

The **Library File Copy in Progress** dialog box opens while all files and registry settings are copied into the library.

After the Library file is saved, the Radia Packager is ready to package the library. The next dialog asks you to **Select the Packaging Option**.

About Radia Packages for MSI Applications

The Radia Packager for Windows Installer creates a Windows Installer Package (MSI) that contains the information and files necessary to install an application. The MSI package maintains installation information in a relational database. Actual binary files are stored compressed within the package itself or in an external .cab file.

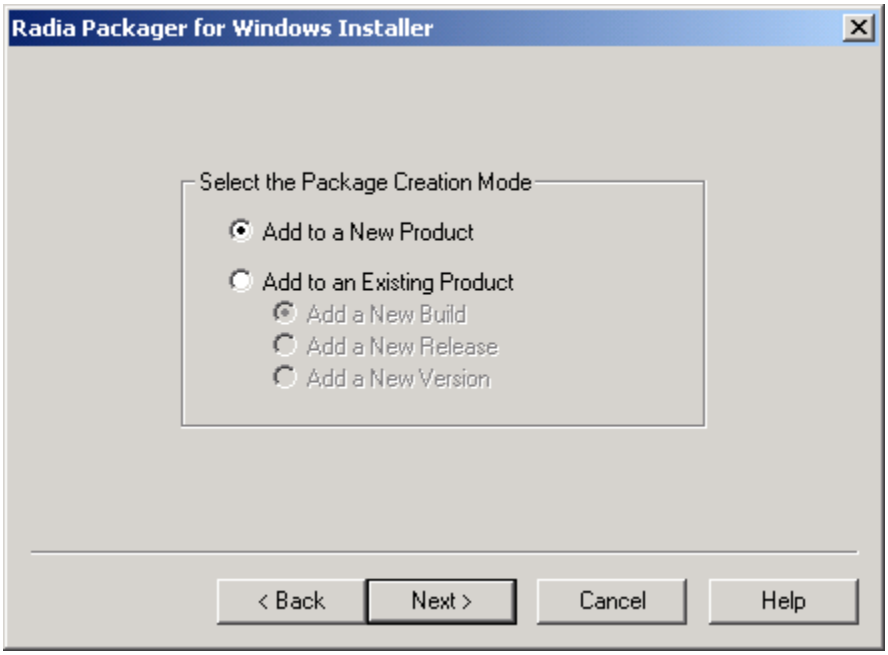

Figure 2.14 ~ Radia Packager for Windows Installer – Select the Packaging Option.

16. Select **Add to a New Product,** and then click **Next**.

The **Enter the Product Name, Version, Release and Build Numbers** dialog box opens. Use the information in this dialog box to track the name and version of Acrobat Reader that this package installs, while also removing older versions of the application when installing new releases.

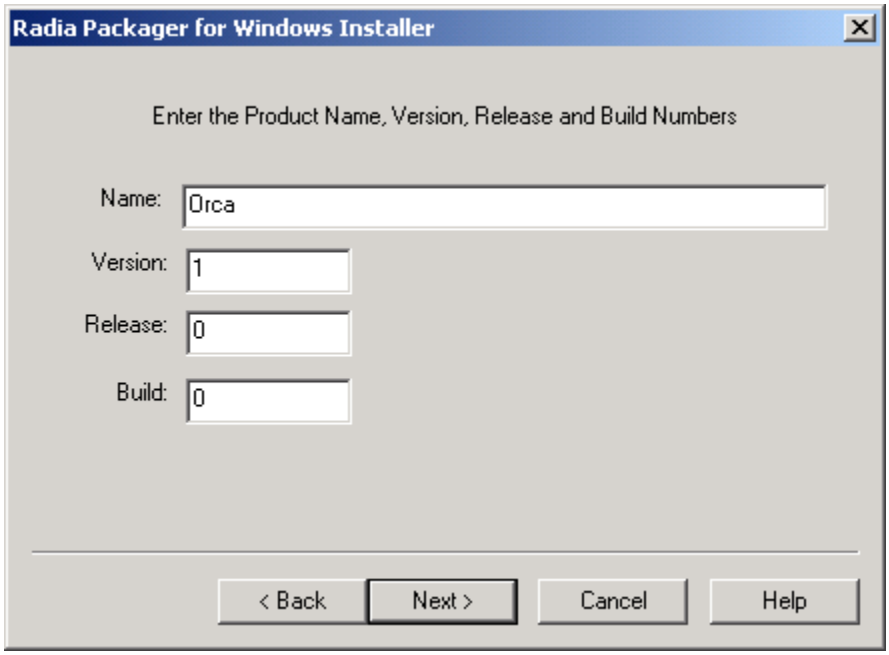

Figure 2.15 ~ Radia Packager for Windows Installer – Select the Product Name, Version, Release and Build Numbers.

17. In the **Name** text box, type a unique name for the product, such as **ORCA**.

Note

Use the appropriate naming and versioning conventions for your company. HP Best Practices for Packaging suggests excluding any version information from the product name.

18. Type the version in the **Version** text box.

19. Type the release number in the **Release** text box.

20. Type the build number in the **Build** text box.

Note

You can easily track the version of third party applications by filling in their version information as the release and build numbers of your repackaged application. For example, Acrobat 6.0 becomes 1.6.0. When you create the upgrade Package to Acrobat Reader 7.0, the new Package will be 1.7.0.

21. Click **Next**.

22. The **Windows Installer Options** dialog box opens. When creating your own packages, use this dialog box to modify the many information and installation settings embedded in your Windows Installer package. The tabs on this dialog allow you to access the same Windows Installer Options that are available from the Radia Packager Wizard Advanced Options dialog. See *Appendix B: Radia Package Wizard Advanced Options* starting on page 247 for more information.

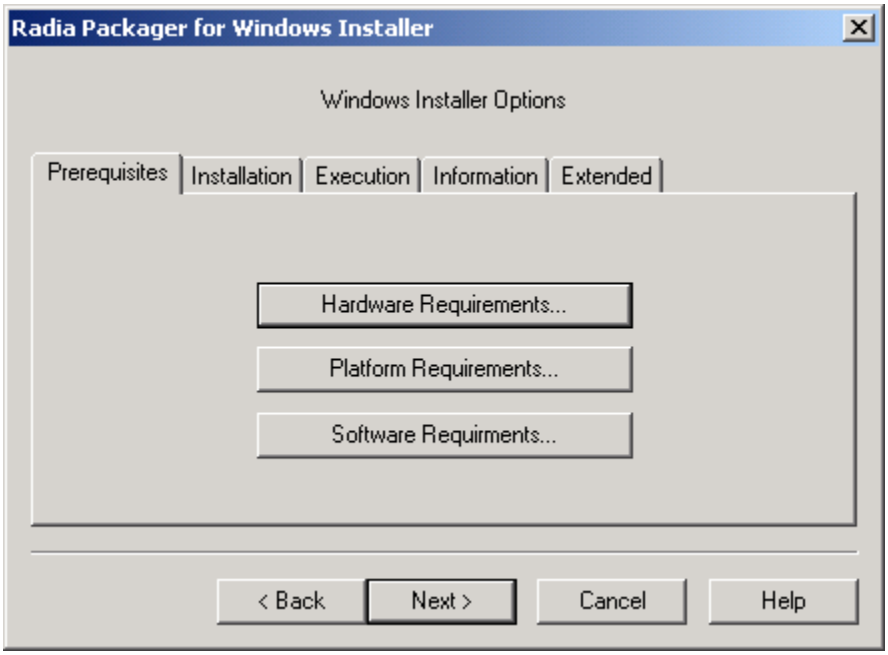

Figure 2.16 ~ Radia Packager for Windows Installer – Windows Installer Options.

23. Click **Next**.

The Radia Packager performs a quick analysis of the files to include in the package.

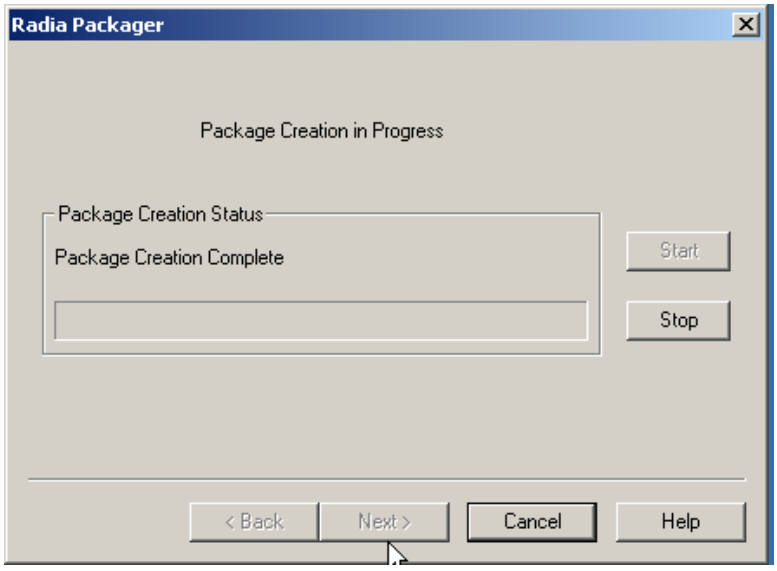

Figure 2.17 ~ Radia Packager for Windows Installer – Package Creation in Progress.

Note

If there are any issues with the package, the **Package Creation Exceptions** dialog box opens. Review any exceptions and click **Next** to continue

The **Package creation is complete** dialog box opens.

Figure 2.18 ~ Radia Packager for Windows Installer – Package creation is complete.

24. Click **Finish** to complete the packaging session.

Viewing the Product

Now that you have created your first package, you should understand the files that are created when you build your package, and how these files are organized. The package is located in a subdirectory of the **product** directory defined in the default settings.

To see how products, packages, and files are organized

1. Open **Windows Explorer**.
2. Navigate to the **Product Path** defined in the Radia Extensions for Windows Installer Settings, Paths(1) tab. By default, this location is *SystemDrive*:\Program Files\Novadigm\AdvPub\Product.
3. In the Product directory, open the directory with the name of your product. In this directory, you will find a folder based on the operating system (Win9x or WinNT, both if you have created a package for each operating system) and a state file, Product.ISState. This state file contains general information for creating update packages for any operating system.

Note

Products are containers for the different versions of a package. In this case, we created a product called ORCA. Within the ORCA Product, we will find the packages we've created for different versions of ORCA on the Windows NT (and 2000), Windows XP, and Windows 9x platforms.

4. Open the **WinNT** or **Win9x** folder.
5. Open the folder named with your versioning information, such as **1.0.0**. This folder contains another state file used to update packages for the specific operating system. This folder will contain multiple versions of an application as you upgrade to new releases and apply bug fixes to your packages.

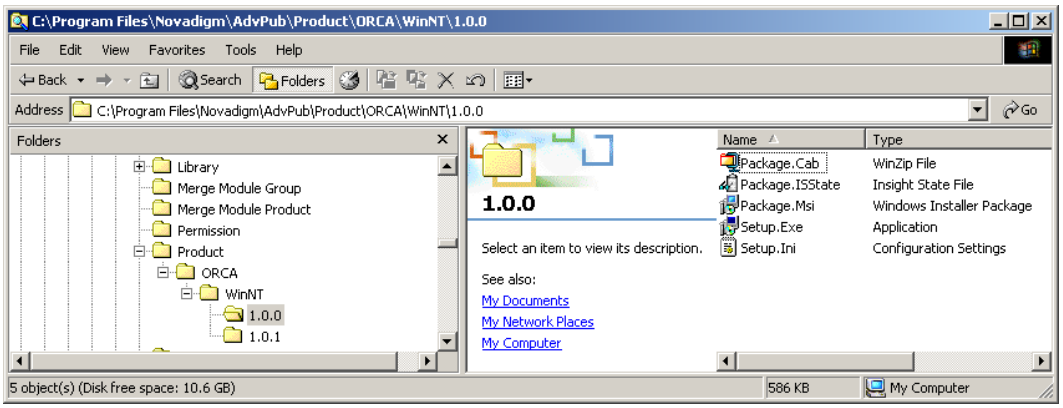

Figure 2.19 ~ Product folder.

The folder named with version information contains five files:

- **Package.Cab**
  The binary files installed by the Windows Installer package in compressed form.
- **Package.ISState**
  A state file that contains detailed information on the files that are installed with your package. This file can be imported into the Radia Application Knowledge Base for historical reference and comparison with other applications.
- **Package.Msi**
  The Windows Installer database.
- **Setup.Exe**
  An executable file that calls MSIEXEC to install the package for you.

- **Setup.Ini**
  The information and parameters used by Setup.Exe to install the package. You can modify this file to perform different tasks.

To view the MSI package after the Radia Packager Session

To view the packaged application, you can run the **Radia MSIEdit** program available from the Radia Extensions for Windows Installer program group.

1. Go to **Start**, **Programs, Radia Packager for Widows Installer**, and select **MSIedit.**

   The MSIedit window opens.

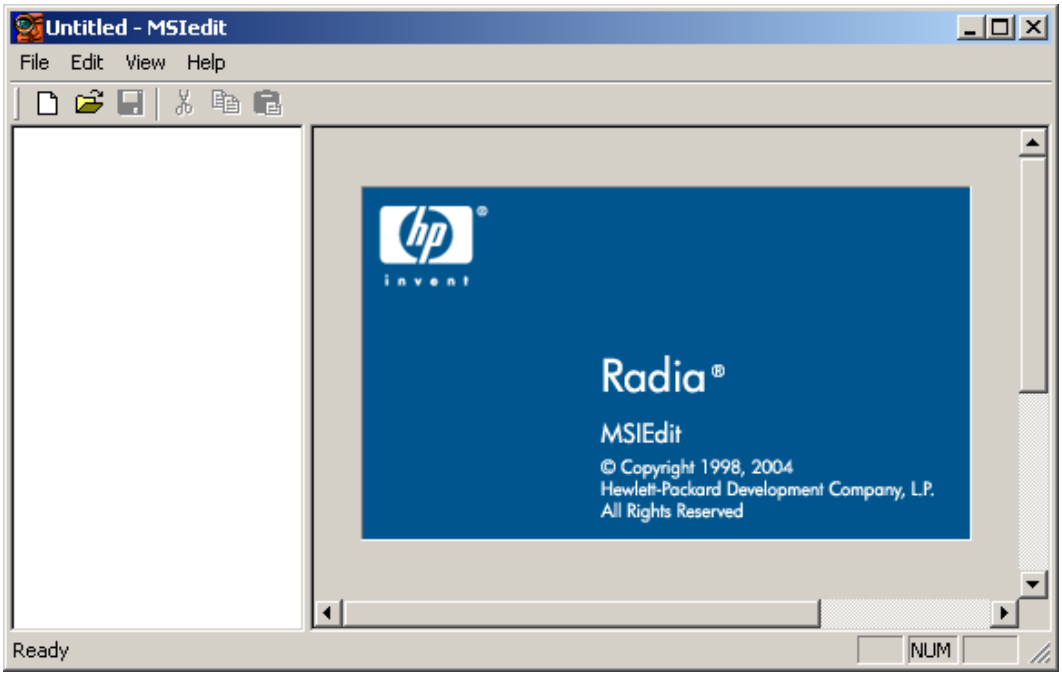

Figure 2.20 ~ Initial MSIedit window.

2. Use the **File Open** menu command to open the Package.msi file for the package just created in your Typical packaging session.

   The *.msi package will be located in the **Product\<package name>\<platform>\<version>** folder of where Radia Extensions for Windows Installer was installed. By default, this is:

   ```
   C:\Program Files\Novadigm\AdvPub\Product\ <Package Name>
   \<Platform>\<Version>
   ```

For example, the ORCA.msi package created in the previous session will be found in the \Product\ORCA\WinNT\1.0.0 folder of the installation directory of Radia Extensions for Windows Installer.

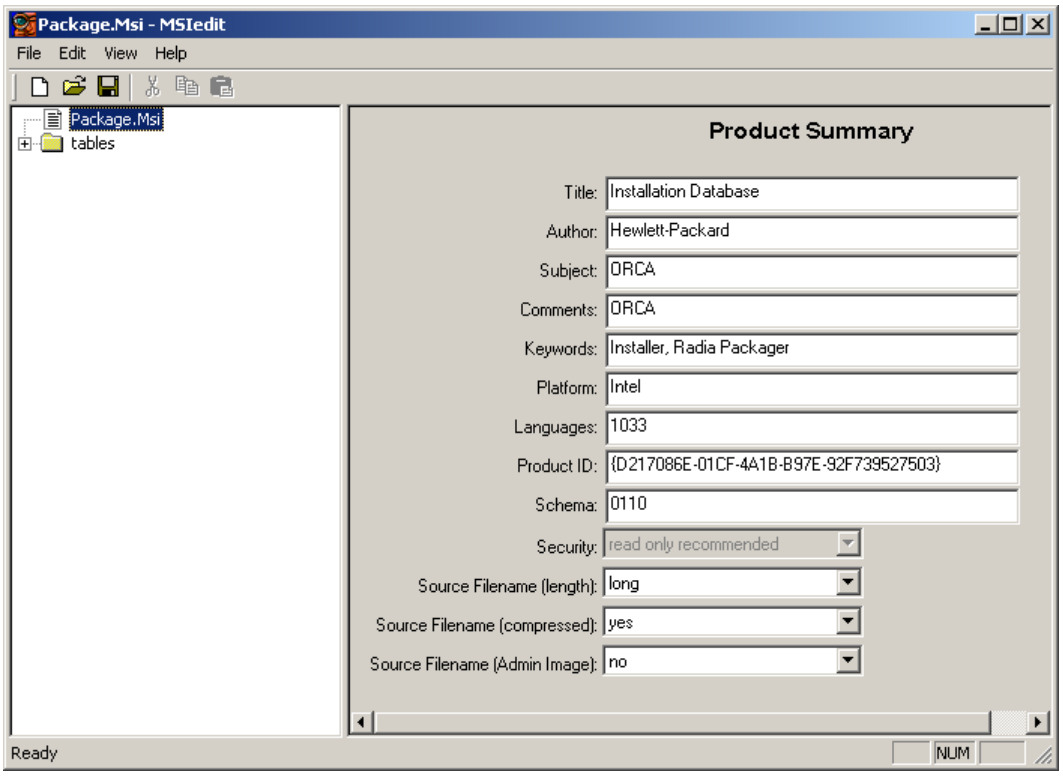

Figure 2.21 ~ Radia MSIEdit window.

3. Use **MSIedit** to view or modify the MSI file configurations and settings for the package you just created.

> **Note**
>
> HP recommends using the Custom options of the Radia Packager menu to make any modifications to your package. See the later chapters in this guide for more information.

4. Exit the MSIedit session.

Summary

- The Radia Packager for Windows Installer streamlines the process of creating a Windows Installer package.
- The **Unified package creation process** creates a Windows Installer Package in a single, streamlined session.
- Select **Installation Analysis** to create a package from the resources identified by comparing scans before and after the installation of an application.
- After the initial scan, the Radia Packager for Windows Installer prompts you to run the install for the application to be packaged, places an icon in the system tray, and then resumes execution to complete the Post-Installation Analysis.
- Following the before and after installation scans, the Radia Packager for Windows Installer creates a library from the installation analysis state file. You can modify the library settings to adjust resource properties and the library contents before the package is built.
- After modifying the library settings, the Radia Packager for Windows Installer creates a Microsoft Windows Installer package from the library. Use the Advanced Options dialog box of the Radia Package Wizard to modify the common settings and information placed into your Windows Installer Packages.
- Finally, you have the options to validate and view the finished package after the package is built.

3

Managing an Installation Package and Transforms

At the end of this chapter, you will:

- Be able to manage the packages created by the Radia Packager for Windows Installer.
- Be able to copy packages to alternate distribution points.
- Be able to manage a package's Administrative Install Point (AIP).
- Be able to add Radia Package Management to an existing package, with or without Machine/User support.
- Be able to create and manage a transform file to customize the installation of a package.

Radia Install Wizard

For tasks that extend beyond the realm of basic packaging, use the Radia Install Wizard. When you have a package that needs to be modified, the Radia Install Wizard provides the tools to manage the package, and create and modify transform (MST) files.

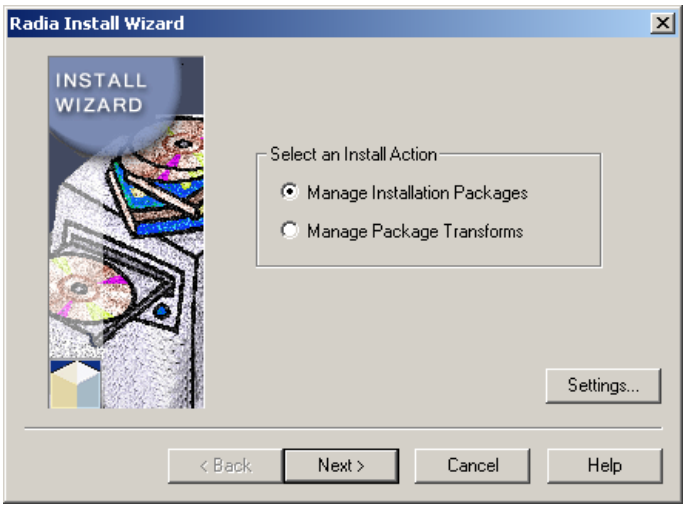

Figure 3.1 ~ Radia Install Wizard – Select an Install Action.

- **Manage Installation Packages**
  Manage packages for installation.
- **Manage Package Transforms**
  Manage package transform files.

Managing Installation Packages

Use the **Manage Installation Packages** features of the Radia Install Wizard to promote a package to an alternative distribution point for deployment, or to copy the package's state file to a location so you can import it into the Radia Configuration Analyzer.

When you are using the Radia Install Wizard to promote a package, you have the option of adding Radia Package Management to the package, if it does not already exist. Radia Package Management can be added with or without Machine/User Support, either directly to the package, or to a separate transform file.

The Manage Installation Packages features of the Radia Install Wizard also lets you manage the Administrative Install Point (AIP) of a package.

Publishing a Packaged Application to a Distribution Point

After you have packaged an application, you may want copy or store it at an alternative distribution point. Use these procedures if you are not using Radia deployment to distribute the package.

Also use this procedure to add Radia Management to an existing package, either with or without Machine/User Support. Radia Management can be added to the package itself, or as a separate transform.

To publish a packaged application to a distribution point

1. From the **Radia Packager for Windows Installer** program group, start the **Radia Install Wizard**.

   The **Select an Install Action** dialog box opens.

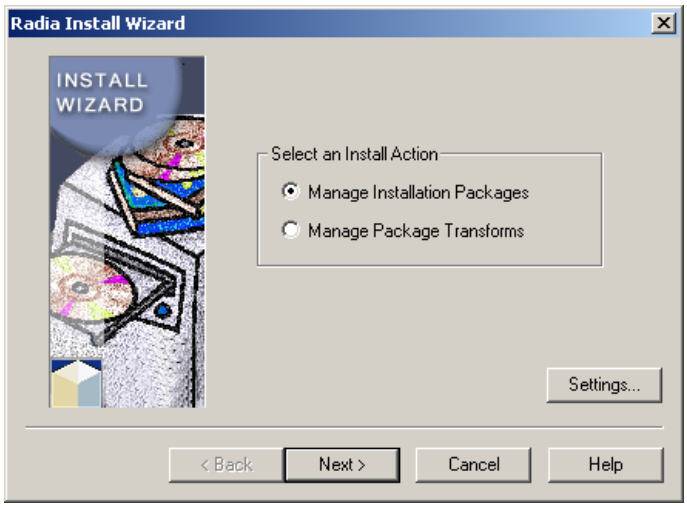

Figure 3.2 ~ Radia Install Wizard – Select an Install Action.

2. Select **Manage Installation Packages**.
3. Click **Next**.

   The **Select a Package Management Action** dialog box opens.

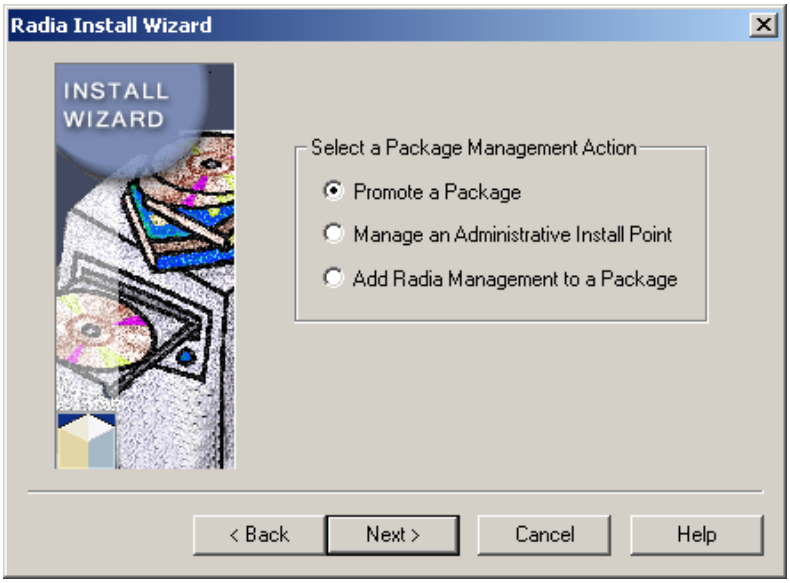

Figure 3.3 ~ Radia Install Wizard – Select a Package Management Action.

4. Select **Promote a Package**.
5. Click **Next**.

   The **Windows Installer Package file name** dialog box opens.

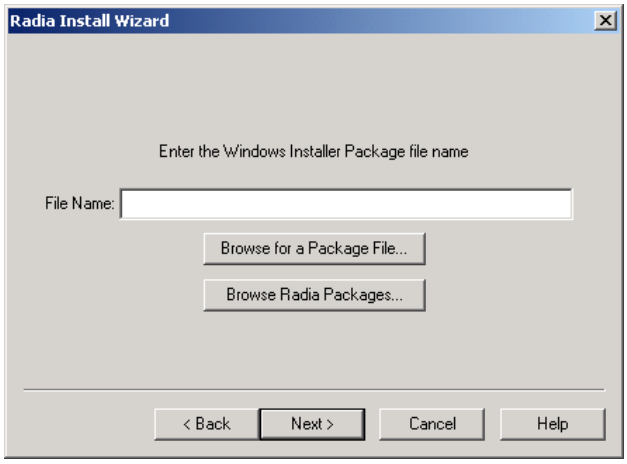

Figure 3.4 ~ Radia Install Wizard – Windows Installer package file name.

6. Type the name of the Windows Installer package file name, or click one of the **Browse** buttons to manually locate the file.
   - **Browse for a Package File** will locate any MSI files on the local computer or network.
   - **Browse Radia Packages** will open the Product directory as defined in the Radia Packager for Windows Installer settings.
7. Click **Next**.

   The **Select the Package Promotion Options** dialog box opens.
8. Select a package promotion option, and click **Next**.
   - Select **Copy Package Files to a Production Directory** if you are deploying the package to a non-Radia distribution point.

   Note

   Use the Radia Publisher to promote Windows Instalelr Cu'CSSwpackages to the Radia Database for deployment by Radia. See the *Radia Publisher Guide* for more information.

   OR
   - Select **Copy Package State to an Import Directory** to make the package available for import by the Radia Configuration Analyzer and the Radia Knowledge Base Manager.
9. Click **Next**.

   The **Select the Package File Copy Method** dialog box opens.

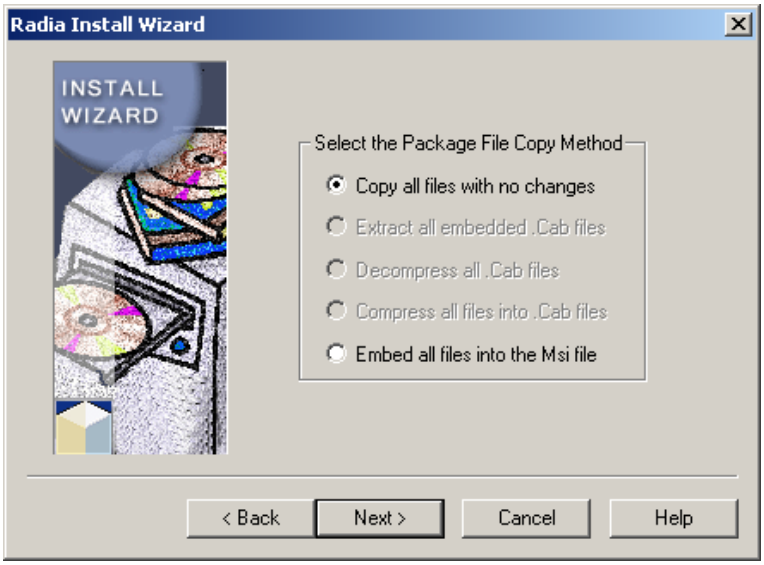

Figure 3.5 ~ Select the Package File Copy Method dialog.

10. Select **Copy all files with no changes** and click **Next**.

The **Package File Search** begins.

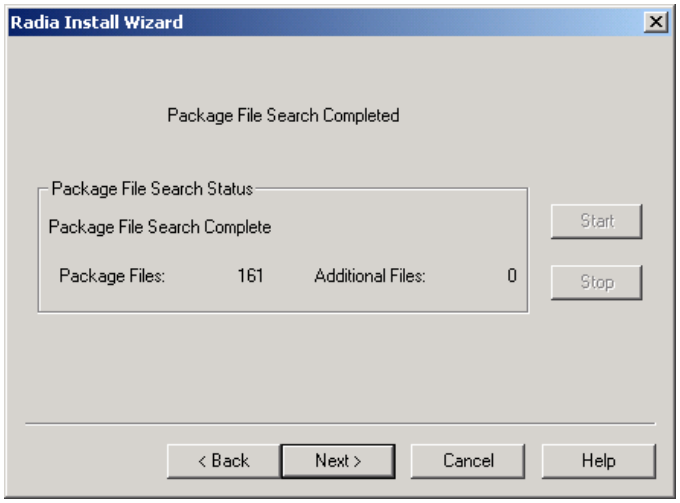

Figure 3.6 ~ Radia Install Wizard – Package File Search.

11. When the **Package File Search** is finished, click **Next**.

The **Select Additional Files** dialog box opens.

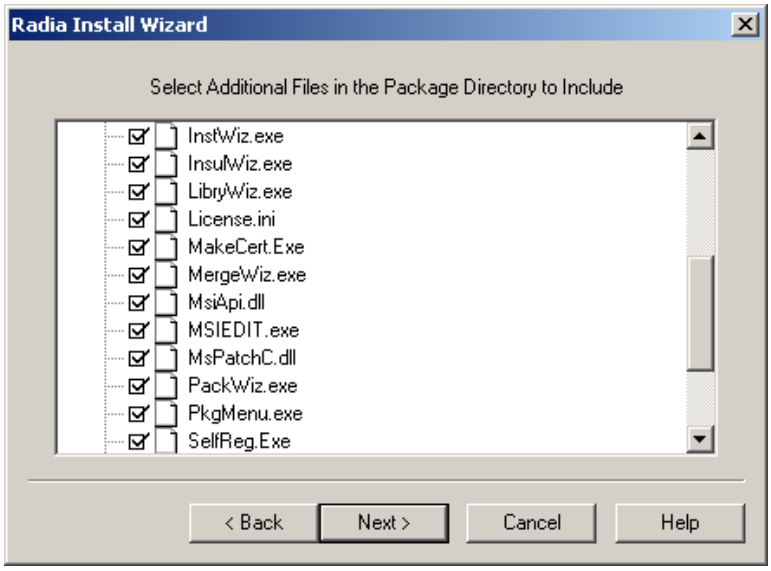

Figure 3.7 ~ Select Additional Files in the Package Directory to Include.

12. Select any additional files to include and click **Next**.

The **Radia Package Management** dialog box opens.

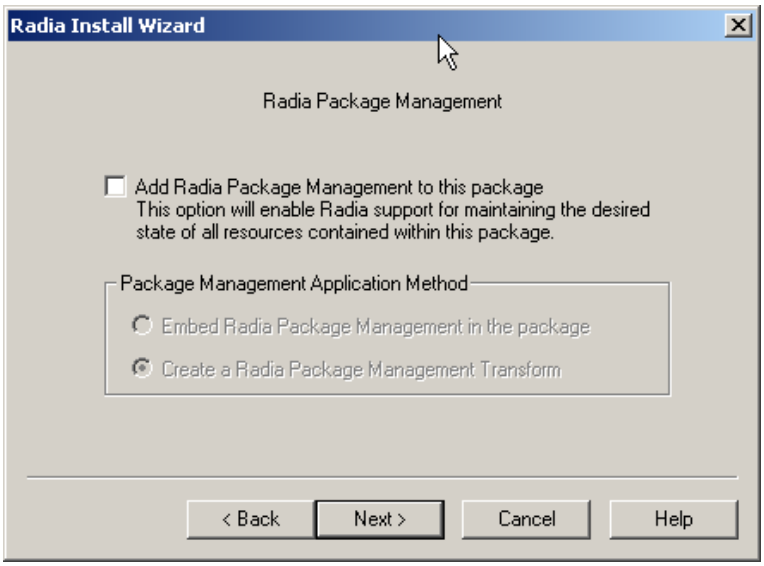

Figure 3.8~ Radia Package Management.

13. Select the check box if you'd like to add Radia Package Management to the package. If you select to add Radia Management, you must then select whether you'd like to embed Radia Package Management in the package or if you'd like to create a Radia Package Management Transform.

14. Click **Next**.

If you selected Radia Management, the **Radia Machine/User Support** dialog box opens. If you did not select Radia Management, the **Enter the Package promotion path** dialog box opens. Skip ahead to the section *Enter the package promotion path* on page 58.

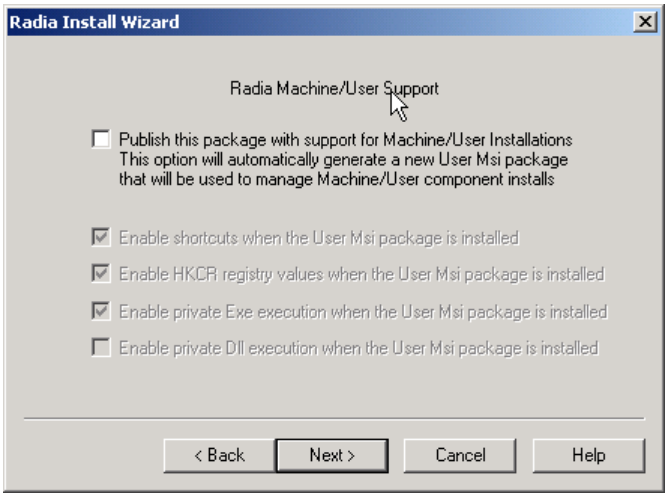

Figure 3.9 ~ Radia Install Wizard – Radia Machine/User Support.

Note

Radia Machine/User support is not supported in Radia versions below 3.x.

15. Select this check box if you would like to enable Radia machine/user support. A new user MSI package will be created and used to manage machine/user component installations. See *Enabling Machine/User Support* on page 200 for more information on this feature.

16. Click **Next**.

The **Enter the Package promotion path** dialog box opens.

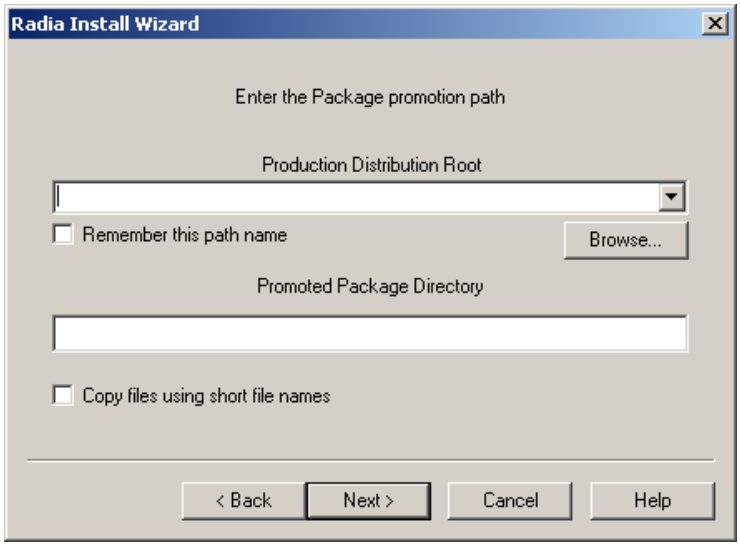

Figure 3.10 ~ Enter the package promotion path.

17. Enter the product distribution root and the promoted package directory information and click **Next**.

The Package analysis begins.

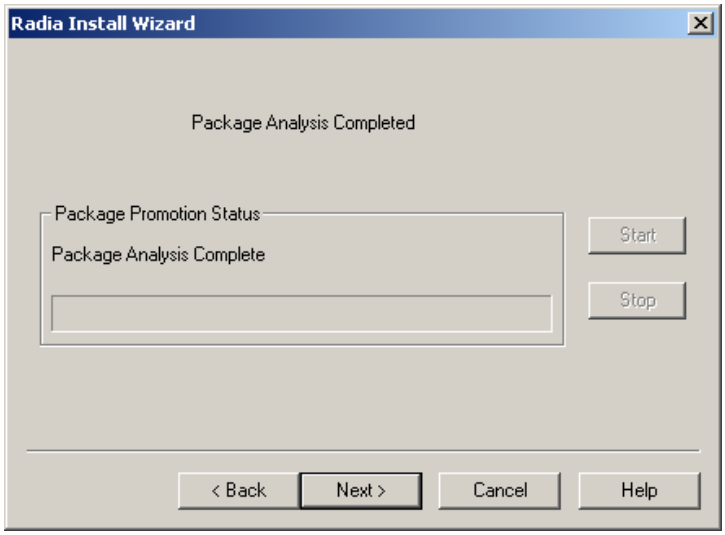

Figure 3.11 ~ Package analysis.

18. When the package analysis is complete, click **Next.**

The promotion process is complete dialog box opens.

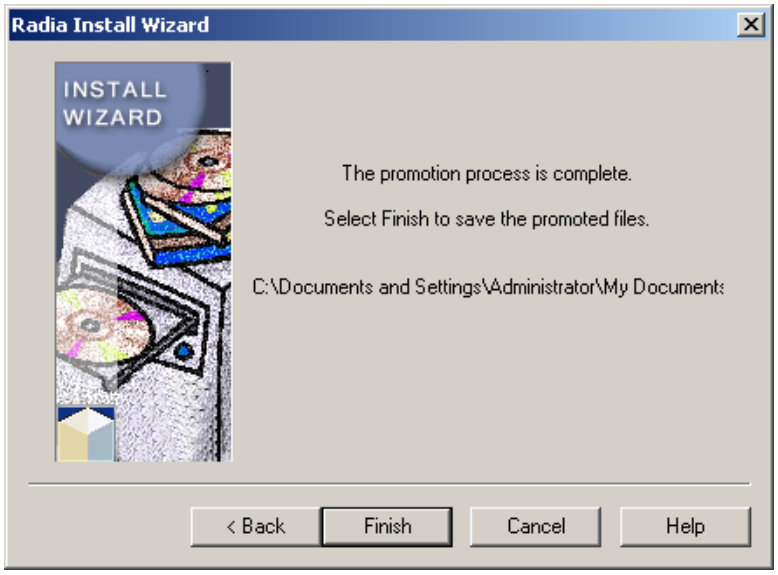

Figure 3.12 ~ Promotion process complete.

19. Click **Finish**.

You have successfully published your package and it is now ready for installation.

Basic Data Management

MSI Basic Mode

In contrast to Radia Advanced Package Management, where Radia provides on-going management of the components of an MSI package, Basic Package Management is used for packages that:

- Do not support administrative installation.
- The administrator does not want Radia to manage components with the exception of simply installing and uninstalling the application (No component verification or repair capability will be instrumented through the Radia Client).

Packages created with the Radia Packager for Windows Installer have a connection to the **MSIBASIC** class in the Radia Database. When the New Application Wizard is used to create a service from an application with one of these connections, you have the opportunity to define the MSI basic behavior methods.

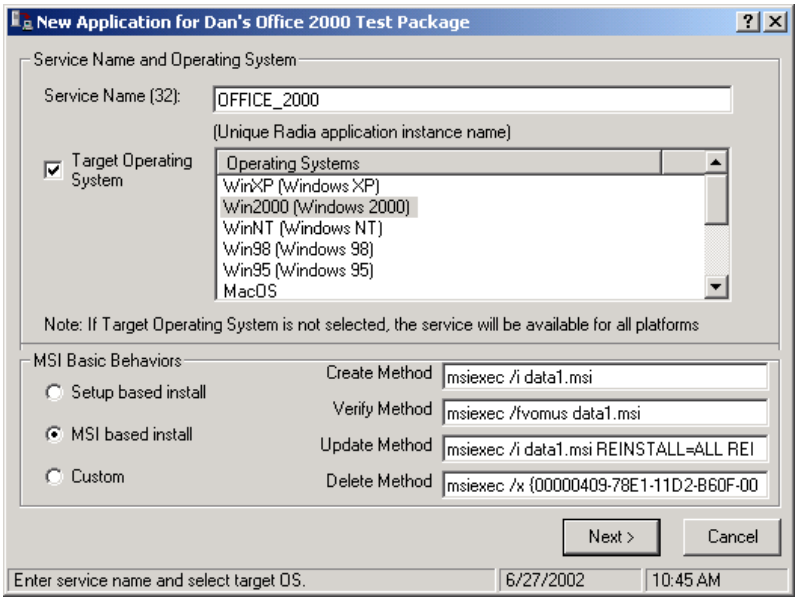

Figure 3.13 ~ Radia Install Wizard – New Application Wizard with MSI Basic Behaviors section.

Packages that cannot be administratively installed will default to Basic Package Management without all of the manageability features enabled in Radia.

Transforms

Transform (MST) – Transform files are used to modify existing Windows Installer applications to customize the installation to meet your needs.

Creating a Transform

If you already have a vendor supplied existing MSI to customize, it is recommended that you create a Windows Installer Transform (MST) with the Radia Install Wizard. When you create a transform with the Radia Install Wizard, you can do as little as modifying a property or as much as modifying pieces of the installation. The following exercise walks you through the creation of a user response transform for Microsoft Office 2000.

To create a user-response transform for a vendor-supplied Windows Installer package

1. From the Radia Packager for Windows Installer Program Group, start the **Radia Install Wizard**.

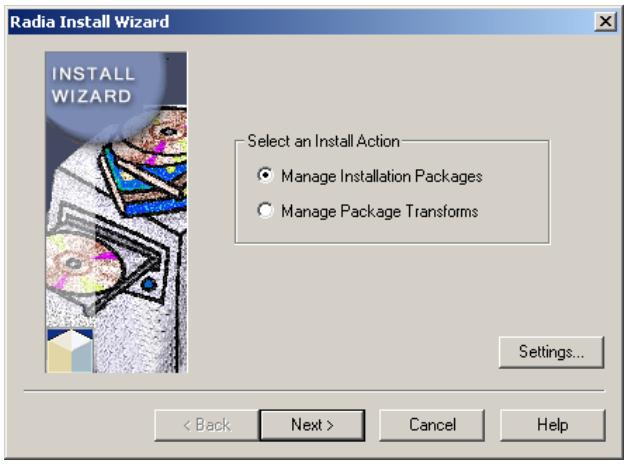

Figure 3.14 ~ Radia Install Wizard – Select an Install Action.

2. Select **Manage Package Transforms**.
3. Click **Next**.

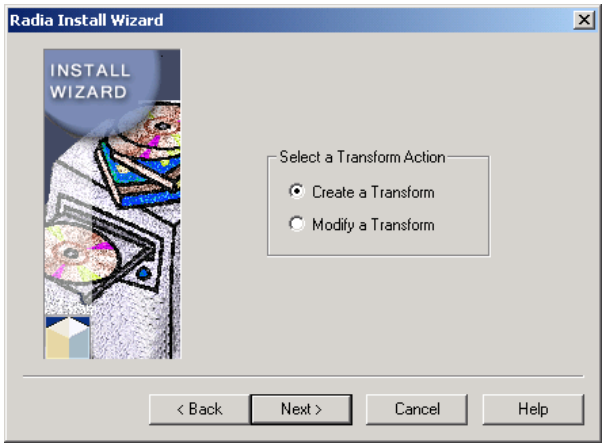

Figure 3.15 ~ Radia Install Wizard – Select a Transform Action.

4. Select **Create a Transform**.
5. Click **Next**.

   The **Enter the name of the Windows Installer Package file** dialog box opens.

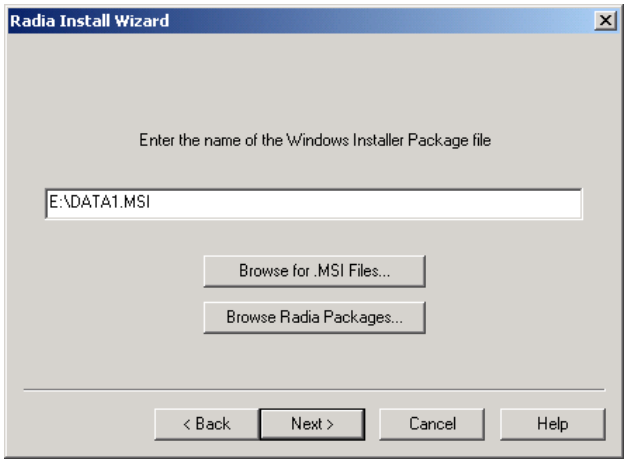

Figure 3.16 ~ Radia Install Wizard – Windows Installer package file name.

6. Browse to the DATA1.MSI included in the Office 2000 setup files.
7. Click **Next**.

The **Enter the name of the directory where you want to save the Transform file** dialog box opens.

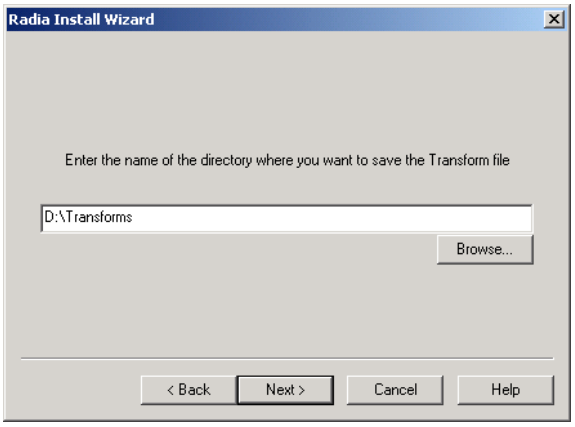

Figure 3.17 ~ Radia Install Wizard – Enter the name of the directory where you want to save the Transform file.

8. Enter the location where you would like to save the transform file, or click **Browse** to manually find the location.
9. Click **Next**.

   The **Enter a unique name for the Transform** dialog box opens.

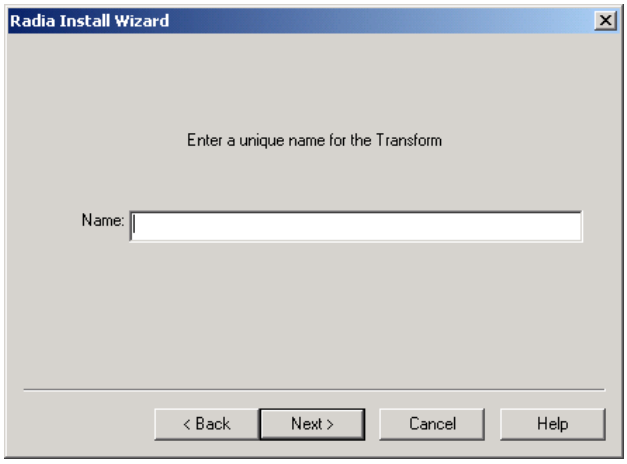

Figure 3.18 ~ Radia Install Wizard – Enter a unique name for the Transform.

10. Enter a unique name for the transform file.

11. Click **Next**.

The **Add existing Transforms to combine or edit with the Transform** dialog box opens.

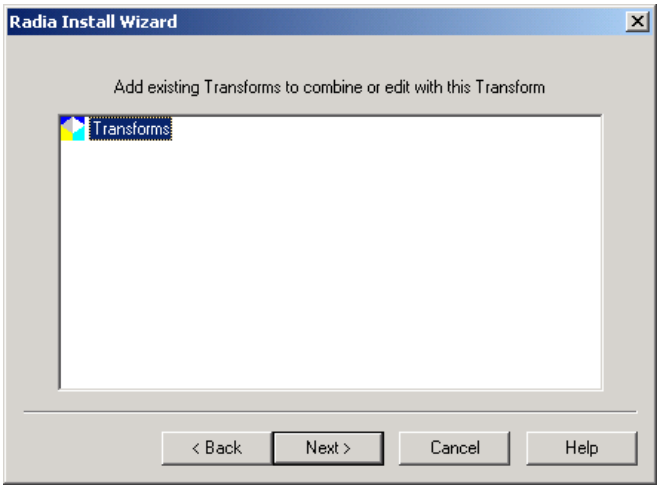

Figure 3.19 ~ Radia Install Wizard – Add existing Transforms to combine or edit with this Transform.

12. Select any other transforms to apply. These transforms will be combined into the final transform created.

13. Click **Next**.

The **Modify Transform Settings** dialog box opens.

Figure 3.20 ~ Radia Install Wizard – Modify Transform Settings.

14. Right-click on the transform name at the top of the Modify Transform Settings dialog box, and select **Execute User Interface** from the shortcut menu. This will start the user interface portion of the Office 2000 setup to gather information.

Figure 3.21 ~ User interface portion of Office2000 Setup.

15. The Radia Install Wizard starts the user interface for the MSI application. Note that the installation package itself is not being executed at this point, simply the user interface.

Proceed through the interface, selecting the information you would install for your end users until you reach the end of the interface.

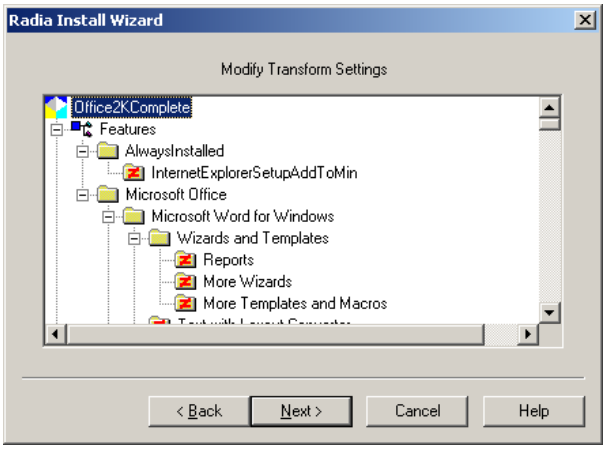

Figure 3.22 ~ Radia Install Wizard – Modify Transform Settings.

16. When the interface is finished, you can see the changes you made in the transform, and continue to modify them as necessary. Usually you will see changes made to the feature installation state or properties.

17. Click **Next**.

The **Suppress the following errors when applying the Transform** dialog box opens.

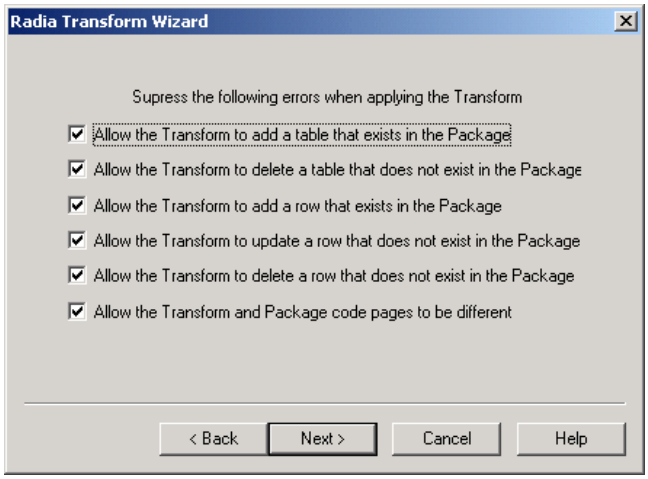

Figure 3.23 ~ Radia Transform Wizard – Suppress the following errors when applying the Transform.

18. Select the type of changes you would like to allow this transform to make.

19. Click **Next**.

The **Perform the following validation tests when applying the Transform** dialog box opens.

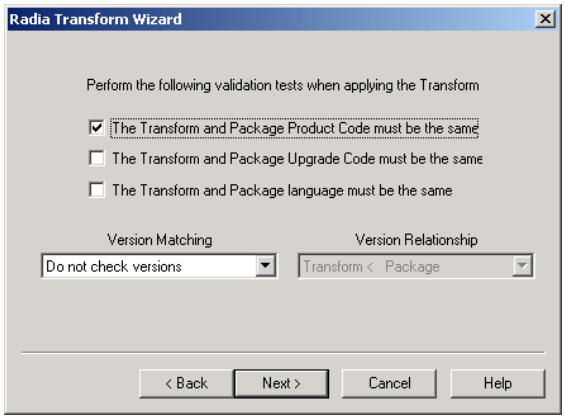

Figure 3.24 ~ Radia Transform Wizard – Perform the following validation tests when applying the Transform.

20. Select the information you want to use to validate that the transform applies to the package.

21. Click **Next**.

22. Wait for the transform analysis to finish.

23. Click **Finish** and exit the wizard.

Modifying a Transform

Radia Packager for Windows Installer gives you the ability to modify a transform, as well as add and modify the files and other resources in a package through a transform.

For example, Microsoft Office does not install the main template, **normal.dot**, for Word when the program is installed. Instead, the template is generated the first time Word is run. To include your own customized template in the Microsoft Office install of Word, you can add the template to your transform.

To modify a transform to include a custom template file

1. From the **Radia Packager for Windows Installer** program group, start the **Radia Install Wizard**.

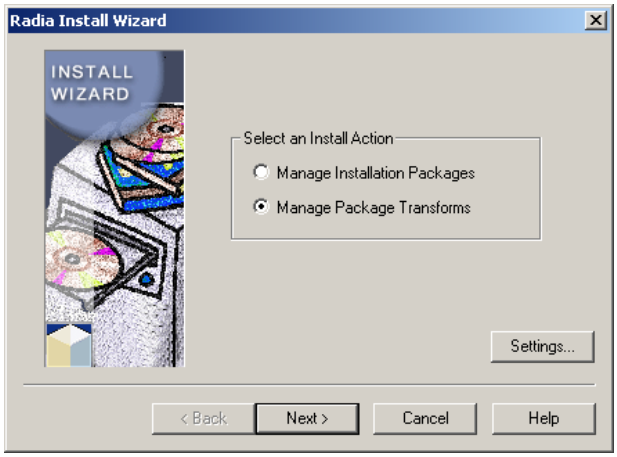

Figure 3.25 ~ Radia Install Wizard – Select an Install Action.

2. Select **Manage Package Transforms**.
3. Click **Next**.

   The **Select a Transform Action** dialog box opens.

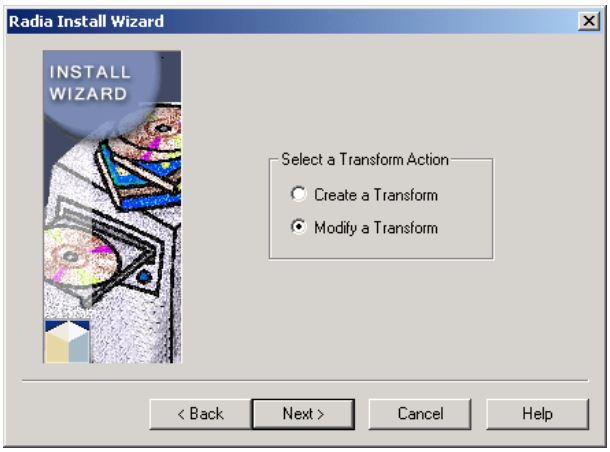

Figure 3.26 ~ Radia Install Wizard – Select a Transform Action.

4. Select **Modify a Transform**.
5. Click **Next**.

The **Enter the name of the Windows Installer Package file** dialog box opens.

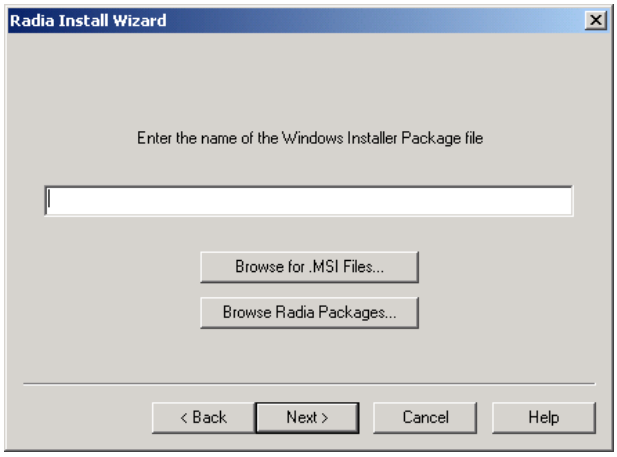

Figure 3.27 ~ Radia Install Wizard – Windows Installer package file name.

6. Enter the name and location of the MSI file for which the transform you would like to modify was created. Or use one of the **Browse** buttons to manually locate the file.
7. Click **Next**.

   The **Enter the name of the Transform file** to modify dialog box opens.

Figure 3.28 ~ Radia Install Wizard – Enter the name of the Transform file to modify.

8. Enter the name of the transform file to modify.
9. Click **Next**.
   The **Modify Transform Settings** dialog box opens.

Figure 3.29 ~ Radia Install Wizard – Modify Transform Setting.

10. Right-click and select **Add a File** from the shortcut menu.
    The **Enter the new target file information** dialog box opens.

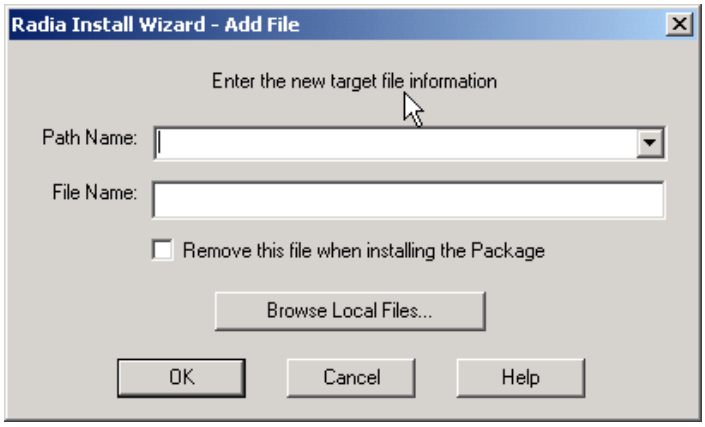

Figure 3.30 ~ Radia Install Wizard – Enter the new target file information.

11. Enter the target information for the file. The Normal.dot file is normally found in the Application Data folder under the Microsoft\Templates subdirectories. The Office installation already installs information to the Application Data folder (denoted as AppDataFolder by Windows Installer), which can be selected from the Path Name drop-down menu. Add **\Microsoft\Templates** to complete the path. Enter **Normal.dot** as the target file name.

12. Click **OK.**

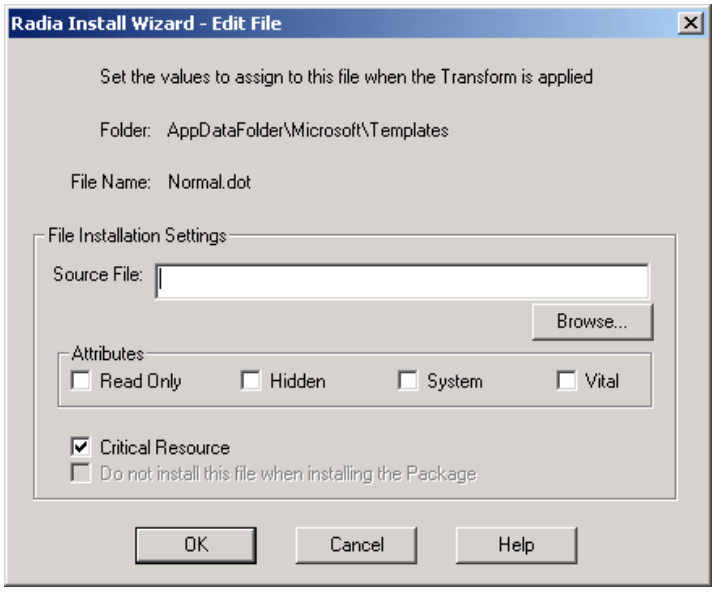

Figure 3.31 ~ Radia Install Wizard – File Installation Settings.

13. Browse for the location of the file you want to use as the source.

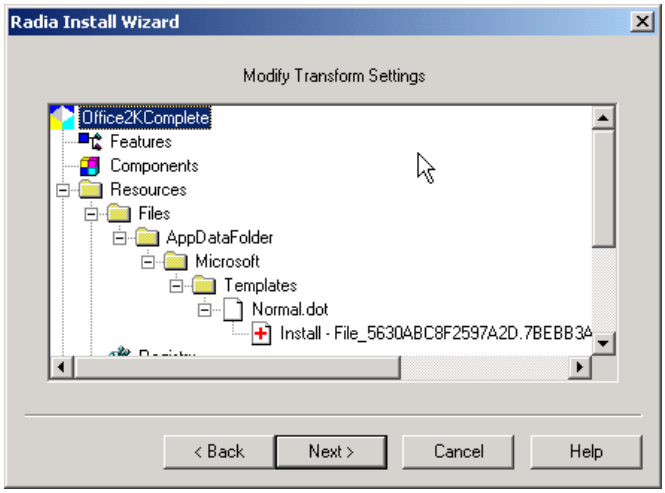

Figure 3.32 ~ Radia Install Wizard – Modify Transform Settings.

Your file is added to the transform.

20. Click **Next** to continue the transform modification. The remaining steps are the same as the creation of a transform.

When you add files to a transform, a .cab file is generated with the transform to contain the files. The name of this .cab file is generated to be unique. An informational text file is created with the same name as the MST to identify this .cab file.

Note

Windows Installer guidelines dictate that this .cab file (and therefore the transform) needs to be in the same directory as the MSI.

Summary

- Use the Radia Install Wizard to promote a package to a non-Radia distribution point, to manage the package's AIP, or to add Radia Management to the package with or without Machine/User Support.
- Create or modify Package Transforms using the Radia Install Wizard.

4

Upgrading a Package

At the end of this chapter, you will:

- Be able to package a new release for an existing product.
- Be able to package a new build with a Patch Package.

The Radia Extensions for Windows Installer simplifies the process of upgrading applications and tracking versions. Different versions of an application are maintained within a container called a *Product*. Each version of the application has its own Package within the Product. The Radia Extensions for Windows Installer maintains and organizes new versions of a Product to ease your management tasks.

Packaging a New Release

A **release** is a significant update that typically contains compiled code fixing bugs, and adds or removes product functionality, such as with a service pack. Packages created for new releases upgrade older release and build levels. When the newer release is installed, it replaces the older release, and any unused components are removed. New releases are not shipped as Patch Packages since they completely replace older releases.

Prerequisites

- A clean machine that has nothing but the operating system installed.
- Create your state file and library for the application update using the Radia State Wizard and the Radia Library Wizard.

To create a new release for the product

1. Start the Radia Packager for Windows Installer and select the Custom option, **Package components**.

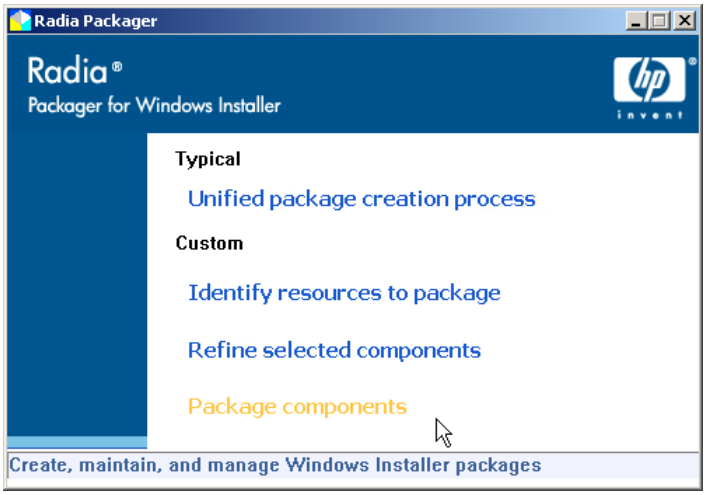

Figure 4.1 ~ Radia Packager for Windows Installer – Package components option.

The **Radia Package Wizard** opens.

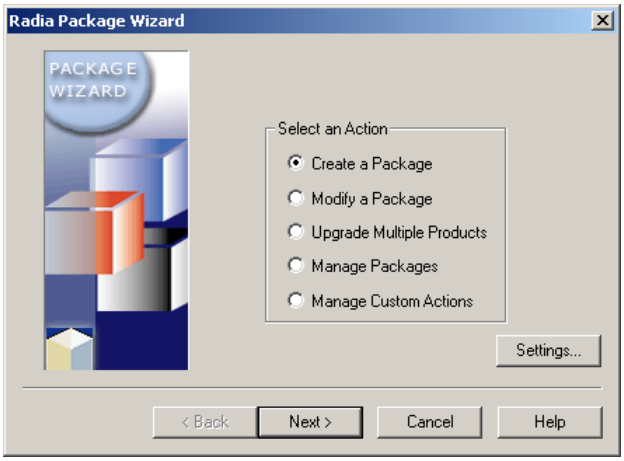

Figure 4.2 ~ Radia Package Wizard – Select an Action.

2. Select **Create a Package**.
3. Click **Next**.

   The **Select the Packaging Option** dialog box opens.

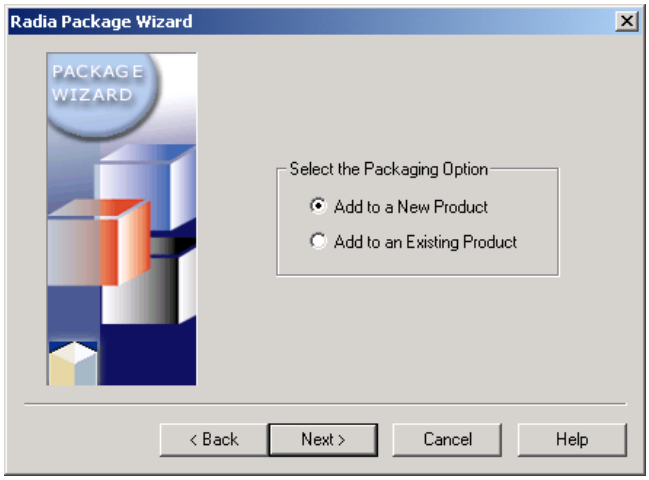

Figure 4.3 ~ Radia Package Wizard – Select the Packaging Option.

4. Select **Add to an Existing Product**.
5. Click **Next**.

   The **Select the Product to Update** dialog box opens.

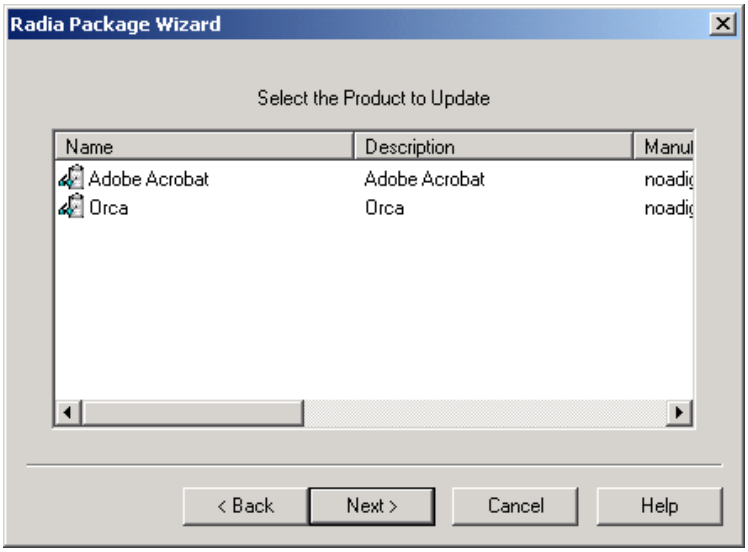

Figure 4.4 ~ Radia Package Wizard – Select the Product to Update.

6. Select the product that you want to update, such as **Acrobat Reader**.
7. Click **Next**.

   The **Select the Library from which the Package will be created** dialog box opens.

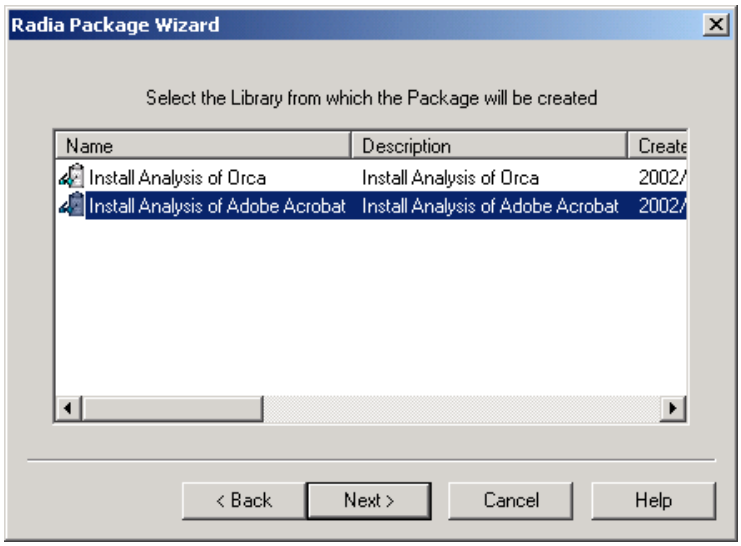

Figure 4.5 ~ Radia Package Wizard – Select the Library from which the Package will be created.

8. Select the library that you created, which will be used to create your new package release, such as Acrobat Reader 4.05.
9. Click **Next**.

   The **Product Upgrade Mode** dialog box opens.

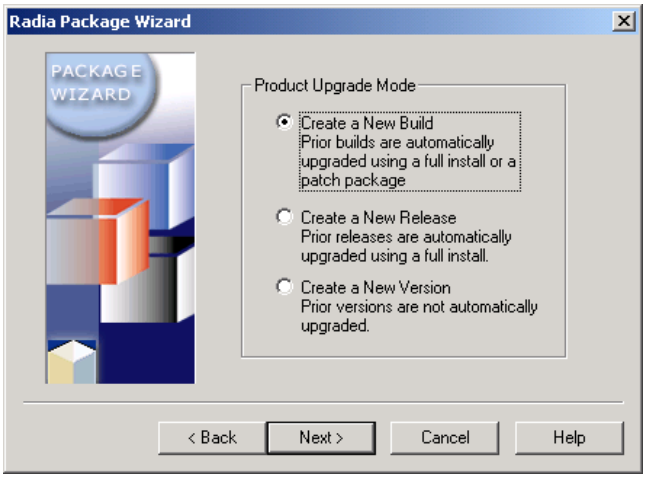

Figure 4.6 ~ Radia Package Wizard – Product Upgrade Mode.

10. Select **Create a New Release**. This removes Acrobat Reader 3.01 when Acrobat Reader 4.0.5 is installed.

11. Click **Next**.

The **Select the Product Release and Build Numbers** dialog box opens. The **Release** is automatically incremented. You can also modify the release or build numbers.

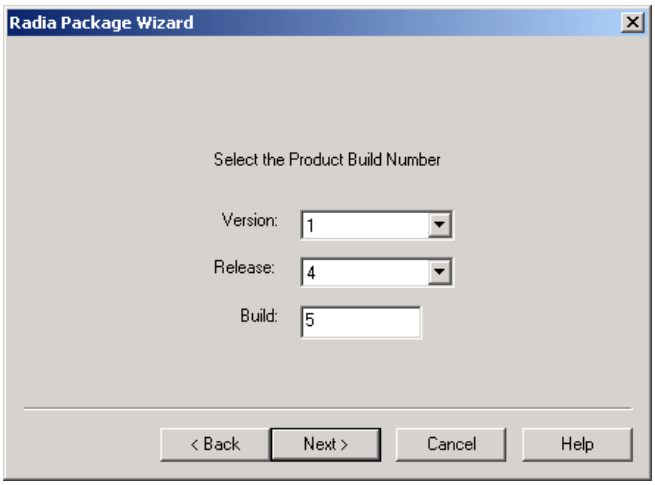

Figure 4.7 ~ Radia Package Wizard – Select the Product Version, Release and Build Numbers.

12. Click **Next**.

The **Advanced Options** dialog box opens. Use this dialog box to modify different information and installation settings embedded in your Windows Installer package. See *Using Radia Package Wizard Advanced Functions* on page 167 for more information.

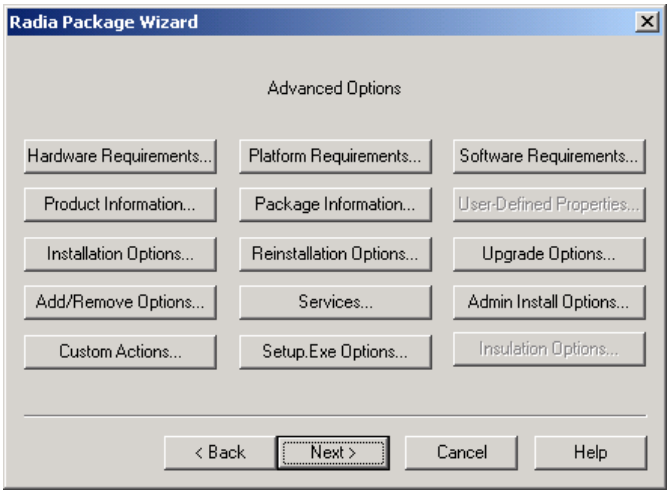

Figure 4.8 ~ Radia Package Wizard – Advanced Options.

13. Click **Next**.

The Radia Package Wizard performs a quick analysis of the files to include in the package.

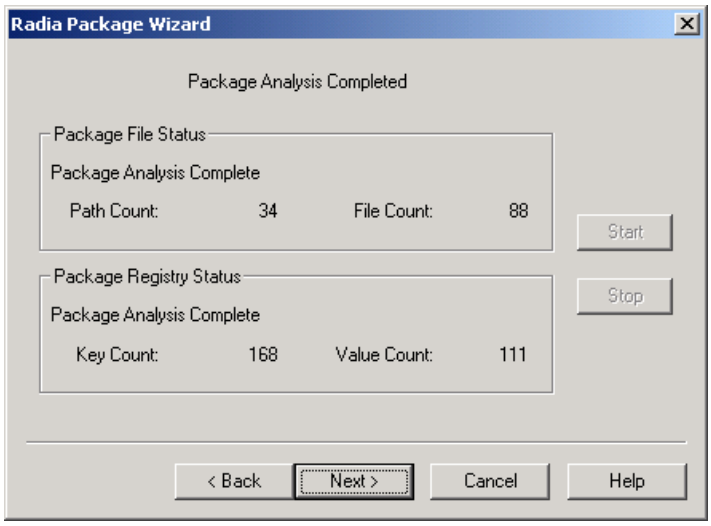

Figure 4.9 ~ Radia Package Wizard – Package Analysis Completed.

14. When the analysis is finished, click **Next**.

The **Package Results** dialog box opens. This dialog box displays all of the files and information included in the Package, and highlights the files, registry settings, etc. that will be added, removed or modified when your Windows Installer package is installed.

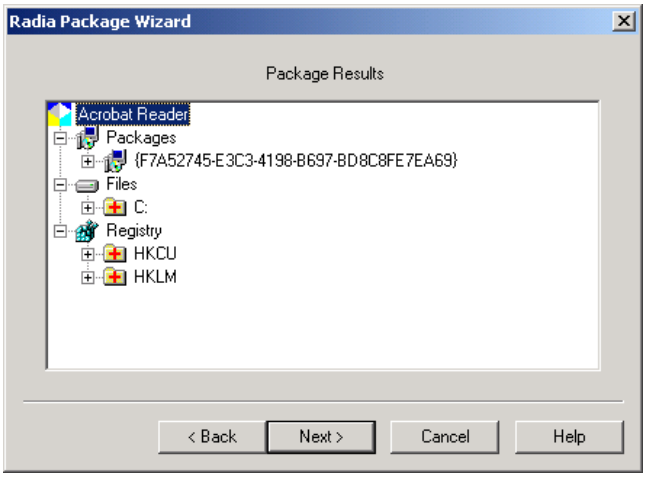

Figure 4.10 ~ Radia Package Wizard – Package Results.

15. Click **Next**.

The **Package File Options** dialog box opens.

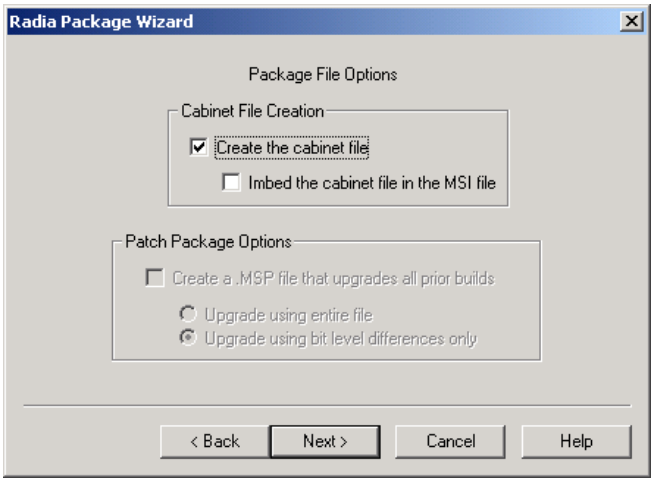

Figure 4.11 ~ Radia Package Wizard – Package File Options.

16. Leave **Create the cabinet file** checked. In most cases, you will want to create the cabinet file.

17. Click **Next** to create the Windows Installer package.

18. Click **Next** again.

The **package creation is complete** dialog box opens.

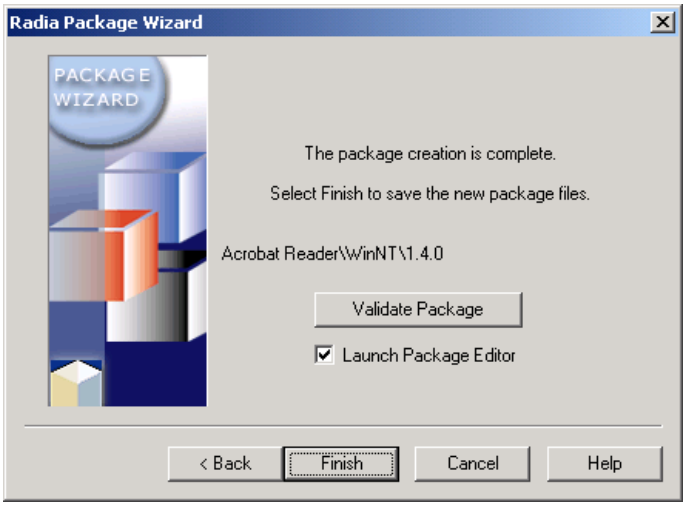

Figure 4.12 ~ Radia Package Wizard – Package creation is complete.

- Click **Validate Package** to check the package.
- If you would like to open the MSI file in the default MSI editor on your computer (**Radia MSIEdit** is the default editor after installation of Radia Extensions for Windows Installer) leave **Launch Package Editor** selected.

19. Click **Finish**.

20. Use Windows Explorer to see that the new package is added to the Product directory structure. Notice how the Package files are organized by: Version, Release, and Build number, for easy management.

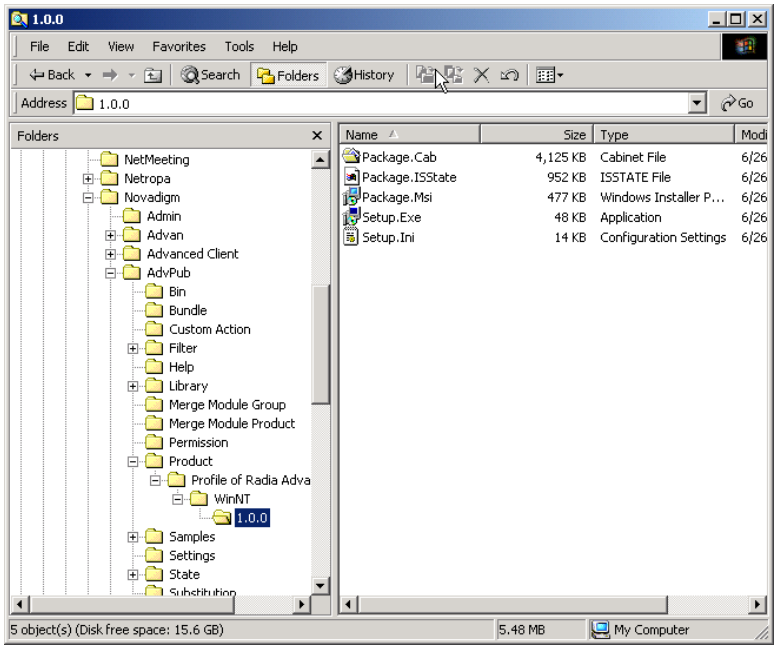

Figure 4.13 ~ Product directory structure.

Packaging a New Build with a Patch Package

A **build** is a small update or bug fix. Packages created for new builds upgrade older build levels. When the newer build is installed, the older build is updated and any unused components are removed, if necessary. New builds can be shipped as patch packages to minimize the amount of data shipped during the installation process to only the necessary changes.

For example, if an update for Acrobat Reader 4.0 is available you will want to create a new build that updates your users to Acrobat Reader 4.05.

Prerequisites

- Create your state file and Library for the updated package using the Radia State Wizard and the Radia Library Wizard.

To create an update with a Patch Package

1. Start the Radia Packager for Windows Installer and select the Custom option **Package components**.

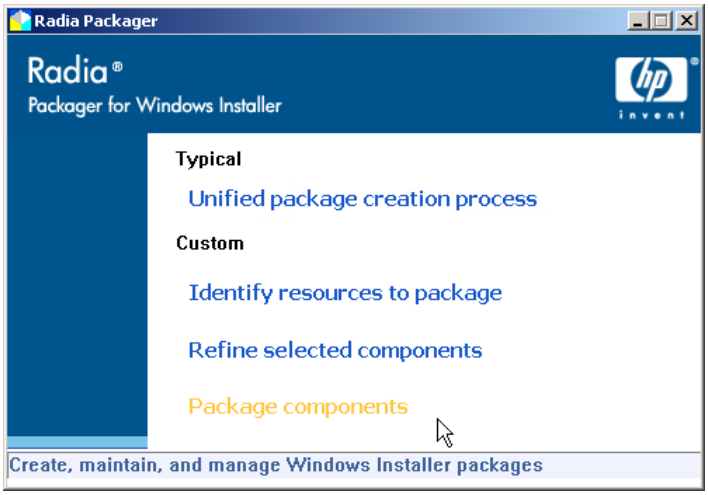

Figure 4.14 ~ Radia Packager for Windows Installer – Package components option.

The **Radia Package Wizard** opens.

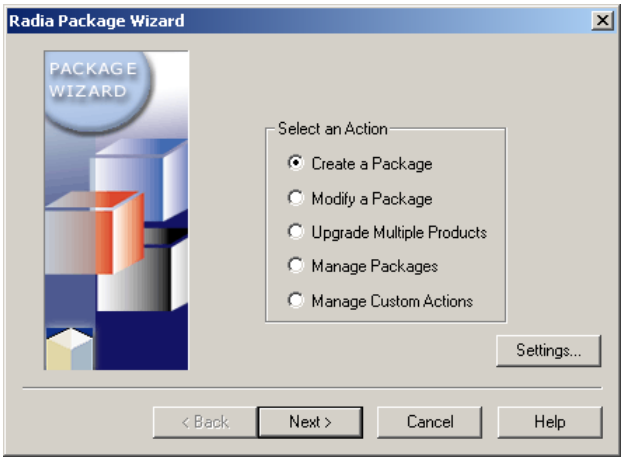

Figure 4.15 ~ Radia Package Wizard – Select an Action.

2. Select **Create a Package**.
3. Click **Next**.
   The **Select the Packaging Option** dialog box opens.

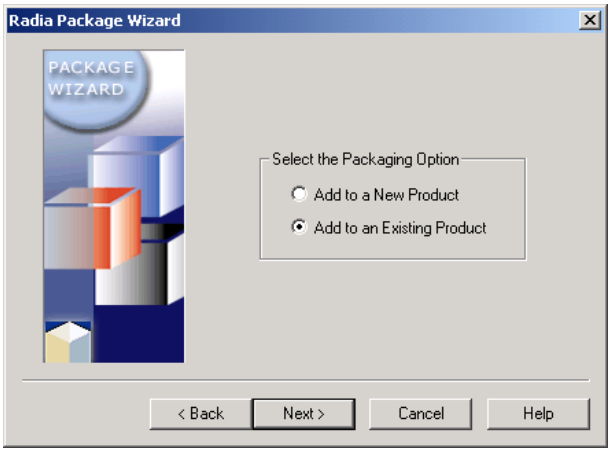

Figure 4.16 ~ Radia Package Wizard – Select the Packaging Option.

4. Select **Add to an Existing Product**.

5. Click **Next**.

The **Select the Product to Update** dialog box opens.

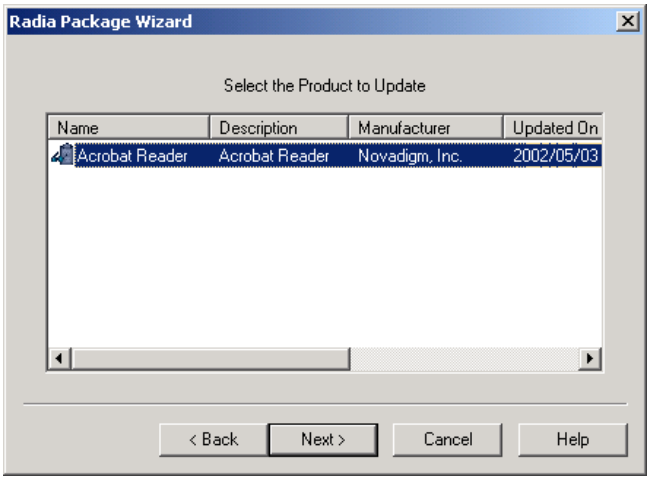

Figure 4.17 ~ Radia Package Wizard – Select the Product to Update.

6. Select the product to update, such as **Acrobat Reader**.

7. Click **Next**.

The **Select the Library from which the Package will be created** dialog box opens.

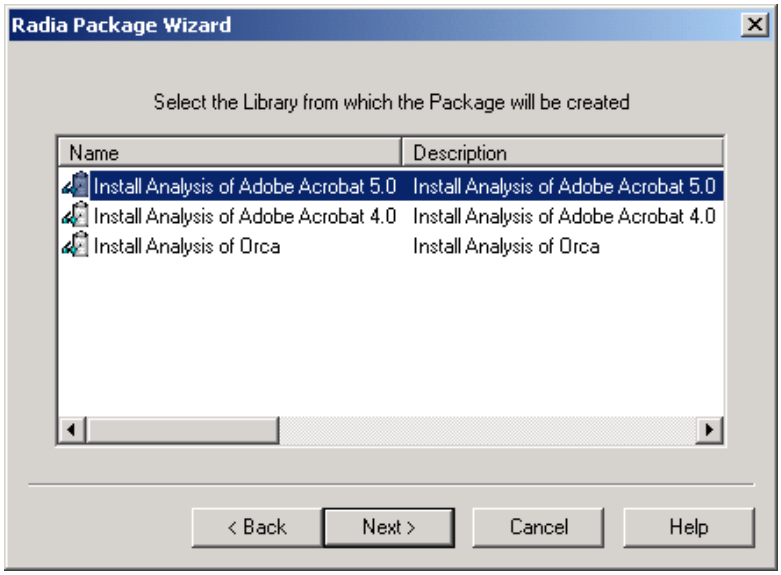

Figure 4.18 ~ Radia Package Wizard – Select the Library from which the Package will be created.

8. Select the new library that you created, which will be used to create your new package release.
9. Click **Next**.

   The **Product Upgrade Mode** dialog box opens.

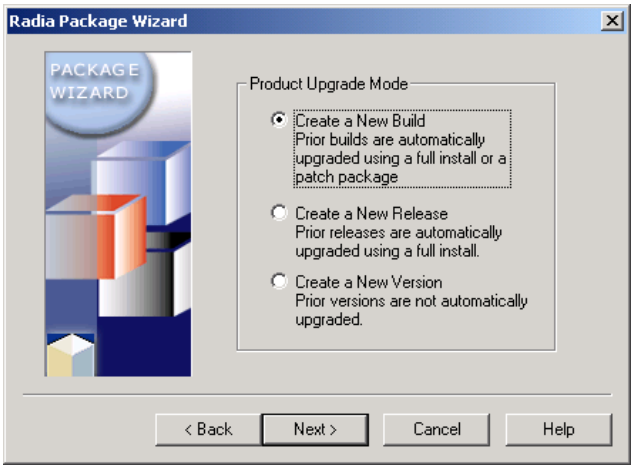

Figure 4.19 ~ Radia Package Wizard – Product Upgrade Mode.

10. Select **Create a New Build**.

11. Click **Next**.

The **Select the Product Build Number** dialog box opens.

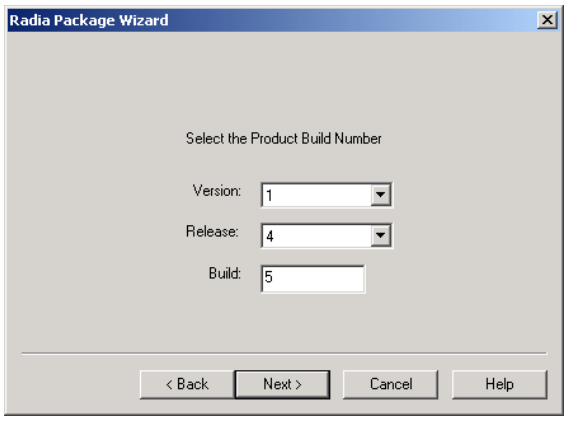

Figure 4.20 ~ Radia Package Wizard – Select the Product Version, Release and Build Numbers.

12. Click **Next**.

The Radia Package Wizard automatically updates the build number of the newest release. In this example, change the build number to 5.

13. Click **Next**.

The **Advanced Options** dialog box opens.

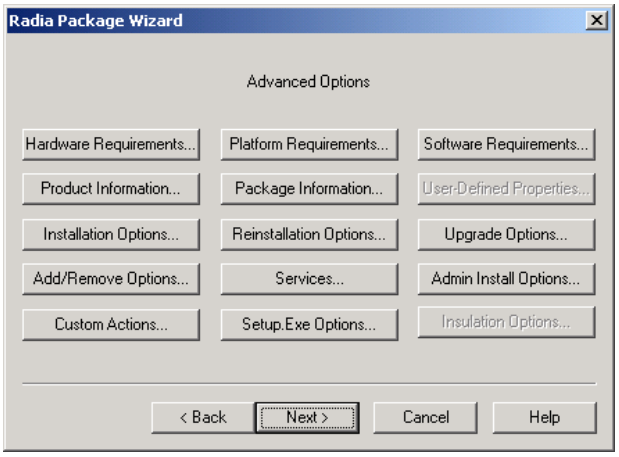

Figure 4.21 ~ Radia Package Wizard – Advanced Options.

14. Click **Next**.

The **Package Analysis Completed** dialog box opens.

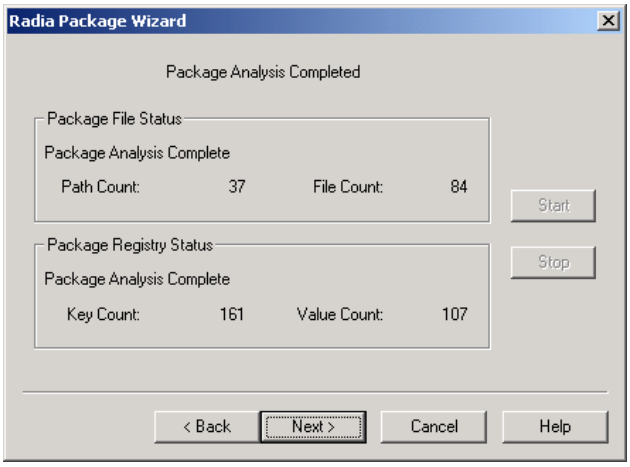

Figure 4.22 ~ Radia Package Wizard – Package Analysis Completed.

15. Click **Next**.

The **Package Results** dialog box opens.

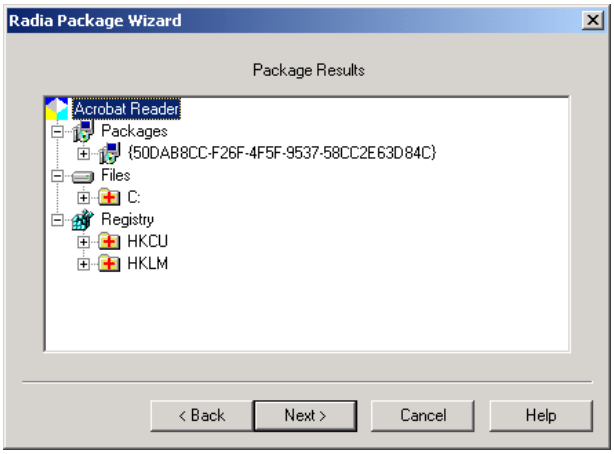

Figure 4.23 ~ Radia Package Wizard – Package Results.

16. Click **Next**.

The **Package File Options** dialog box opens.

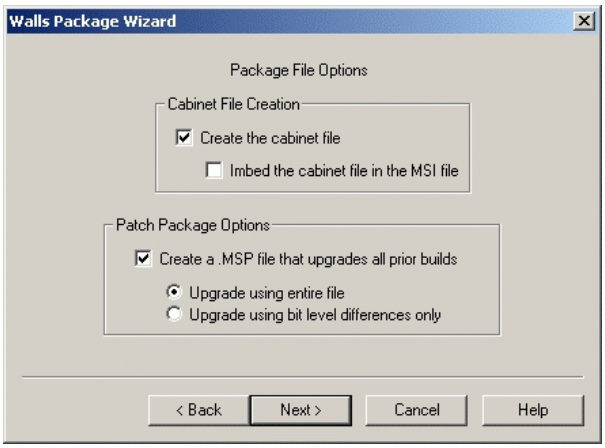

Figure 4.24 ~ Radia Package Wizard – Package File Options.

17. Select **Create an .MSP file that upgrades all prior builds**. An .MSP file is a Patch Package.
18. Select **Upgrade using the entire file**.
19. Click **Next**.

    The package is created.

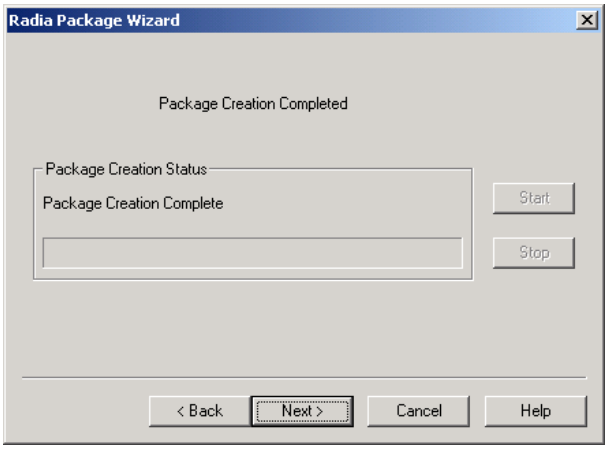

Figure 4.25 ~ Radia Package Wizard – Package Creation Status.

20. Click **Next**.

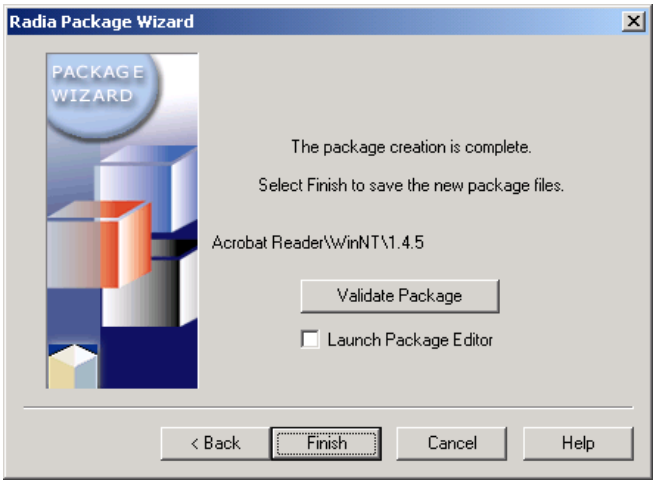

Figure 4.26 ~ Radia Package Wizard – Package creation is complete.

- Click **Validate Package** to check your package.
- If you would like to open the MSI file in the default MSI editor on your computer (**Radia MSIEdit** is the default editor after installation of Radia Extensions for Windows Installer), leave **Launch Package Editor** checked.

21. Click **Finish** to save the new package.

In the new build's directory, there is a Patch Package (.MSP) and a full Package (.MSI). This combination allows you to apply the patch to machines that already have 1.4.0 installed or to install the full package to machines that don't have the application installed.

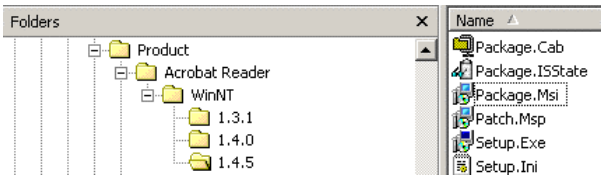

Figure 4.27 ~ Patch package and full package.

Summary

- Use the **Package components** option of the Radia Packager for Windows Installer to package a new release or package a new build with a patch package.
- This **Package components** option launches the **Radia Package Wizard**.

5

Using Advanced Functions

At the end of this chapter, you will be able to:

- Combine and manage state files.
- Modify a library.
- Create permissions.
- Use substitutions.
- Rebuild a package.
- Edit a package template.
- Use Radia Advanced Features Machine/User support

Note to I-Lab Users

Radia Advanced Features are only licensed in the Radia Extensions I-Lab product for testing purposes within a laboratory environment.

Using Radia State Wizard Advanced Functions

The Radia Extensions for Windows Installer provides you with a front-end for performing many different state file tasks, ranging from creating state files to combining states to deleting states. This chapter covers the functions of the Radia State Wizard that have not been previously introduced in the earlier chapters on creating packages. To access these functions from the Radia Packager for Windows Installer, select the Custom option **Identify resources to package.**

Combining States

The Radia State Wizard allows you combine two state files together to create one of the following combined states:

- **Delta State**
  Includes only the different resources between two states.
- **Intersection State**
  Includes only the resources common to both states. Create an Intersection State to identify potential resource conflicts between applications.
- **Union State**
  Includes all the resources of both states, regardless of whether the resources are shared between the two states. Typically, union state files are created using an impact and profile state of the same application.

After you create a combined state file, you have many options:

- View this state file using the **Manage States** option in the Radia State Wizard.
- Import this state file into the Radia Application Knowledge Base using the Radia Configuration Analyzer Administrator's console.
- Export the state file into .csv format and view with another spreadsheet package.
- Use this state file to create another state file.

To combine two state files

> **Note**
>
> In this procedure, you will create a union state. To create the union state, an initial Install Analysis must exist as well as a second installation analysis created after modifying a few of the application's settings.

1. Start the Radia Packager for Windows Installer and select the Custom option, **Identify resources to package.**

   OR

   From the Radia Packager for Windows Installer Program Group, select **State Wizard**.

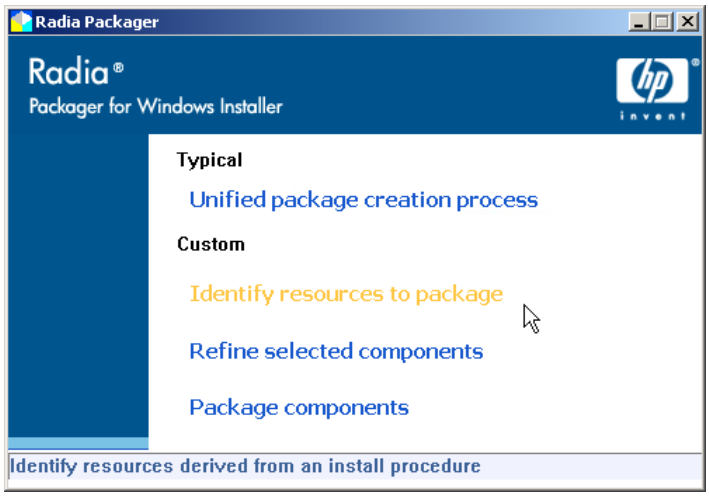

Figure 5.1 ~ Select the Custom option: Identify resources to package.

The **Select an Action** dialog box of the Radia State Wizard opens.

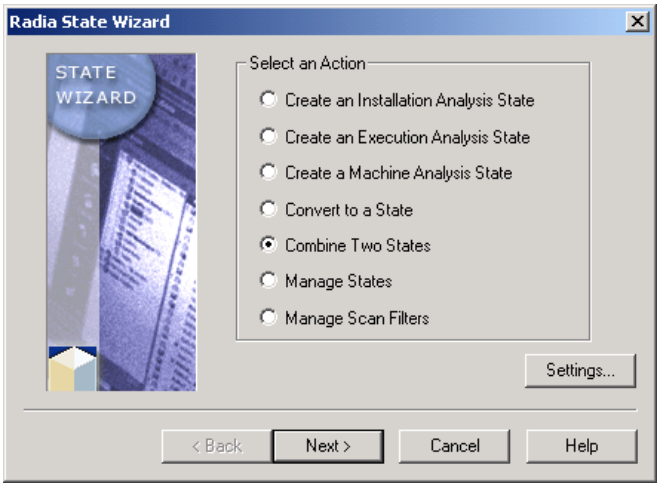

Figure 5.2 ~ Radia State Wizard – Select an Action.

2. Select **Combine Two States**.

3. Click **Next**.
   The **Select Combined State Creation Technique** dialog box opens.

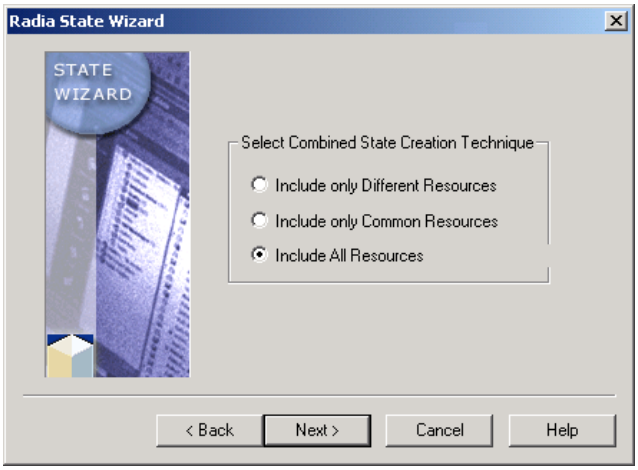

Figure 5.3 ~ Radia State Wizard – Select Combined State Creation Technique.

4. Select **Include All Resources**
5. Click **Next**.
   The **Select the First State to be Combined** dialog box opens.

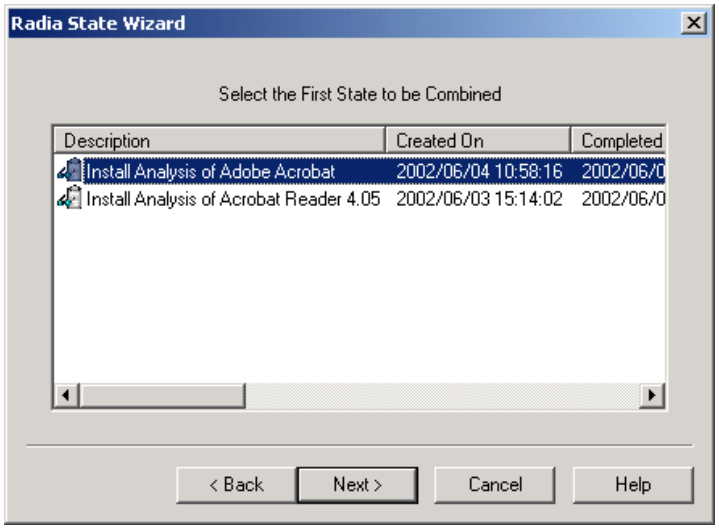

Figure 5.4 ~ Radia State Wizard – Select the First State to be Combined.

6. Select the first State to be combined.
7. Click **Next**.

   The **Select the Second State to be Combined** dialog box opens.

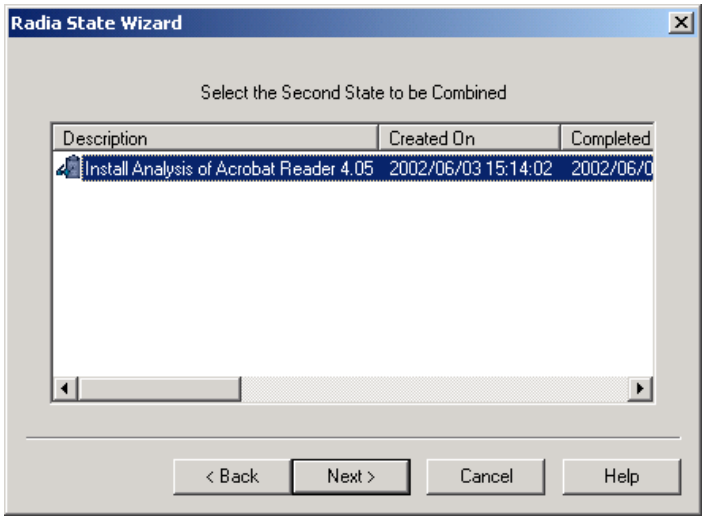

Figure 5.5 ~ Radia State Wizard – Select the Second State to be Combined.

8. Select the second state to be combined.

9. Click **Next**.

The **Description** dialog box opens.

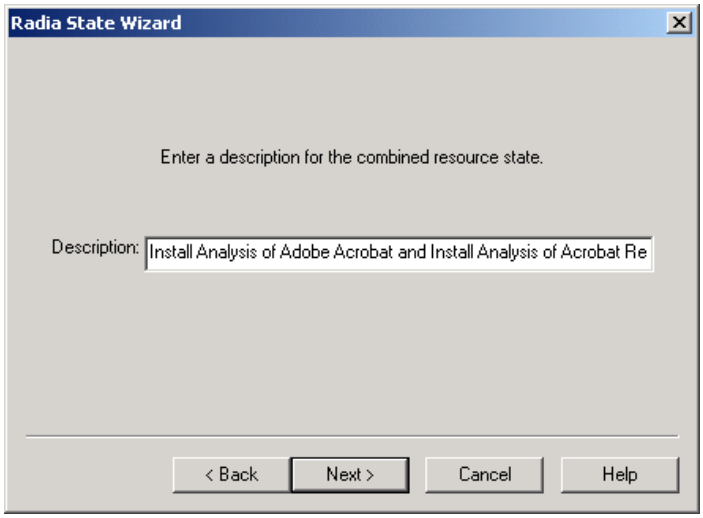

Figure 5.6 ~ Radia State Wizard – Enter a Description.

10. In the **Description** text box, type a description for the combined state file.

11. Click **Next**.

The states are combined.

12. When the combination analysis is done, click **Next**.

The **State Data Collection Results** dialog box opens.

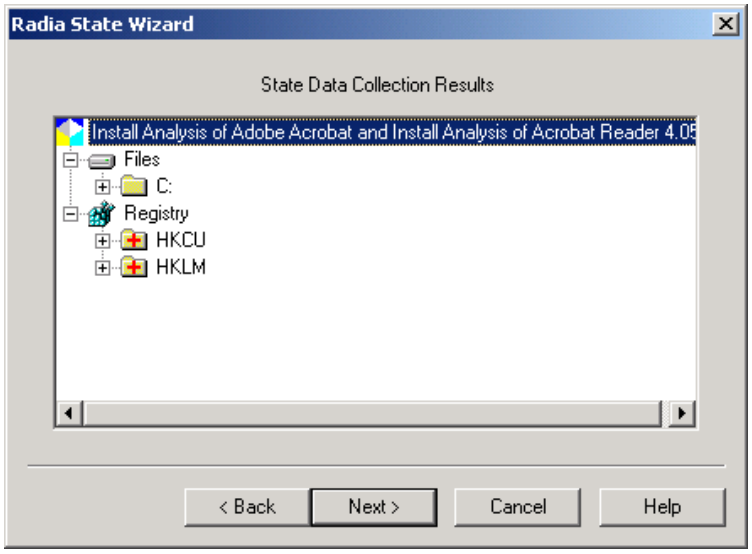

Figure 5.7 ~ Radia State Wizard – State Data Collection Results.

13. Click **Next**.

14. Click **Finish**.

Managing States

With the Radia State Wizard, you can also perform basic state management tasks such as deleting or viewing states. You can access the Radia State Wizard by selecting the Radia Packager for Windows Installer Custom option, **Identify resources to package**, or from the Radia Packager for Windows Installer program group.

To view states files

1. Start the Radia Packager for Windows Installer and select the Custom option, **Identify resources to package.**

OR

From the Radia Packager for Windows Installer Program Group, select **State Wizard**.

The **Select an Action** dialog box of the Radia State Wizard opens.

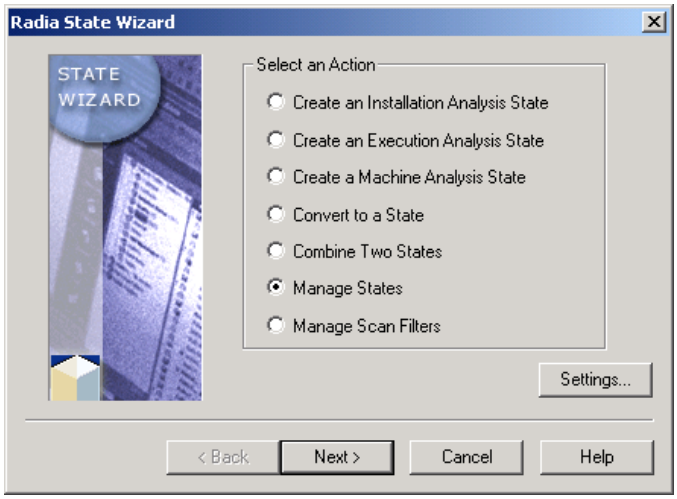

Figure 5.8 ~ Radia State Wizard – Select an Action.

2. Select **Manage States**.
3. Click **Next**.

   The **Select a State Management Action** dialog box opens.

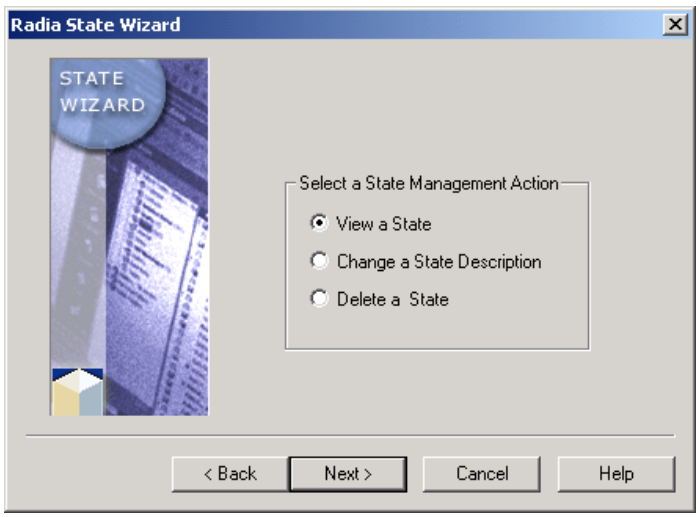

Figure 5.9 ~ Radia State Wizard – Select a State Management Action.

4. Select **View a State**.
5. Click **Next**.

   The **Select the State to be Managed** dialog box opens.

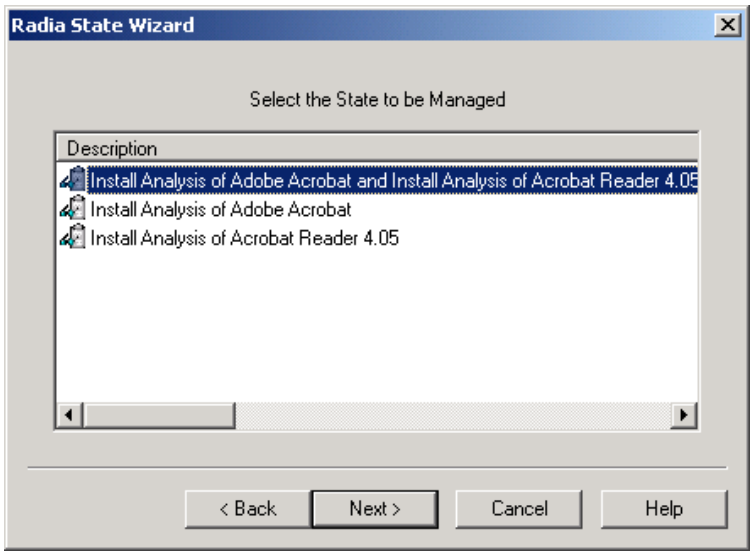

Figure 5.10 ~ Radia State Wizard – Select the State to be Managed.

Shortcut Tip

On any state selection dialog box, right-click the state that you want to view and from the shortcut menu, select **View**.

6. Click **Next** to view the state.

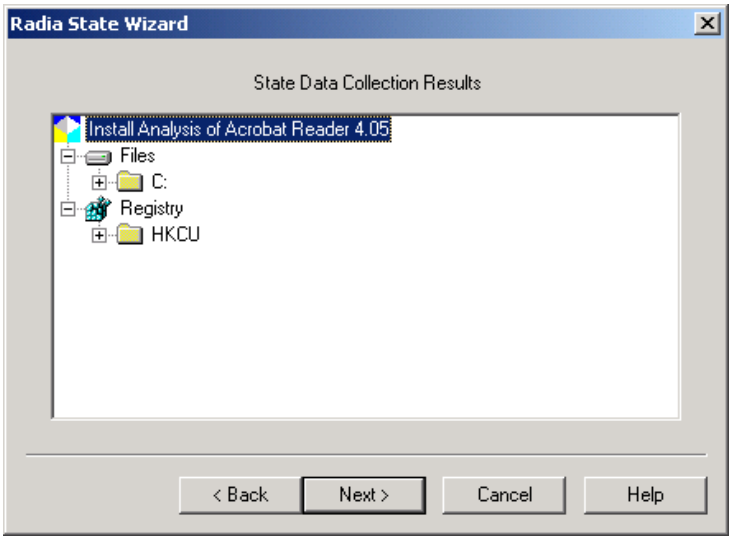

Figure 5.11 ~ Radia State Wizard – State Data Collection Results.

7. Click **Next**.
8. Click **Finish** to exit the Radia State Wizard.

Changing a State Description

You may find the need to change a state description. This description is used by both the Radia State Wizard and the Radia Configuration Analyzer to describe the state.

To change the description of a state file

1. Start the Radia Packager for Windows Installer and select the Custom option, **Identify resources to package**.

   OR

   From the Radia Packager for Windows Installer Program Group, start the **Radia State Wizard**.

   The **Select an Action** dialog box opens.

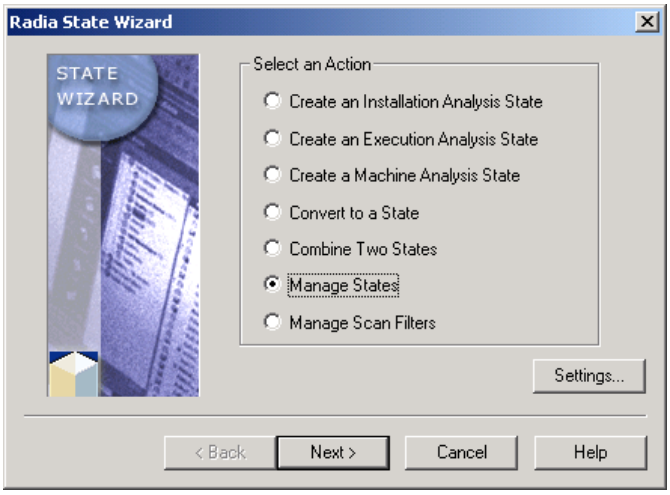

Figure 5.12 ~ Radia State Wizard – Select an Action.

2. Select **Manage States**.
3. Click **Next**.

   The **Select a State Management Action** dialog box opens.

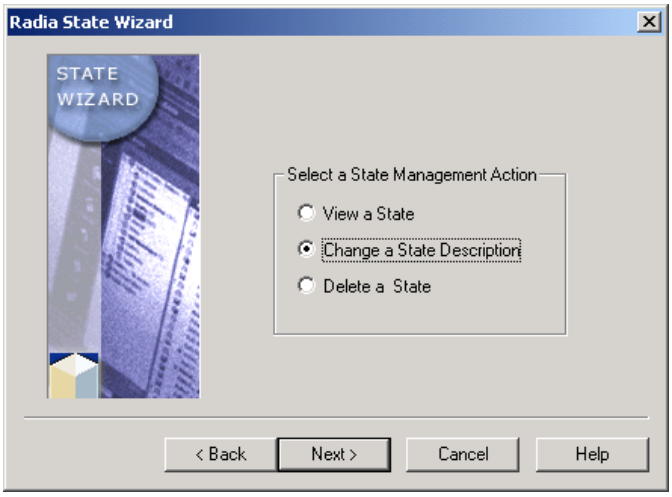

Figure 5.13 ~ Radia State Wizard – Select a State Management Action.

4. Select **Change a State Description**.
5. Click **Next**.

   The **Select the State to be Managed** dialog box opens.

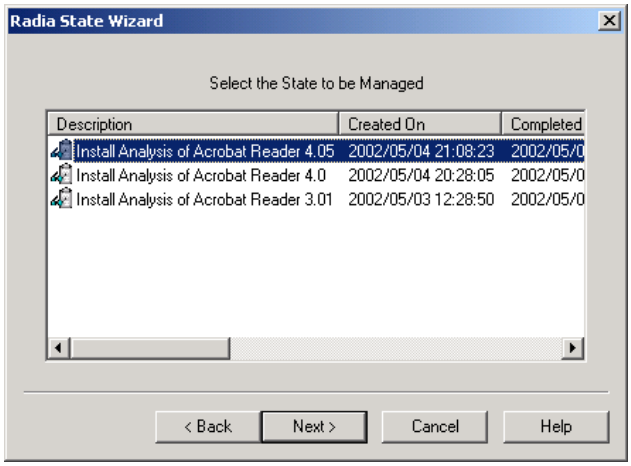

Figure 5.14 ~ Radia State Wizard – Select the State to be Managed.

6. Select the state for which you would like to change the description.
7. Click **Next**.

   The **Description** dialog box opens.

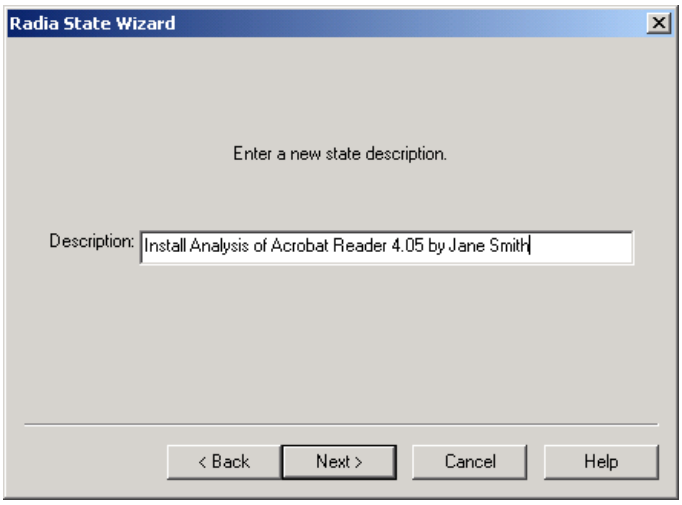

Figure 5.15 ~ Radia State Wizard – Enter a Description.

8. In the **Description** text box, type a new description.
9. Click **Next**.

   The **Select Finish to change the description for the selected state** dialog box opens.

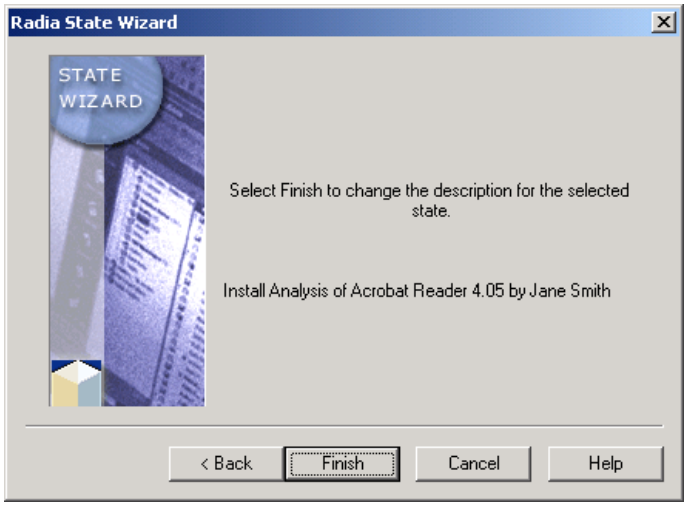

Figure 5.16 ~ Radia State Wizard – Select Finish to change the description for the selected state.

10. Click **Finish** to save your change.

Deleting a State

To delete a state

1. Start the Radia Packager for Windows Installer and select the Custom option, **Identify resources to package**.

OR

From the Radia Packager for Windows Installer Program Group, start the **Radia State Wizard**.

The **Select an Action** dialog box opens.

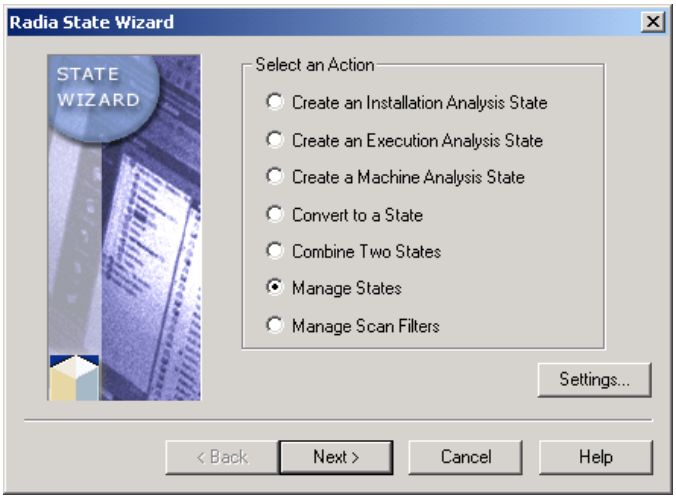

Figure 5.17 ~ Radia State Wizard – Select an Action.

2. Select **Manage States**.
3. Click **Next**.

   The **Select a State Management Action** dialog box opens.

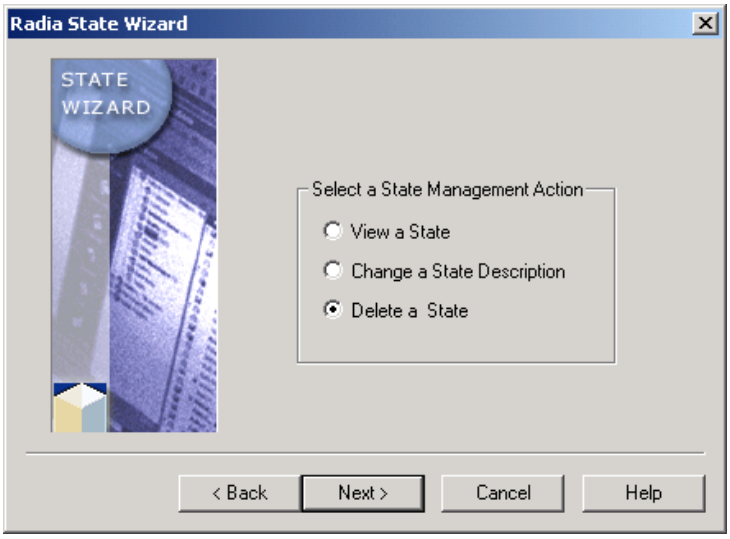

Figure 5.18 ~ Radia State Wizard – Select a State Management Action.

4. Select **Delete a State**.
5. Click **Next**.

   The **Select the State to be Managed** dialog box opens.

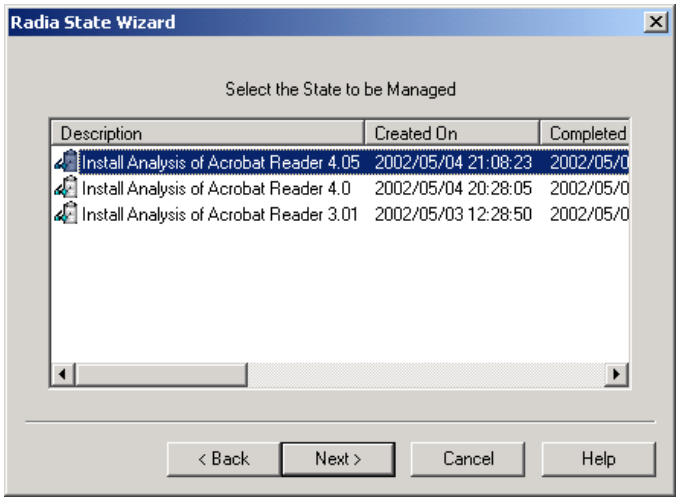

Figure 5.19 ~ Radia State Wizard – Select the State to be Managed.

Shortcut Tip

On any state selection dialog box, right-click the state that you want to delete and from the shortcut menu, select **Delete**.

6. Click **Finish** to delete the state.

Using Radia Library Wizard Advanced Functions

Many important packaging and application management functions are performed within a library by the Radia Library Wizard. To access these library functions from the Radia Packager for Windows Installer, select the Custom option **Refine selected components.**

Modifying a Library

Modifying Installation Settings

When Acrobat Reader was installed for analysis, it was installed to **C:\Acrobat**. If you want to allow the user to enter the location for Acrobat upon installation, you can modify the target path for an application so that the user can enter it. To do this you will modify the installation settings so that the path for Acrobat Reader will be substituted with information entered by the user.

To modify the target path of an application installation

1. Start the Radia Packager for Windows Installer and select the Custom option, **Refine selected components.**

   OR

   From the Radia Packager for Windows Installer Program Group, start the **Radia Library Wizard**.

   The **Select an Action** dialog box opens.

Figure 5.20 ~ Radia Library Wizard – Select an Action.

2. Select **Modify a Library**.
3. Click **Next**.

   The **Select the Type of Modification** dialog box opens.

Figure 5.21 ~ Radia Library Wizard – Select the Type of Modification.

4. Select **Modify Installation Settings**.
5. Click **Next**.

   The **Select the Library to Modify** dialog box opens.

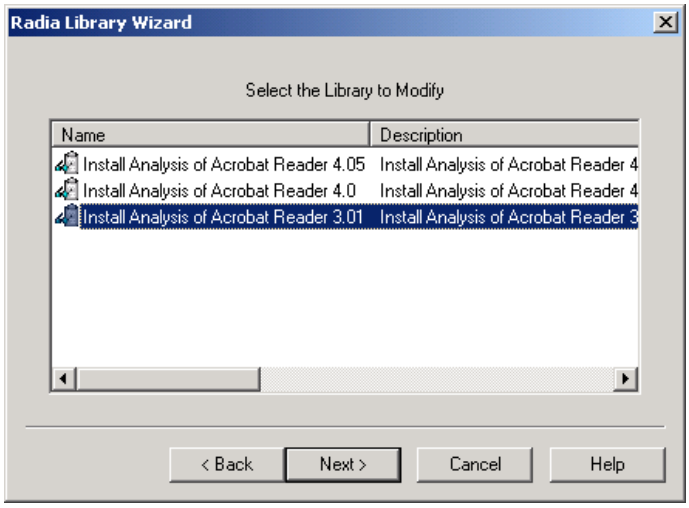

Figure 5.22 ~ Radia Library Wizard –Select the Library to Modify.

6. Select the library that you want to modify, such as **Install Analysis of Acrobat Reader 3.01**.
7. Click **Next**.

   The **Modify Library Installation Settings** dialog box opens.

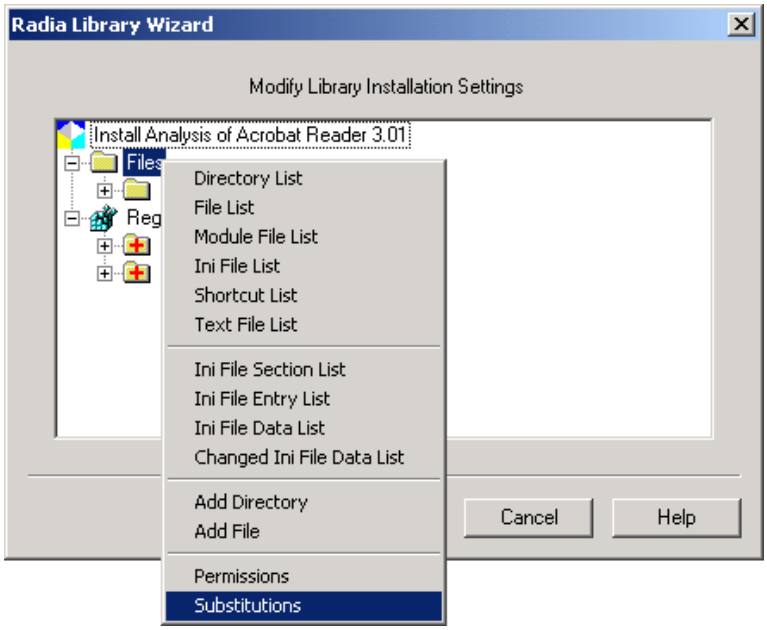

Figure 5.23 ~ Radia Library Wizard – Modify Library Installation Settings.

8. Right-click the name of the library, and then select **Substitutions** from the shortcut menu. The **Substitutions** dialog box opens.

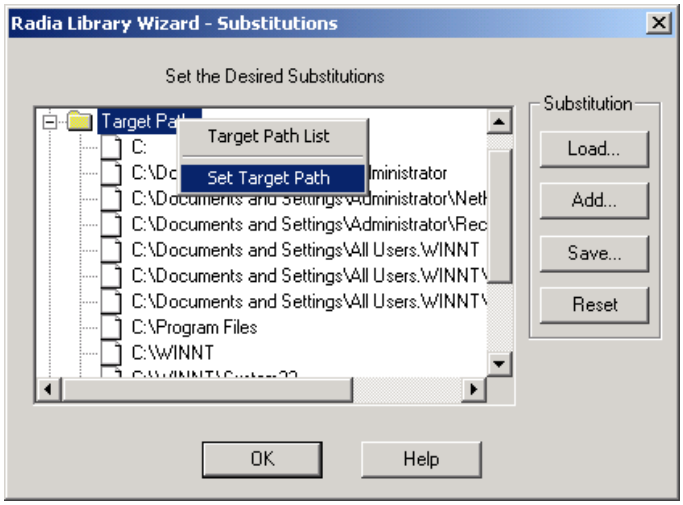

Figure 5.24 ~ Radia Library Wizard – Substitutions.

9. Right-click **Target Paths**, and then from the shortcut menu select **Set Target Path**.
The **Set Target Path** dialog box opens.

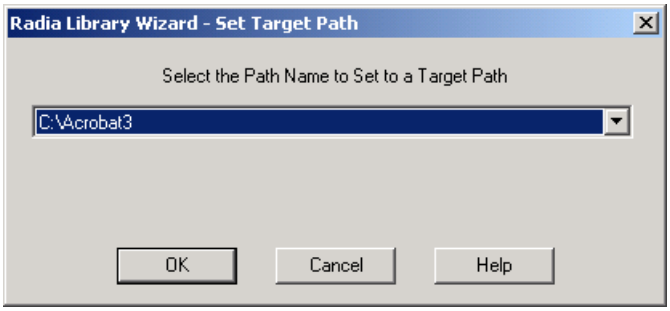

Figure 5.25 ~ Radia Library Wizard – Set Target Path.

10. Select the path that you want to modify.

11. Click **OK**.
The **Edit Target Path** dialog box opens.

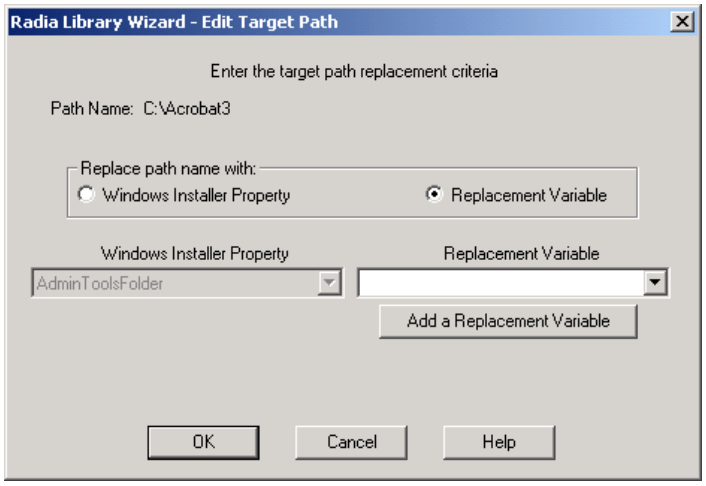

Figure 5.26 ~ Radia Library Wizard – Edit Target Path.

For this example, you will replace the target path with a replacement variable.

12. In the **Replace path name with** area, select **Replacement Variable**.

13. Click **Add a Replacement Variable** to create a replacement variable.

The **Add Replacement Variable** dialog box opens.

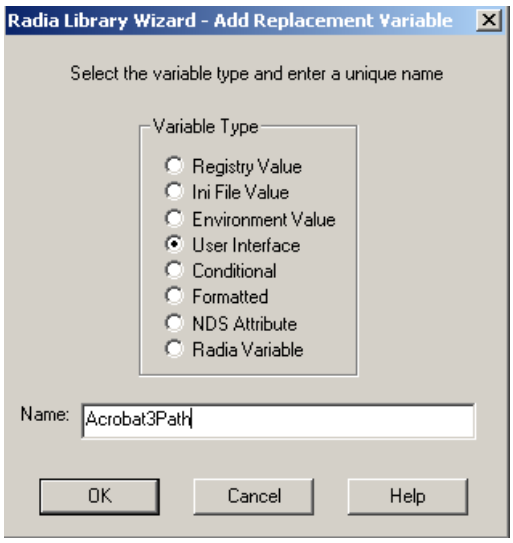

Figure 5.27 ~ Radia Library Wizard – Add Replacement Variable.

14. In the **Variable Type** area, select **User Interface**.

15. In the **Name** text box, type the name for your path.

16. Click **OK**. A message box prompts you that the variable name must be in all uppercase letters. This adjustment will be made automatically for you.

The **Edit User Interface Variable** dialog box opens.

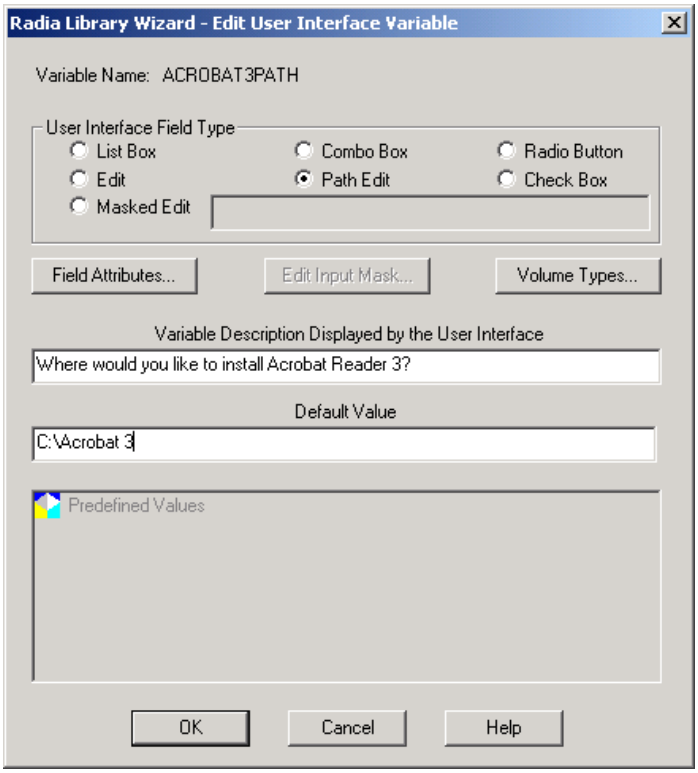

Figure 5.28 ~ Radia Library Wizard – Edit User Interface Variable.

17. In the **User Interface Field Type** area, select **Path Edit**. This allows the user to browse for an installation location for the application.

18. In the **Variable Description Displayed by the User Interface** text box, type instructions to the user.

19. In the **Default Value** text box, type a default value.

20. Click **OK** to return to the **Edit Target Path** dialog box.

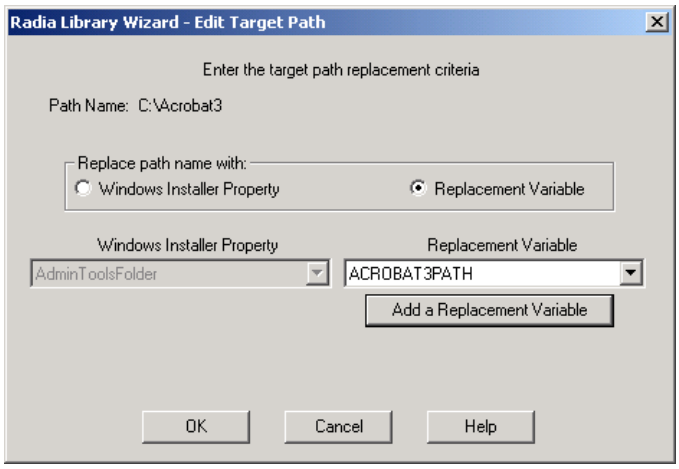

Figure 5.29 ~ Radia Library Wizard – Edit Target Path.

21. Verify that the path you indicated earlier is set as the Replacement Variable.

22. Click **OK** to return to the **Substitutions** dialog box.

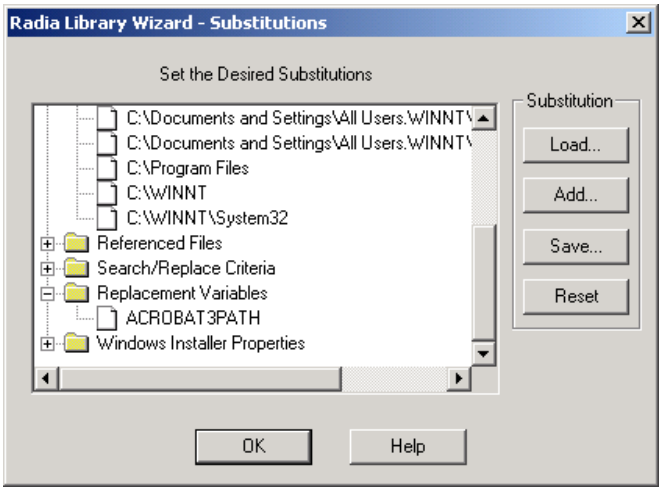

Figure 5.30 ~ Radia Library Wizard – Substitutions.

23. Click **OK**.

24. Click **Next**.

The **Library modification is complete** dialog box opens.

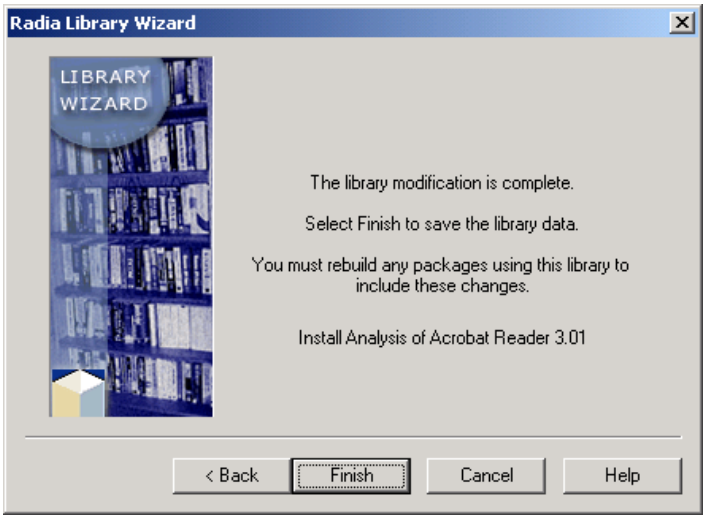

Figure 5.31 ~ Radia Library Wizard – The library modification is complete.

25. Click **Finish**.

After modifying the library, rebuild your package with the Radia Package Wizard. See *Rebuilding a Package* on page 167 for more information on rebuilding a package.

Modifying the Installation Filter

Your Acrobat Reader 3.01 package contains files that are unnecessary after repackaging the application into Windows Installer format. You no longer need the uninstall files, because the ability to remove an application is built right into Windows Installer. In this exercise, you will learn to modify a library filter by removing the uninstall files installed by the original Acrobat Reader 3.01 installation.

Note

Only files that are referenced in the filter can be included or excluded from the filter.

To modify the installation filter

1. Start the Radia Packager for Windows Installer and select the Custom option, **Refine selected components**.

OR

From the Radia Packager for Windows Installer Program Group, start the **Radia Library Wizard.**

The **Select an Action** dialog box opens.

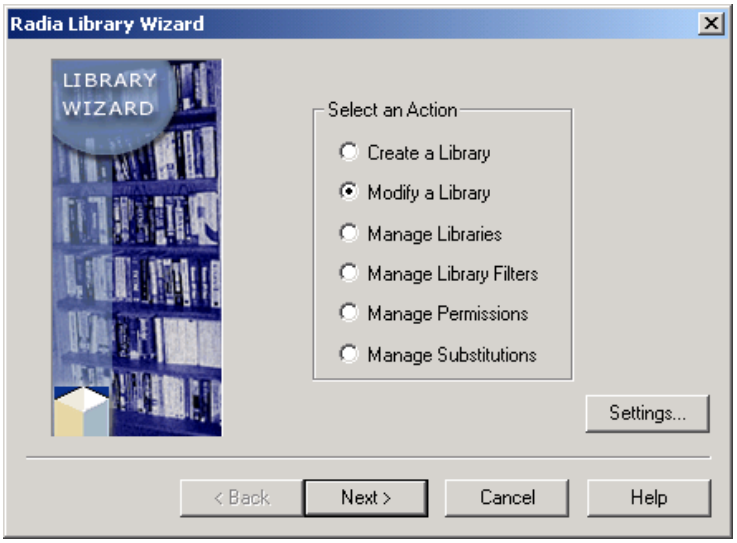

Figure 5.32 ~ Radia Library Wizard – Select an Action.

2. Select **Modify a Library**.
3. Click **Next**.

   The **Select the Type of Modification** dialog box opens.

Figure 5.33 ~ Radia Library Wizard – Select the Type of Modification.

4. Select **Modify the Installation Filter**.
5. Click **Next**.

   The **Select the Library to Modify** dialog box opens.

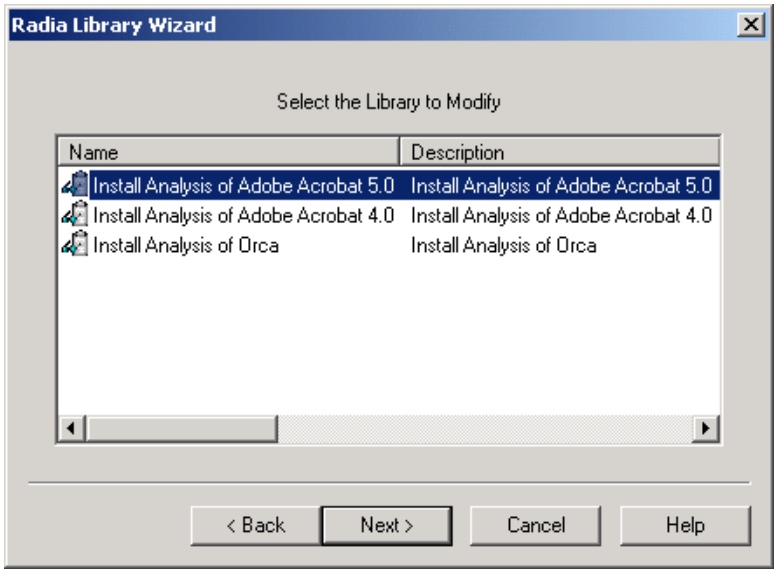

Figure 5.34 ~ Radia Library Wizard – Select the Library to Modify.

6. Select the library that you want to modify.
7. Click **Next**.

   The **Select Paths, Files and Registry Entries to Include** dialog box opens.

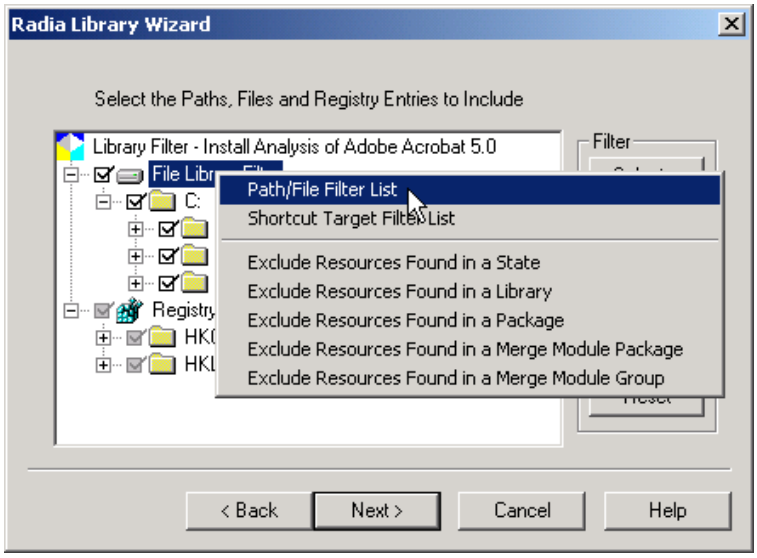

Figure 5.35 ~ Radia Library Wizard – Select the Paths, Files and Registry Entries to Include.

Rather than expanding the information, you can also view the information in the filtering dialog box in a list format.

8. Right-click **File Library Filter**.

9. From the shortcut menu, select **Path/File Filter List**.

The **Path/File Filter List** window opens.

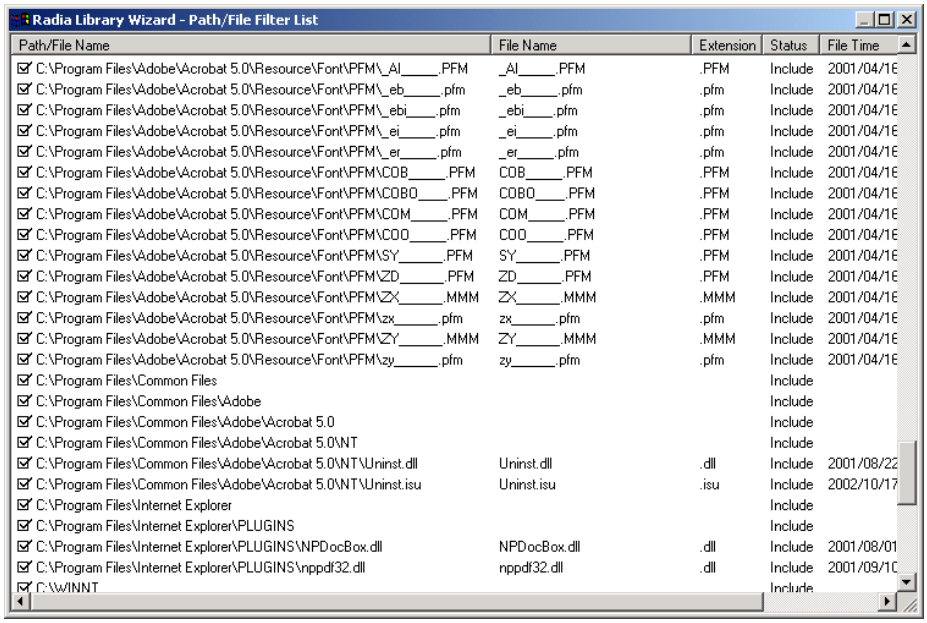

Figure 5.36 ~ Radia Library Wizard – Path/File Filter List window.

10. In the **Path/File Filter List** dialog box, clear the check marks for **C:\WINNT\UNINST.EXE** and for the shortcut **C:\Documents and Settings\All Users\Start Menu\Programs\Adobe Acrobat\Uninstall Acrobat Reader 5.0.lnk**. While you are there, you may also remove the shortcut to the ReadMe file if you don't want a shortcut to it in the Start Menu.

11. Click **Close** to return to the **Select Paths, Files, and Registry Entries to Include** dialog box.

12. Click **Next**.

The Radia Library Wizard performs a quick analysis of the files in your filter.

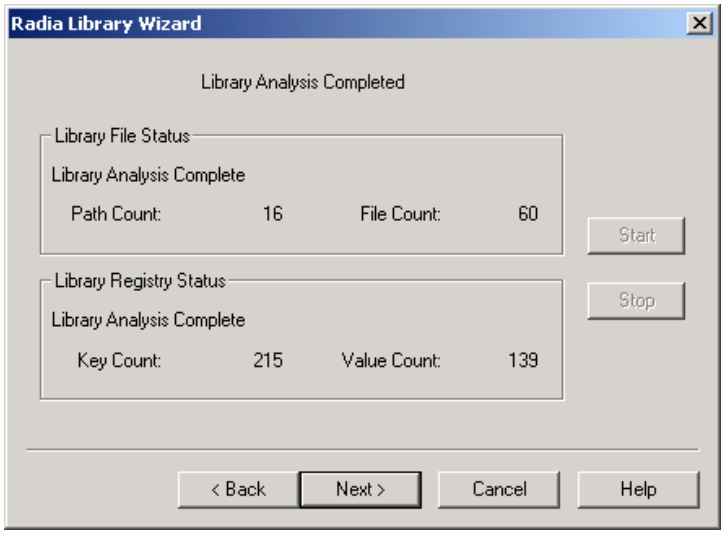

Figure 5.37 ~ Radia Library Wizard – Library Analysis Completed.

13. Click **Next**.

The **Modify Library Installation Settings** dialog box opens. Use this dialog box to modify the installation settings of the folders, files, and registry keys in your library.

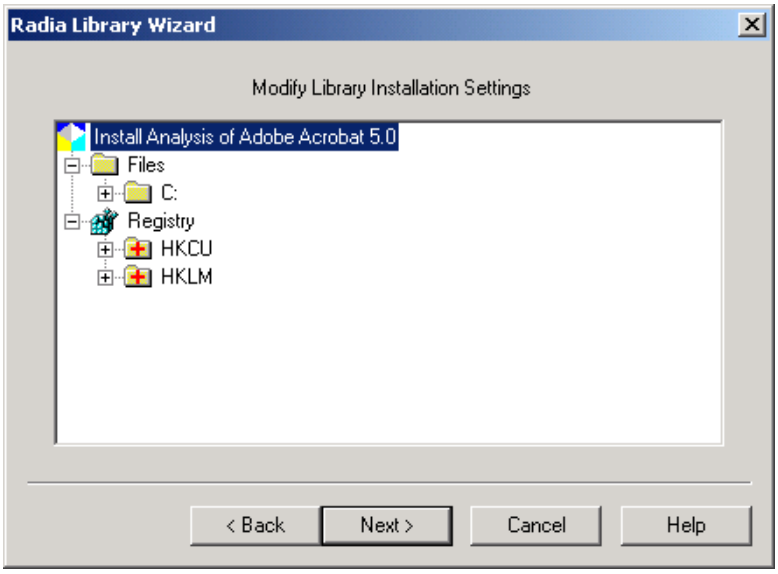

Figure 5.38 ~ Radia Library Wizard – Modify Library Installation Settings.

14. Click **Next**.

The Radia Library Wizard copies any files that need to be recopied.

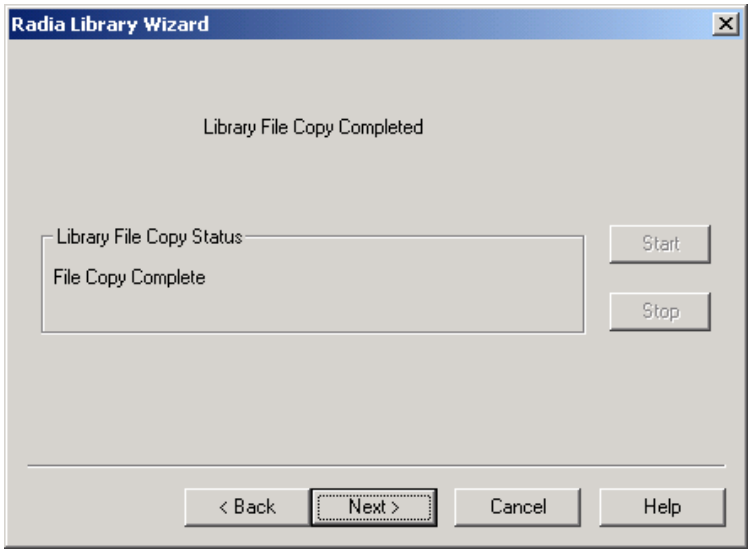

Figure 5.39 ~ Radia Library Wizard – Library File Copy Completed.

15. Click **Next**.

The **library modification is complete** dialog box opens.

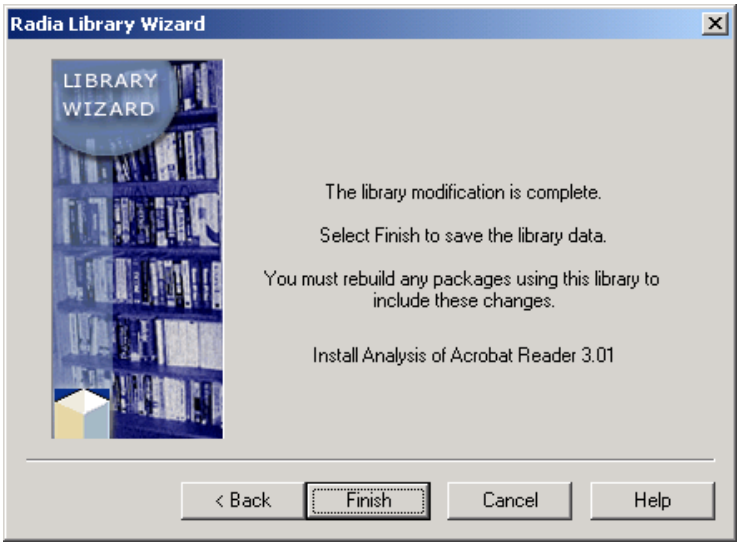

Figure 5.40 ~ Radia Library Wizard – The library modification is complete.

16. Click **Finish**.

Compressing a Library

The Radia Library Wizard allows you to compress a library for archiving or shipping purposes.

To compress a library

> **Note**
>
> The Radia Extensions for Windows Installer compresses libraries into .cab format. The resulting file is a .LIBCAB file.

1. Start the Radia Packager for Windows Installer and select the Custom option, **Refine selected components.**

OR

From the Radia Packager for Windows Installer Program Group, start the **Radia Library Wizard.**

The **Select an Action** dialog box opens.

Figure 5.41 ~ Radia Library Wizard – Select an Action.

2. Select **Manage Libraries**.
3. Click **Next**.

   The **Select a Library Management Action** dialog box opens.

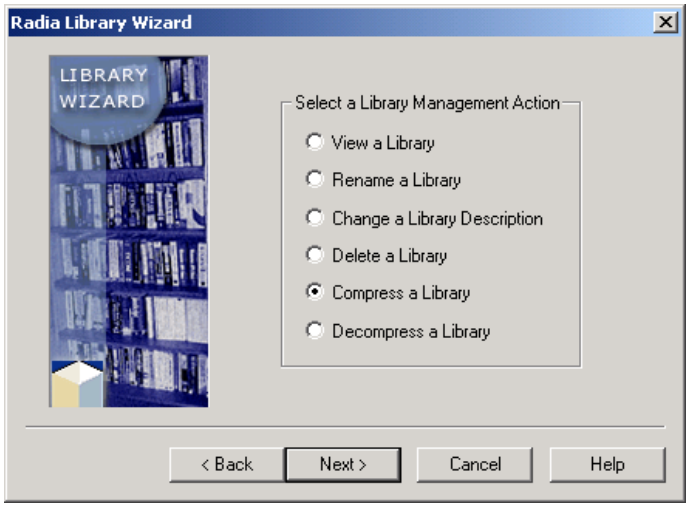

Figure 5.42 ~ Radia Library Wizard – Select a Library Management Action.

4. Select **Compress a Library**.
5. Click **Next**.

   The **Select the Library to be Managed** dialog box opens.

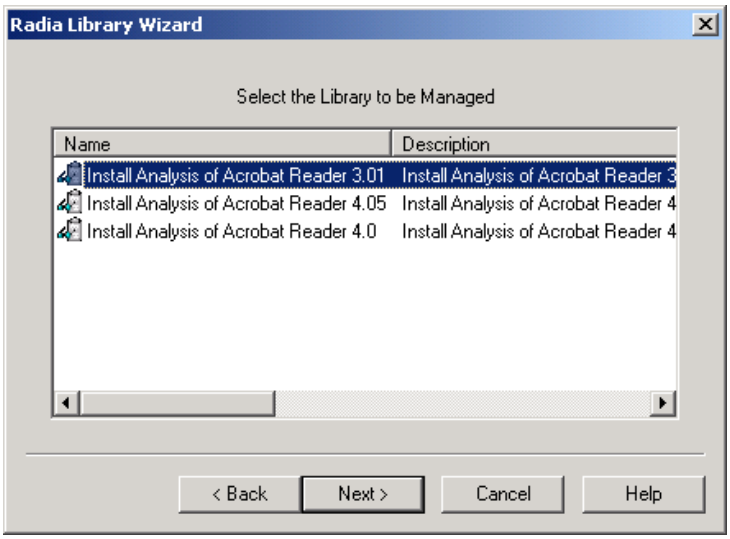

Figure 5.43 ~ Radia Library Wizard – Select the Library to be Managed.

6. Select the library that you want to compress. For example, now that you have updated packages available for Acrobat Reader, you may want to archive the first library you created.
7. Click **Next**.

   The **Library Compression Completed** dialog box opens.

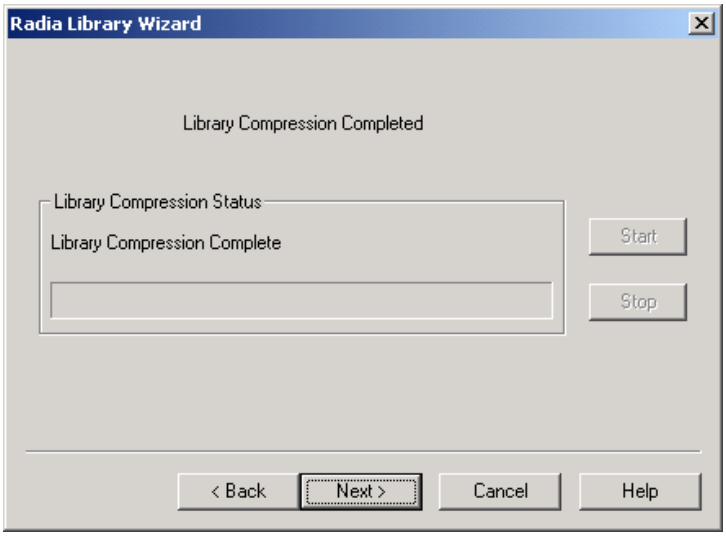

Figure 5.44 ~ Radia Library Wizard – Library Compression Completed.

8. Click **Next**.

The **Select Finish to complete compressing the selected library** dialog box opens.

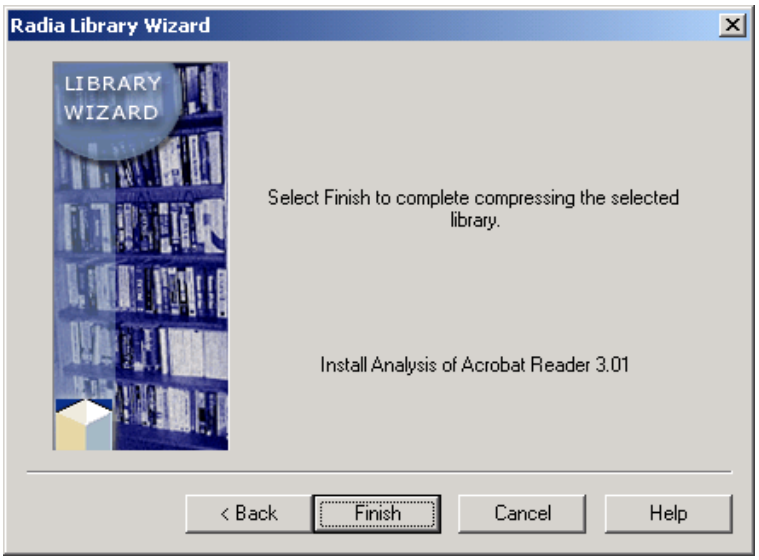

Figure 5.45 ~ Radia Library Wizard – Select Finish to complete compressing the selected library.

9. Click **Finish**.

The library is now compressed into a single LIBCAB (.cab format) file in the **Library** directory.

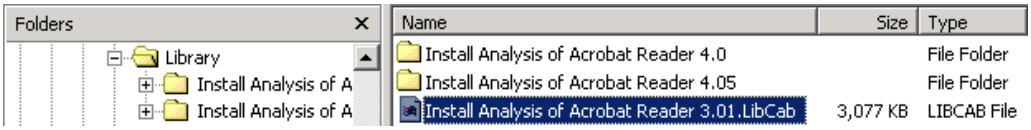

Figure 5.46 ~ Radia Library Wizard – LibCab file in the Library directory.

Decompressing a Library

To decompress a library

1. Perform the same steps as you did in *Compressing a Library* on page 135, but in the Radia Library Wizard, select **Decompress a Library**.

Figure 5.47 ~ Radia Library Wizard – Select a Library Management Action.

2. A list of compressed libraries opens.

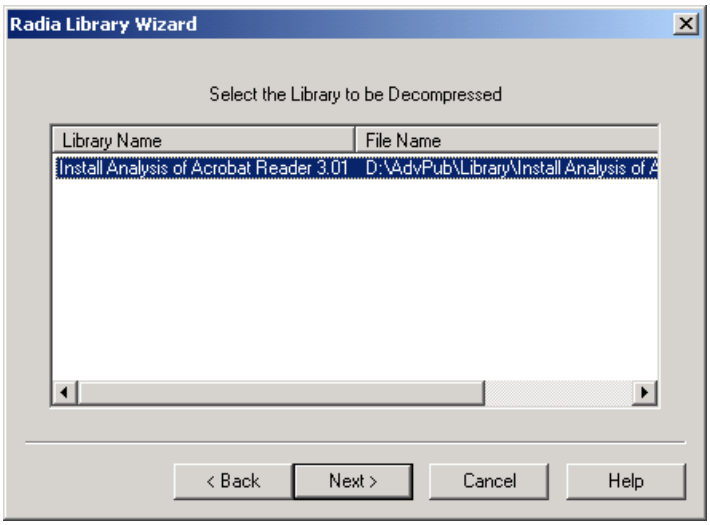

Figure 5.48 ~ Radia Library Wizard – Select the Library to be Decompressed.

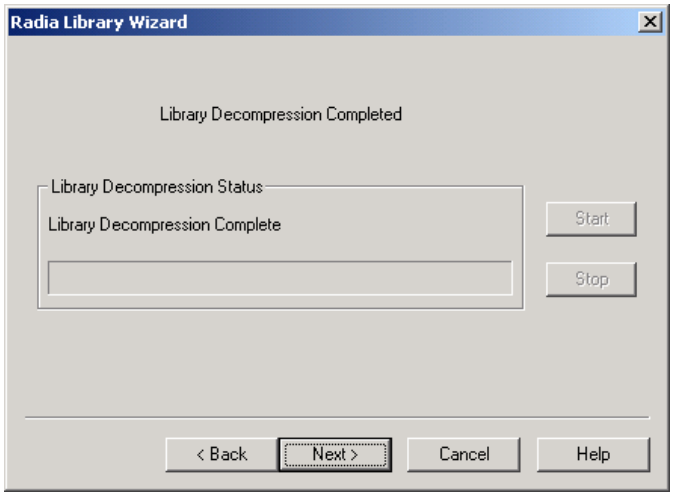

Figure 5.49 ~ Radia Library Wizard – Library Decompression Complete.

3. Click **Next**.

   Library decompression is finished.

4. Click **Finish**.

Creating Permissions

If you are using the NTFS file system on your workstations, you may need to adjust permission settings on certain folders, files, and registry settings within your package. The Radia Library Wizard allows you to set these settings within your library.

In the following example, you will to set up permissions to give everyone in your development group full access, power users to have read and execute access, and users to have only read access.

Note

In order to use advanced permissions in the Radia Library Wizard, Radia Advanced Features for Windows Installer must be installed on the target desktop.

Creating a Lock Permission List

To create a Permission Access Control Group

1. Start the Radia Packager for Windows Installer and select **Refine selected components.**

   OR

   From the Radia Packager for Windows Installer Program Group, start the **Radia Library Wizard.**

2. From the Radia Library Wizard, select **Manage Permissions**.

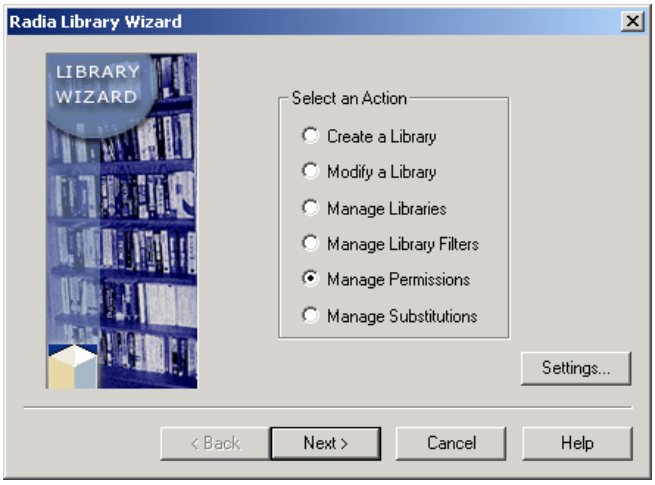

Figure 5.50 ~ Radia Library Wizard – Select an Action.

3. Click **Next**.
4. The **Select a Permission Management Action** dialog box opens.

Figure 5.51 ~ Radia Library Wizard – Select a Permission Management Action.

5. Select **Create a Permission**.
6. Click **Next**.

   The **Set Desired Permissions** dialog box opens. Use this dialog box to add the Development group to the list of groups and users.

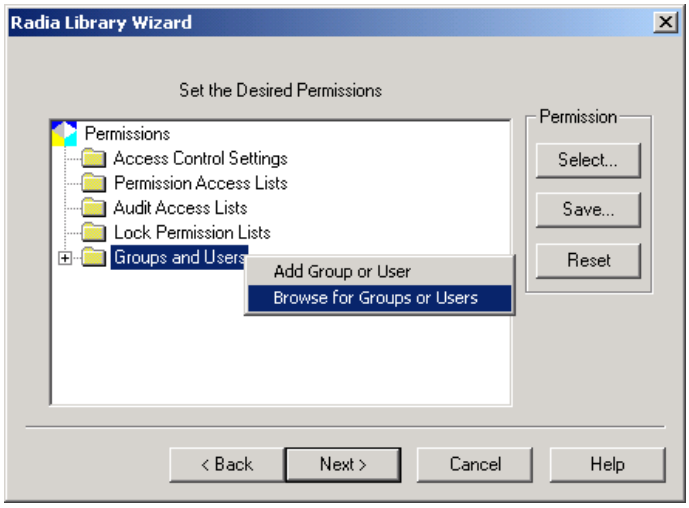

Figure 5.52 ~ Radia Library Wizard – Set Desired Permissions.

7. Right-click **Groups and Users** and select **Browse for Groups or Users**.
The **Browse Users** dialog box opens.

Figure 5.53 ~ Radia Library Wizard – Browse Users.

8. In the **Server** text box, type the name of the machine from which you would like to browse the groups and users.

9. In the **Account Type Selection** area, select **Groups and Users**.

The **Select the Users to Add to the Permission** dialog box opens.

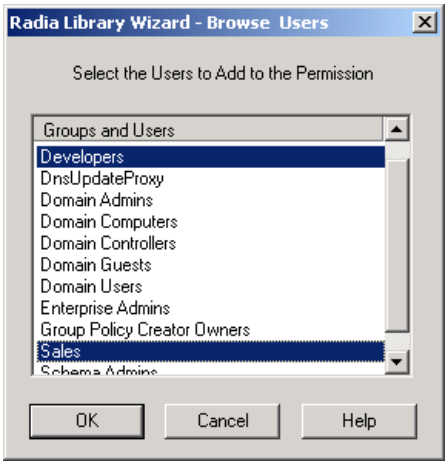

Figure 5.54 ~ Radia Library Wizard – Browse Users.

10. Select the groups and users you would like to add.

- To select **multiple groups and users**, hold down the **Ctrl** key on your keyboard and select the appropriate groups and users.
- To select a **range of groups and users**, hold down the **Shift** key on your keyboard.

11. Click **OK**.

The **Set the Desired Permissions** dialog box opens. The groups and users that you selected have been added to your list of available Groups and Users.

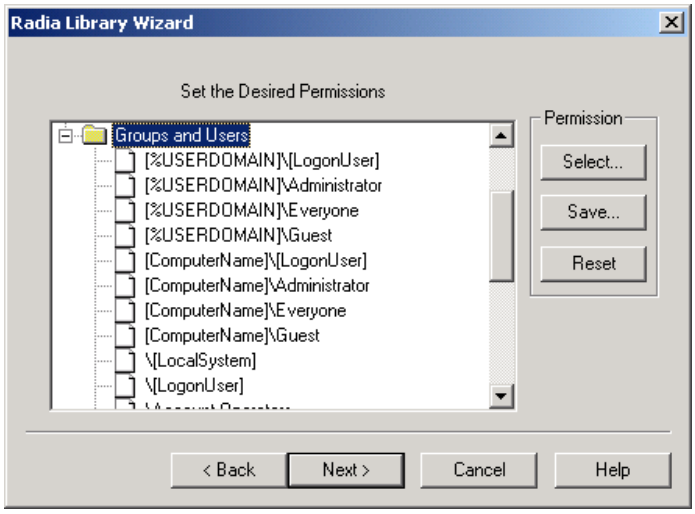

Figure 5.55 ~ Radia Library Wizard – Set Desired Permissions.

Next you need to create a **Lock Permission List**. When you apply a Lock Permission List to a folder, file, or registry key, all permissions associated with each group in the list are applied to that resource.

12. Scroll to the top of the **Set Desired Permissions** dialog box, and right-click **Lock Permission Lists** (see below).

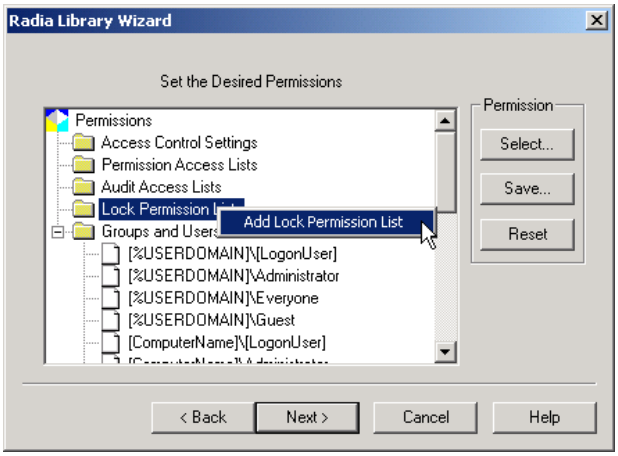

Figure 5.56 ~ Radia Library Wizard – Set Desired Permissions.

13. From the shortcut menu, select **Add Lock Permission List** from the shortcut menu.
The **Add Lock Permission List** dialog box opens.

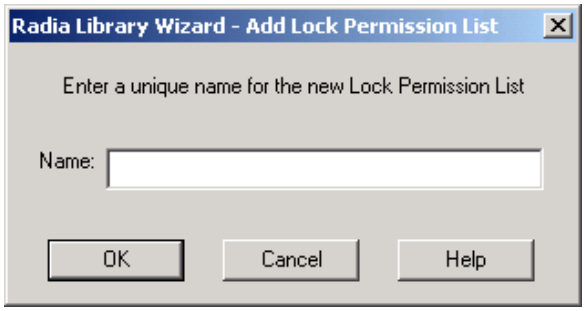

Figure 5.57 ~ Radia Library Wizard – Add Lock Permission List.

14. In the **Name** text box, type a unique name for the new Lock Permission List.

15. Click **OK**.

16. In the **Set the Desired Permissions** dialog box, right-click the name of the newly created Lock Permission List.

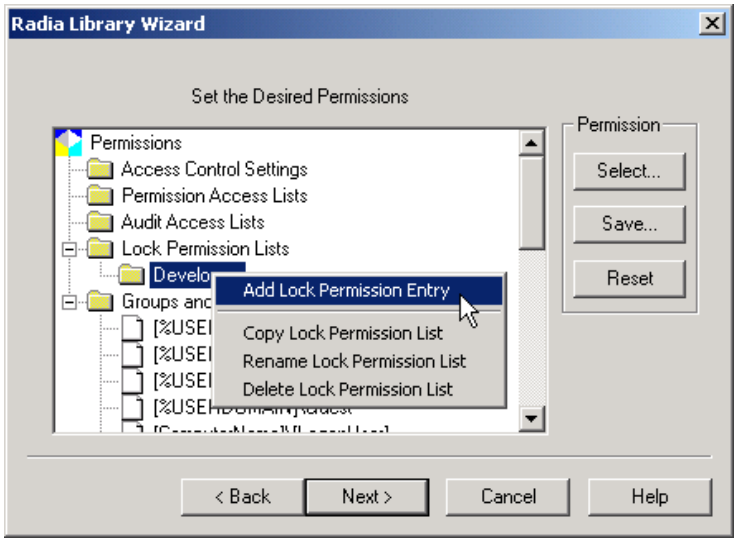

Figure 5.58 ~ Radia Library Wizard – Set Desired Permissions.

17. Select **Add Lock Permission Entry**.

The **Add Lock Permission** dialog box opens.

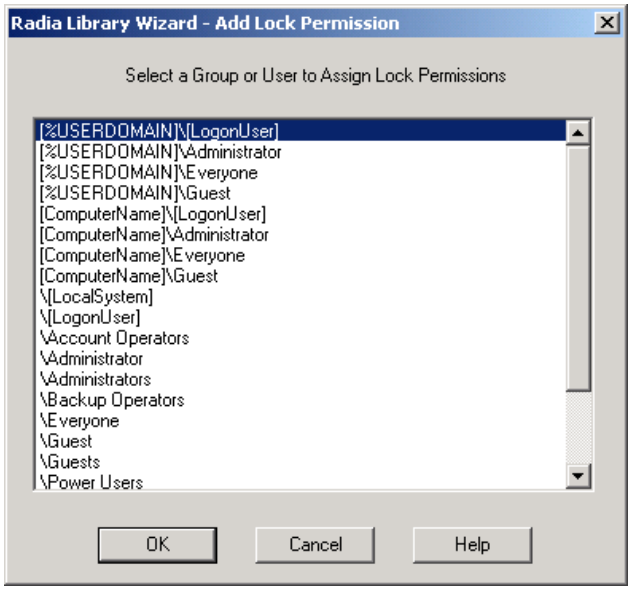

Figure 5.59 ~ Radia Library Wizard – Add Lock Permission.

18. From the list, select a user or group to assign Lock Permissions.

19. Click **OK**.

The **Lock Permissions** dialog box opens.

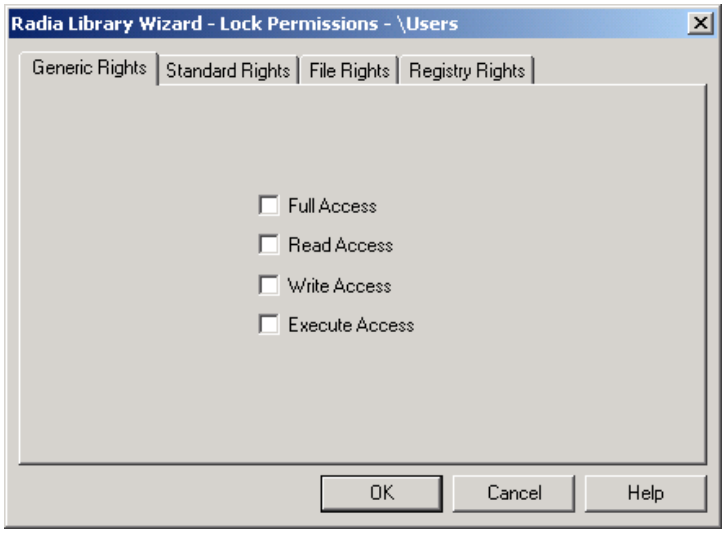

Figure 5.60 ~ Radia Library Wizard – Lock Permissions – Generic Rights tab.

20. Select **Full Access**.

Click **OK** to return to the **Set Desired Permissions** dialog box.

If you would like to add Lock Permissions for multiple groups or users, repeat the previous steps.

Note

For ease of management, it is always better to assign rights to groups rather than to individuals. For more information on permissions, look on the Microsoft Web site or refer to a book on Windows NT or Windows 2000.

21. Click **Next**.

The **Set the Description for the Permission** dialog box opens.

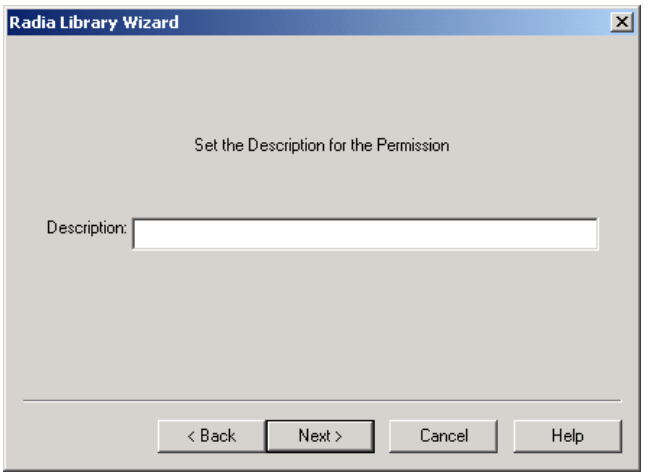

Figure 5.61 ~ Radia Library Wizard – Set the Description for the Permission.

22. In the **Description** text box, type a description for the permission file.

23. Click **Next**.

The final dialog box opens.

Figure 5.62 ~ Radia Library Wizard – Select Finish to save the selected permission.

24. Click **Finish** to complete the creation of your Permission File.

Using Substitutions

Substitutions allow you to have more flexibility and control over the installation of an application. With substitutions you can allow an application to dynamically add or replace information in a package, give a user the ability to enter information, or even add information to a certain location in a text or INI file. You saw one use of a substitution earlier when changing the installation target path for Acrobat Reader 3.01. In this example, you will personalize the Notice to User message included with Acrobat Reader.

Creating a Substitution

To create a substitution within a package

1. Start the Radia Packager for Windows Installer and select the custom option **Refine selected components.**

   OR

   From the Radia Packager for Windows Installer Program Group, start the Radia **Library Wizard.**

2. Select **Manage Substitutions** from the Radia Library Wizard.

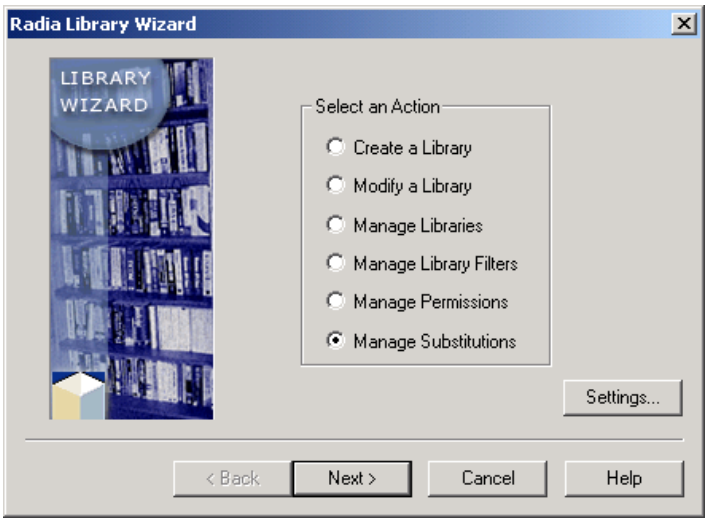

Figure 5.63 ~ Radia Library Wizard – Select an Action.

3. Click **Next**.

   The **Select a Substitution Management Action** dialog box opens.

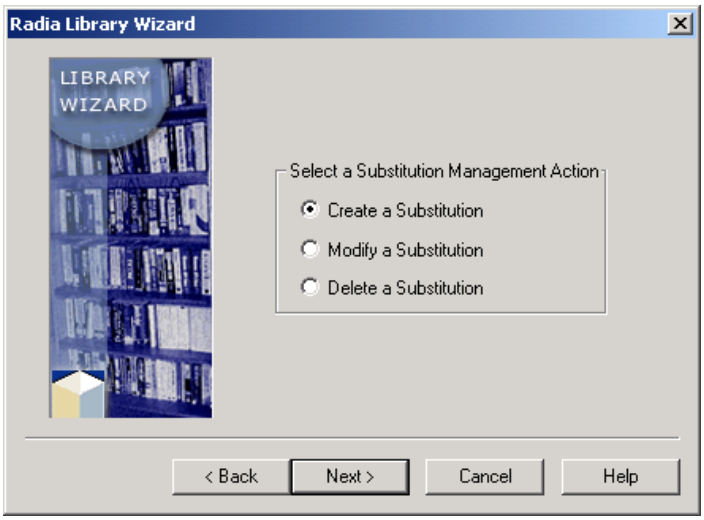

Figure 5.64 ~ Radia Library Wizard – Select a Substitution Management Action.

4. Select **Create a Substitution**.
5. Click **Next**.

   The **Set the Desired Substitutions** dialog box opens. The first step is to determine what you would like to replace, and set up a search to locate this information.

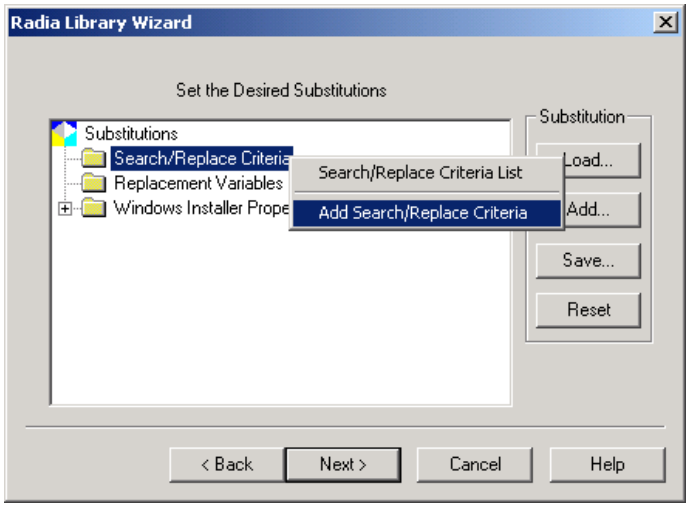

Figure 5.65 ~ Radia Library Wizard – Substitutions.

6. Right-click **Search/Replace Criteria** and select **Add Search/Replace Criteria**.
   The **Add Search String** dialog box opens.

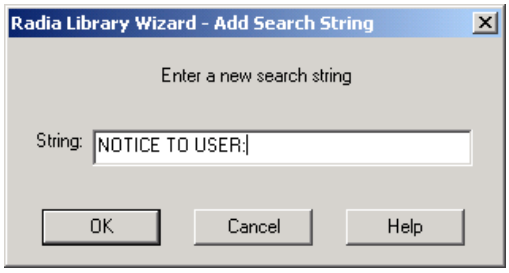

Figure 5.66 ~ Radia Library Wizard – Add Search String.

7. In the **String** text box, type string for which you would like to search such as **NOTICE TO USER**.
8. Click **OK**.
   The **Edit Search/Replace Criteria** dialog box opens. Use this dialog box to create a replacement variable to take the place of the string **NOTICE TO USER**.

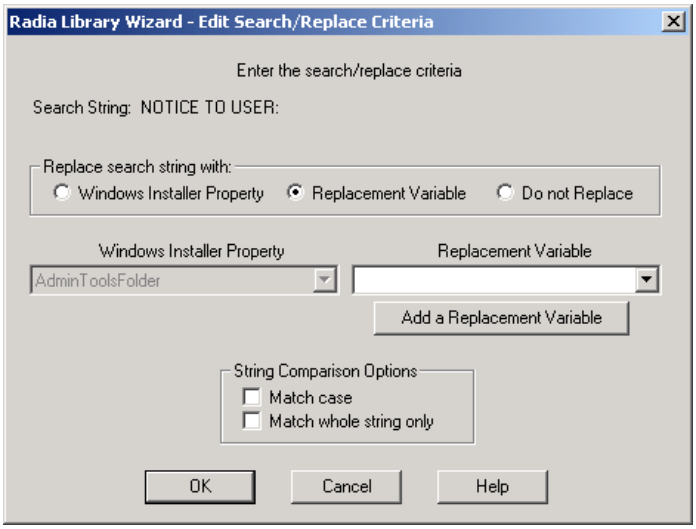

Figure 5.67 ~ Radia Library Wizard – Edit the Search/Replace Criteria.

9. In the **Replace search string with** area, select **Replacement Variable**.

10. Click **Add a Replacement Variable**.

The **Add Replacement Variable** dialog box opens.

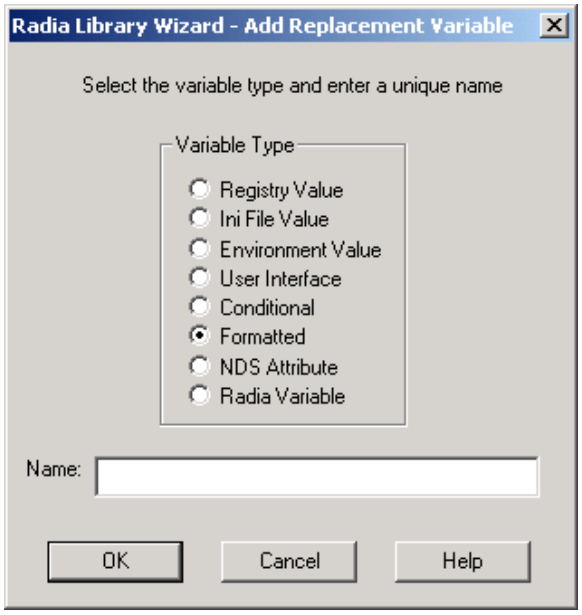

Figure 5.68 ~ Radia Library Wizard – Add Replacement Variable.

11. In the **Variable Type** area, select **Formatted**.

12. In the **Name** text box, type **Notice_To_User**.

13. Click **OK**.

The **Edit Formatted Variable** dialog box opens. You can use this dialog box to personalize the NOTICE TO USER message. To personalize this message, you will combine the text with some dynamic information, such as the identification of the user who is logged on.

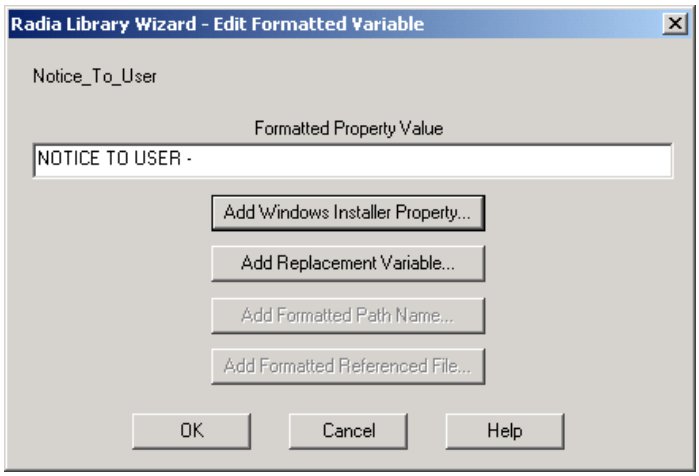

Figure 5.69 ~ Radia Library Wizard – Edit Formatted Variable.

14. In the **Formatted Property Value** text box, type **NOTICE TO USER** –.

15. Click **Add Windows Installer Property.**

The **Add Formatted Property** dialog box opens.

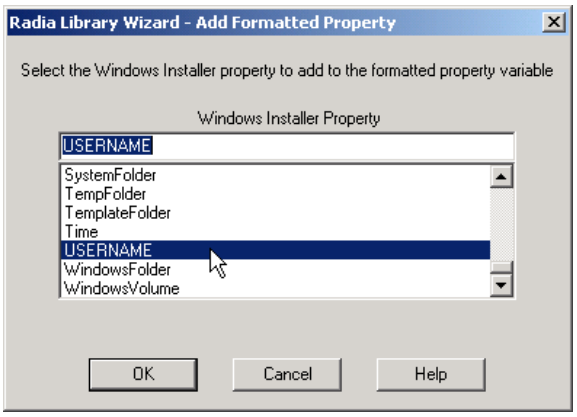

Figure 5.70 ~ Radia Library Wizard – Add Formatted Property.

16. In the **Windows Installer Properties** list, select **USERNAME**.

17. Click **OK** to return to the **Edit Formatted Variable** dialog box.

18. Type a colon (:) after **[USERNAME]**.

19. Click **OK**.

20. Click **OK** again to return to the **Set the Desired Substitutions** dialog box. Your search/replace criteria and replacement variable are now available for use as substitutions.

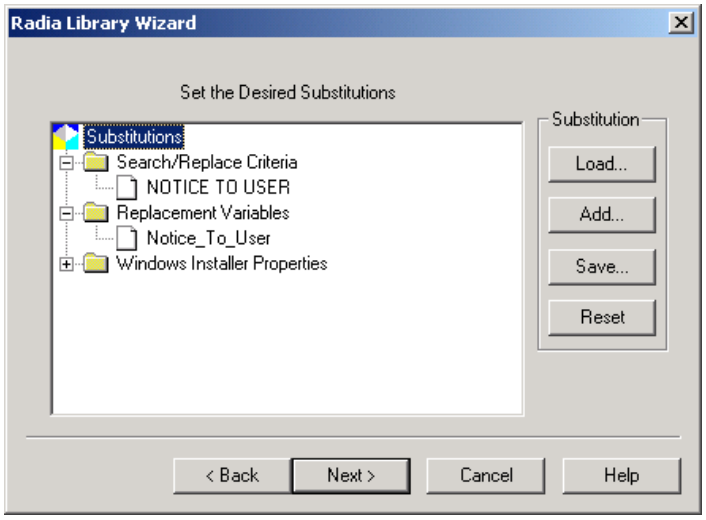

Figure 5.71 ~ Radia Library Wizard – Substitutions.

21. Click **Next**.

22. The **Set the Description for the Substitution** dialog box opens.

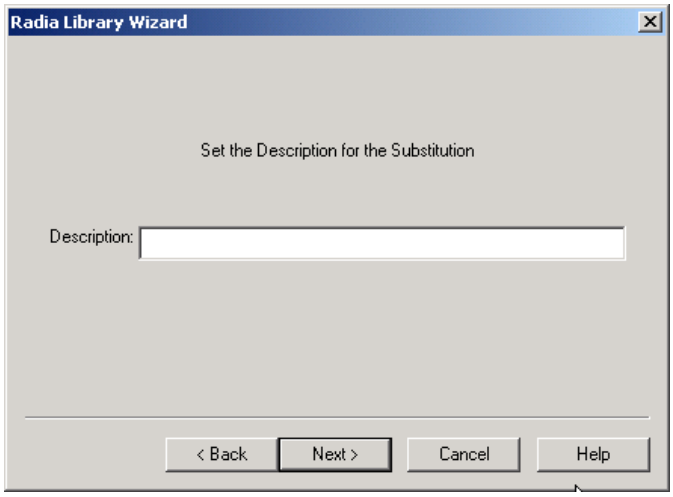

Figure 5.72 ~ Radia Library Wizard – Set Description for Substitution.

23. In the **Description** text box, type a description of the substitution.

24. Click **Next**.

The final dialog box opens.

Figure 5.73 ~ Radia Library Wizard – Save the selected substitution.

25. Click **Finish** to complete your substitution.

Applying a Substitution to a Text File

Now that you have created your substitution, you need to apply that Substitution to your Acrobat Reader installation.

To apply a substitution within a package

1. Start the Radia Packager for Windows Installer and select the custom option **Refine selected components.**

OR

From the Radia Packager for Windows Installer Program Group, start the Radia **Library Wizard.**

2. Select **Modify a Library** from the Radia Library Wizard.

Figure 5.74 ~ Radia Library Wizard – Select an Action.

3. Click **Next**.

4. The **Select the Type of Modification** dialog box opens.

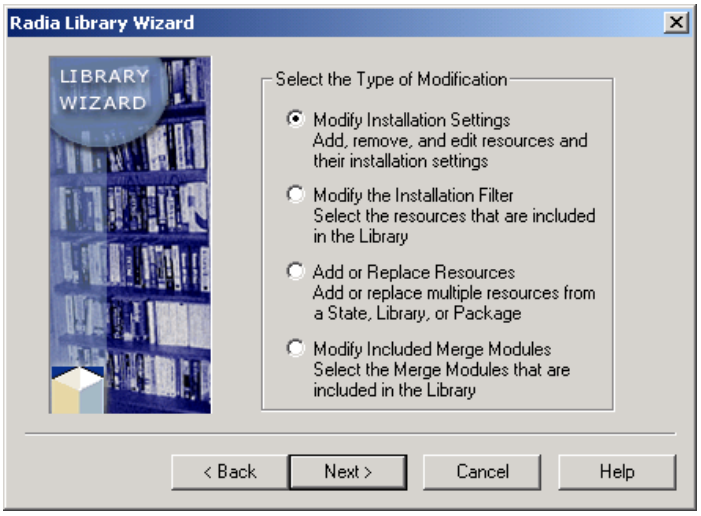

Figure 5.75 ~ Radia Library Wizard – Select the Type of Modification.

5. Select **Modify Installation Settings**.
6. Click **Next**.

   The **Select the Library to Modify** dialog box opens.

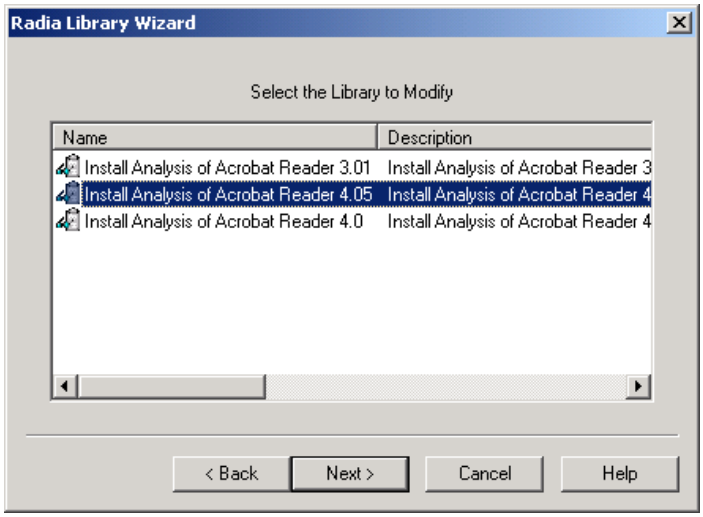

Figure 5.76 ~ Radia Library Wizard – Select the Library to Modify.

7. Select the library to modify.
8. Click **Next**.

   The **Modify Library Installation Settings** dialog box opens.

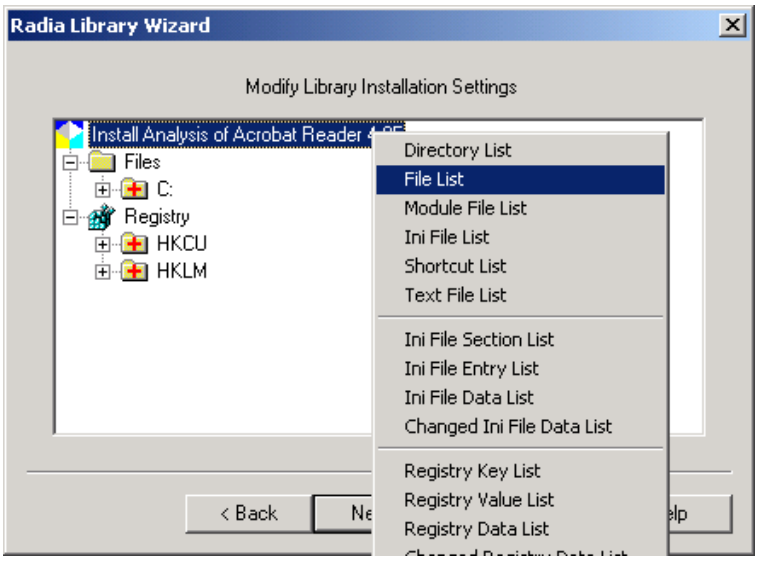

Figure 5.77 ~ Radia Library Wizard – Modify Library Installation Settings.

9. Right-click the name of the library you selected. (For this example, we selected **Install Analysis of Acrobat Reader 4.05**).
10. From the shortcut menu, select **File List**.

    The **File List** dialog box opens. Before you can apply your substitution, you will verify that the file is being installed as a Text File and not a Data File.

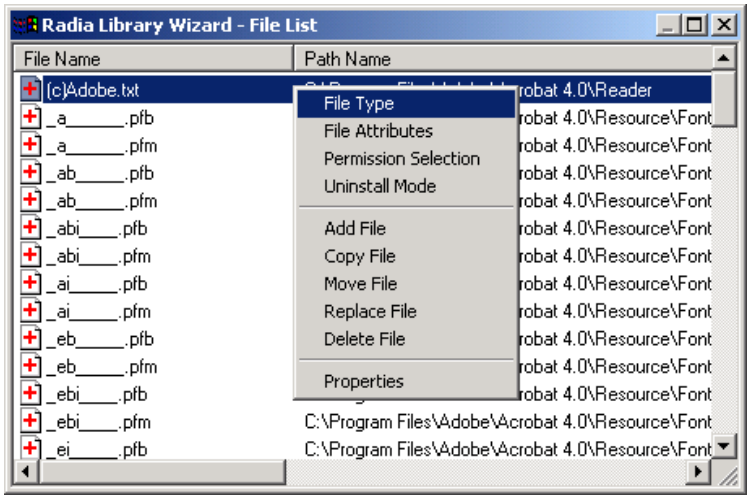

Figure 5.78 ~ Radia Library Wizard – File List.

11. Right-click the text file, for example **(c)Adobe.txt**, and select **File Type** from the shortcut menu.

The **File Type** dialog box opens.

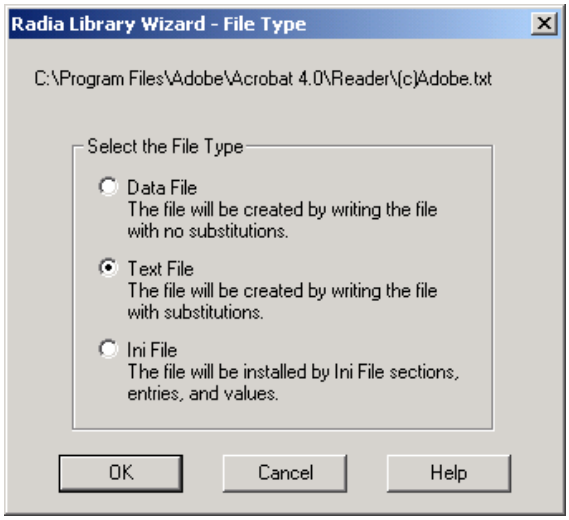

Figure 5.79 ~ Radia Library Wizard – Select the File Type.

12. In the **Select the File Type** area, select **Text File**. This allows you to apply your substitution to the file.

13. Click **OK** to return to the **File List** dialog box.

14. Right-click **(c)Adobe.txt**, and select **Substitution Selection**.

The **Substitution Selection** dialog box opens. The Radia Library Wizard searches the text document for the Search/Replace Criteria set.

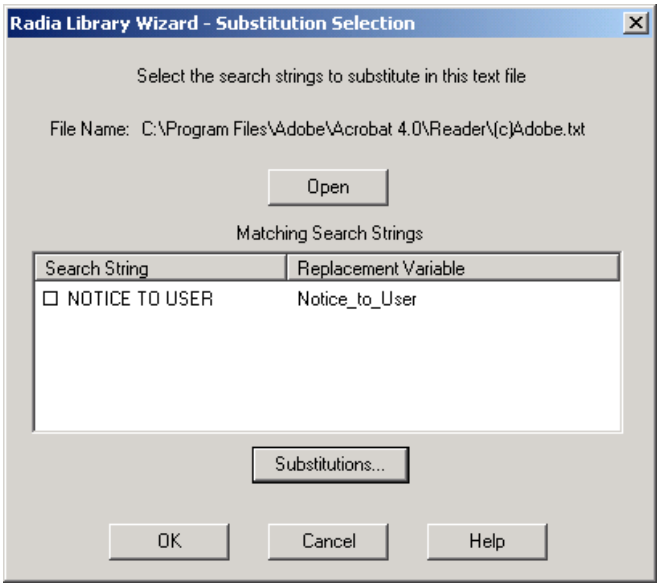

Figure 5.80 ~ Radia Library Wizard – Substitution Selection.

15. Select the check box next to the Search String that you want to replace the text with.

16. Click **OK**.

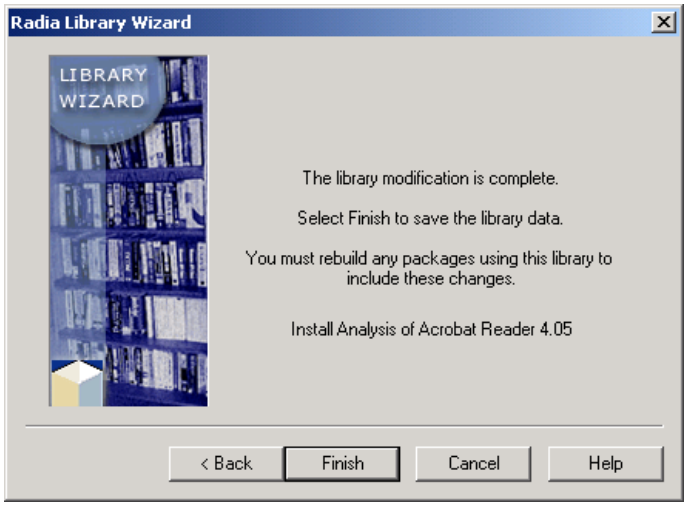

Figure 5.81 ~ Radia Library Wizard – The library modification is complete.

17. Click **Finish**. Now, you must rebuild your package before you can install your package with the substitution. See *Rebuilding a Package* on page 167 for more information.

After rebuilding and installing your package, you will see the changes applied to the text in the text file.

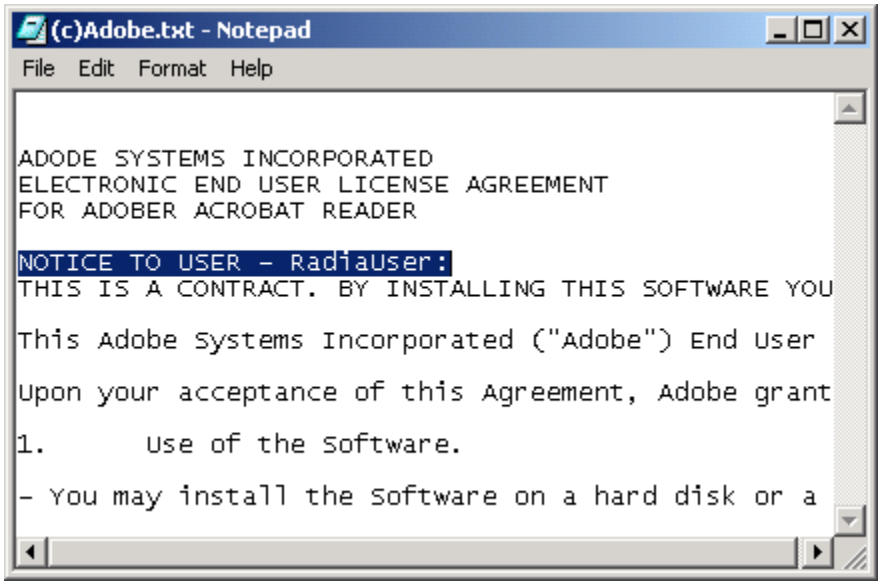

Figure 5.82 ~ Changes reflected in Adobe.txt file.

Using Radia Package Wizard Advanced Functions

The Radia Package Wizard provides you with an easy interface to work with all of the different built-in features of Windows Installer and more. In nearly all cases, the Radia Packager for Windows Installer removes the need for a manual package-editing tool and for learning about the Windows Installer database table structure.

To access these functions from the Radia Packager for Windows Explorer, select the custom option **Package components.**

Rebuilding a Package

In some situations, you may need to rebuild a package. For example, if you make changes to a library after creating a package, you must rebuild the package for the changes to take effect.

To rebuild a package for which the library has been modified

1. Start Radia Packager for Windows Installer and select the custom option **Package components**

   OR

   From the Radia Packager for Windows Installer Program Group, start the **Radia Package Wizard**.

   The **Select an Action** dialog box opens.

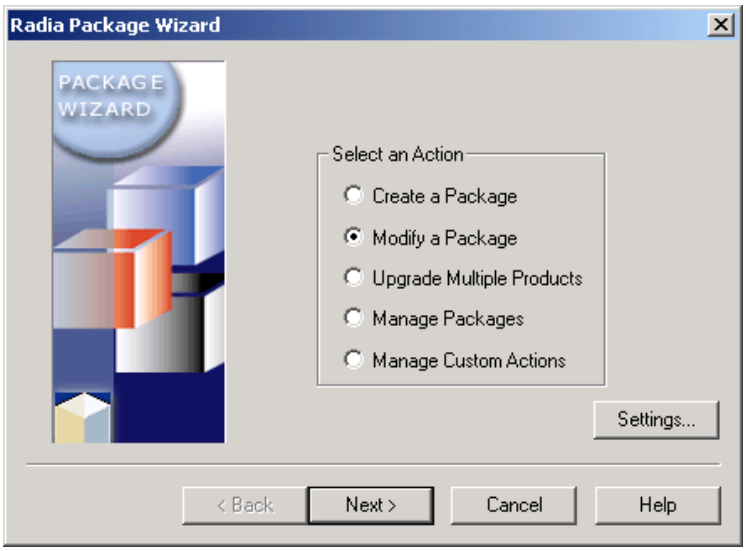

Figure 5.83 ~ Radia Package Wizard – Select an Action.

2. Select **Modify a Package**.
3. Click **Next**.
   The **Package Modification Action** dialog box opens.

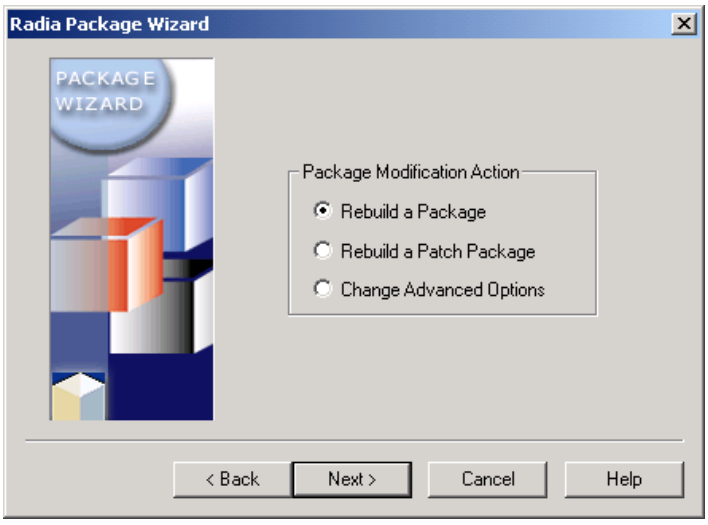

Figure 5.84 ~ Radia Package Wizard – Package Modification Action.

4. Select **Rebuild a Package**.
5. Click **Next**.
   The **Select the Product to Modify** dialog box opens.

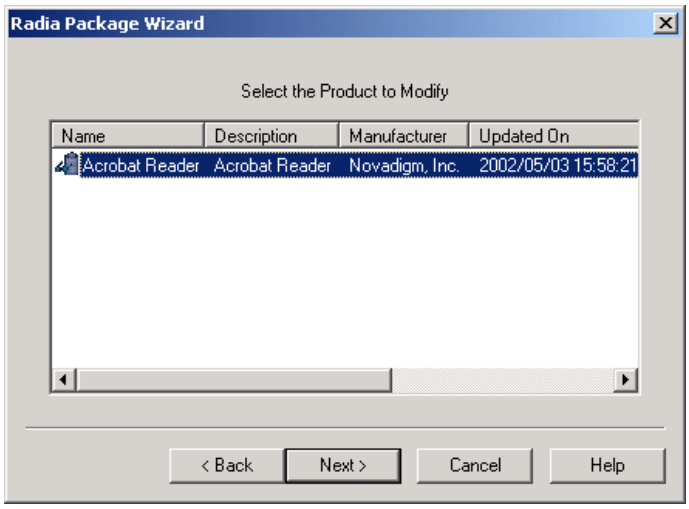

Figure 5.85 ~ Radia Package Wizard – Select the Product to Modify.

6. Select the appropriate product, then click **Next**.
The **Select the Package to Modify** dialog box opens.

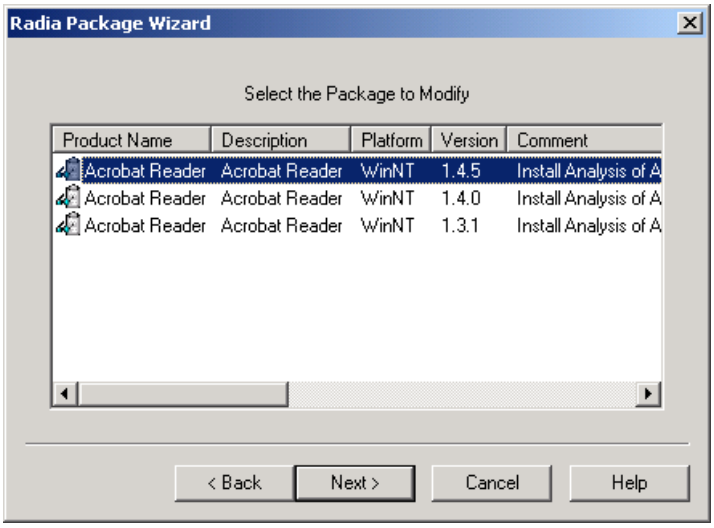

Figure 5.86 ~ Radia Package Wizard – Select the Package to Modify.

7. Select the appropriate package, and then click **Next**.

The **Select the Library from which the Package will be created** dialog box opens.

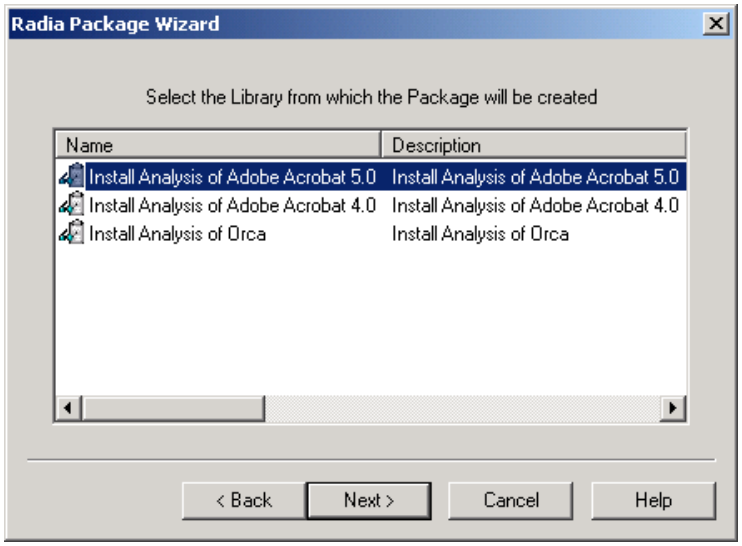

Figure 5.87 ~ Radia Package Wizard – Select the Library from which the Package will be created.

8. Select the name of the library that you modified.

9. Click **Next**.

The **Advanced Options** dialog box opens.

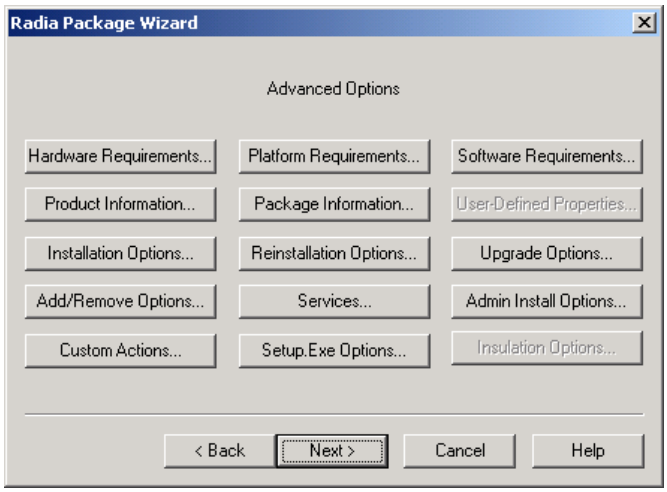

Figure 5.88 ~ Radia Package Wizard – Advanced Options.

10. Make any necessary modifications to the Advanced Options.

11. Click **Next**.

The **Package Analysis Completed** dialog box opens.

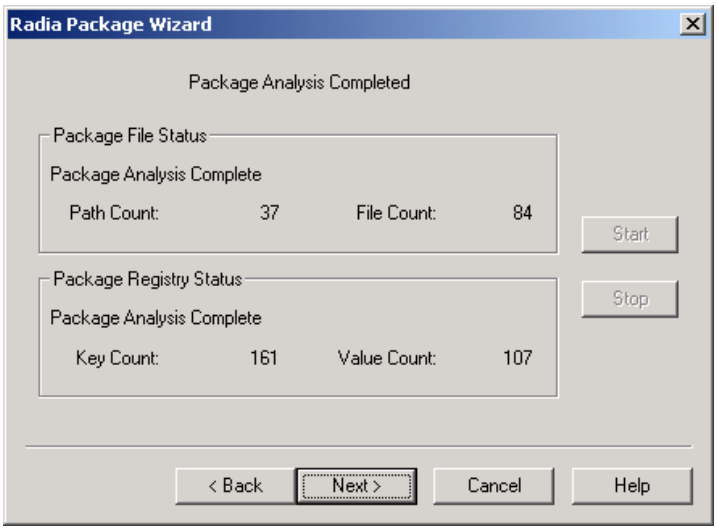

Figure 5.89 ~ Radia Package Wizard – Package Analysis Completed.

12. Click **Next**.

The **Package Results** dialog box opens.

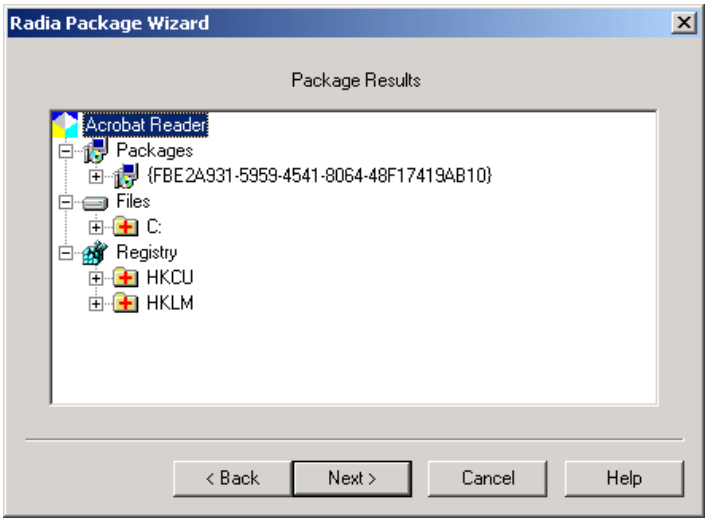

Figure 5.90 ~ Radia Package Wizard – Package Results.

13. Click **Next**.

The **Package File Options** dialog box opens.

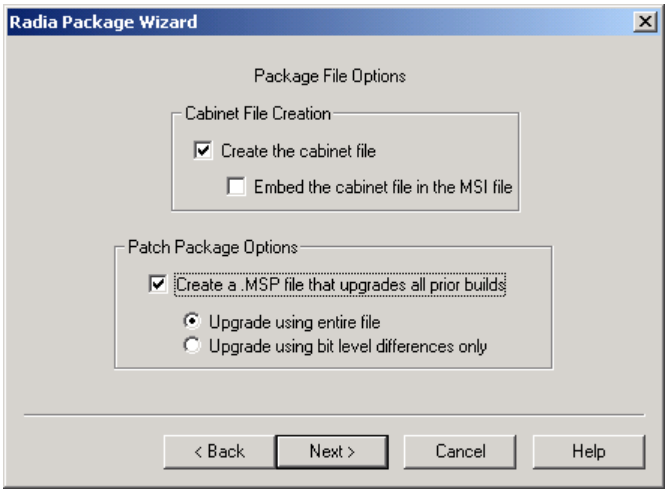

Figure 5.91 ~ Radia Package Wizard – Package File Options.

14. Leave **Create the cabinet file** selected.

15. Select **Create a .MSP file that upgrades all prior builds** and make sure **Upgrade using entire file** is selected.

> **Note**
>
> The option **Create a .MSP file that upgrades all prior builds** is only available when working with updates (new builds) of packages.

16. Click **Next**.

The package is created.

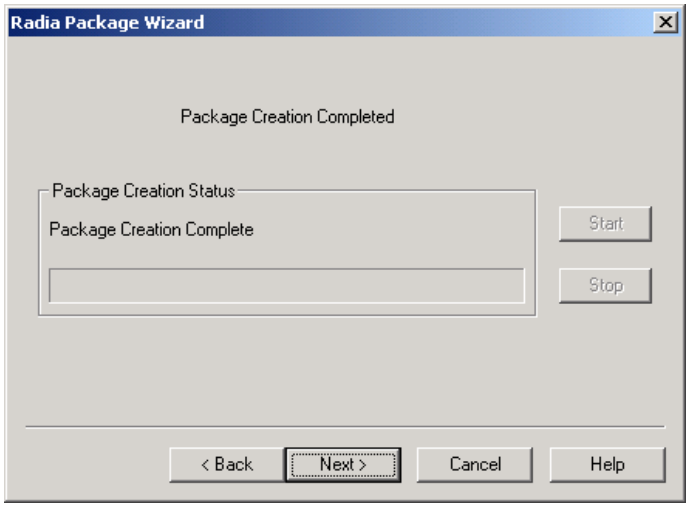

Figure 5.92 ~ Radia Package Wizard – Package Creation Completed.

17. Click **Next**.

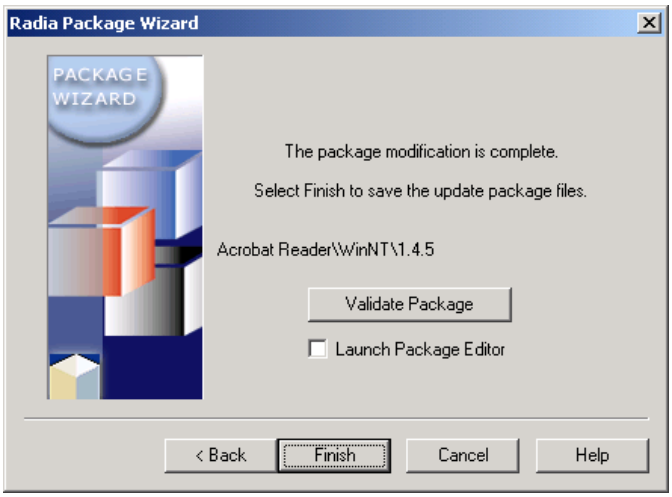

Figure 5.93 ~ Radia Package Wizard – The package modification is complete.

18. Click **Finish**.

Using Advanced Options

The **Radia Package Wizard Advanced Options** dialog box is a portal to all Windows Installer Options available. In this exercise, you will set the Advanced Options to:

- Confirm that WRITE.EXE exists on the target machine before installing the Acrobat Reader Package.
- Install the Acrobat Reader package for all users of the machine.

To set a software requirement

1. Start Radia Packager for Windows Explorer and select the custom option **Package components**.

   OR

   From the Radia Packager for Windows Installer Program Group, start the **Radia Package Wizard**.

   The **Select an Action** dialog box opens.

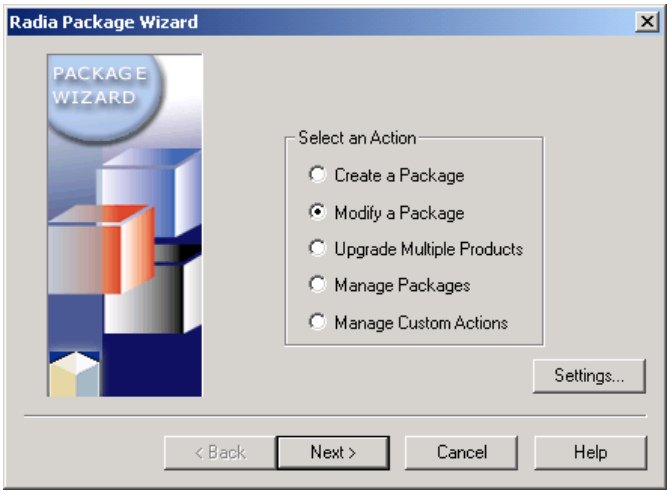

Figure 5.94 ~ Radia Package Wizard – Select an Action.

2. Select **Modify a Package**.
3. Click **Next**.

   The **Package Modification Action** dialog box opens.

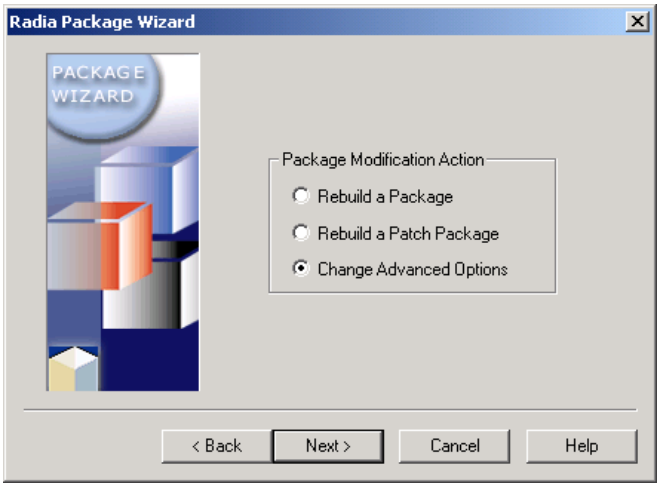

Figure 5.95 ~ Radia Package Wizard – Package Modification Action.

4. Select **Change Advanced Options**.
5. Click **Next**.

   The **Select the Product to Modify** dialog box opens.

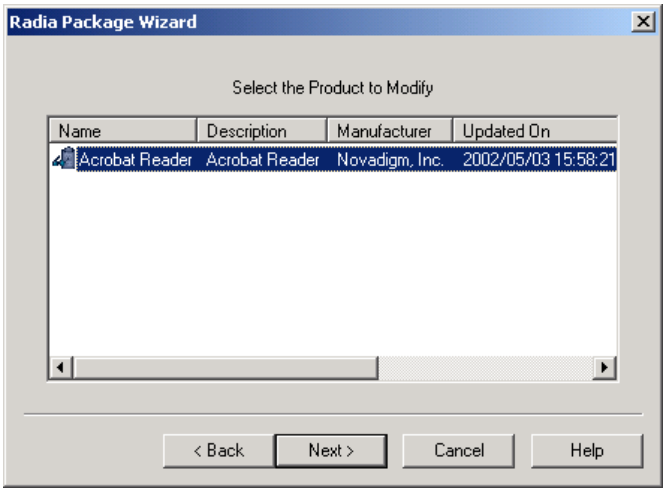

Figure 5.96 ~ Radia Package Wizard – Select the Product to Modify.

6. Select the appropriate product, and then click **Next**.
The **Select the Package to Modify** dialog box opens.

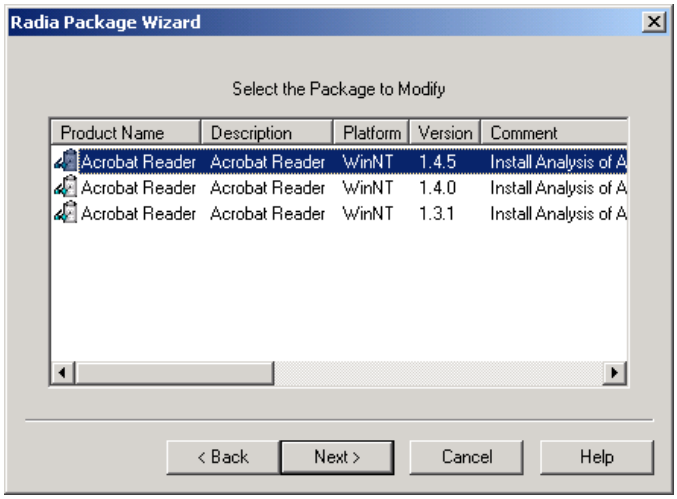

Figure 5.97 ~ Radia Package Wizard – Select the Package to Modify.

7. Select the appropriate package, and then click **Next**.
The **Advanced Options** dialog box opens.

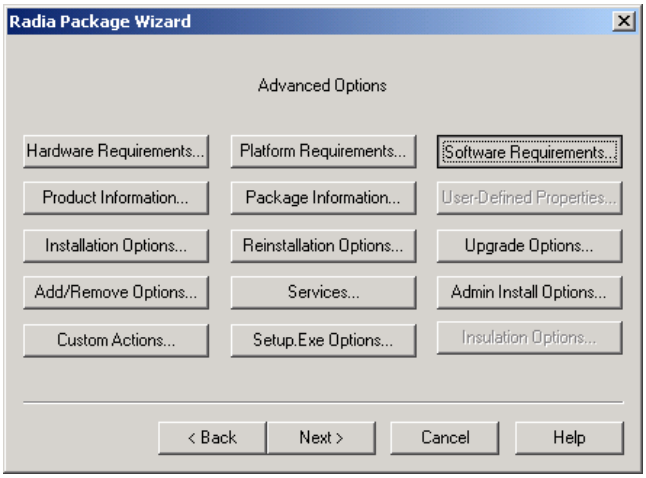

Figure 5.98 ~ Radia Package Wizard – Advanced Options.

8. Click **Software Requirements**.

The **Package Software Requirements** dialog box opens.

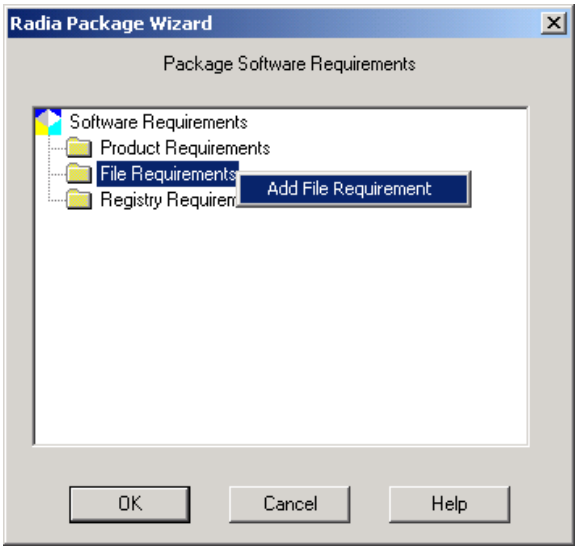

Figure 5.99 ~ Radia Package Wizard – Package Software Requirements.

9. Right-click **File Requirements**, and select **Add File Requirement**.
The **Add File Requirement** dialog box opens.

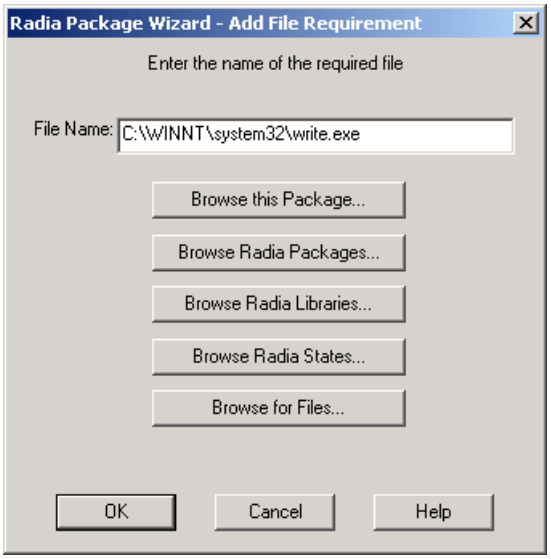

Figure 5.100 ~ Radia Package Wizard – Add File Requirement.

10. Click **Browse for Files** and navigate to **WINNT\system32\write.exe**.

11. Click **OK**.

12. Click **OK** again.
The **Edit File Requirements** dialog box opens.

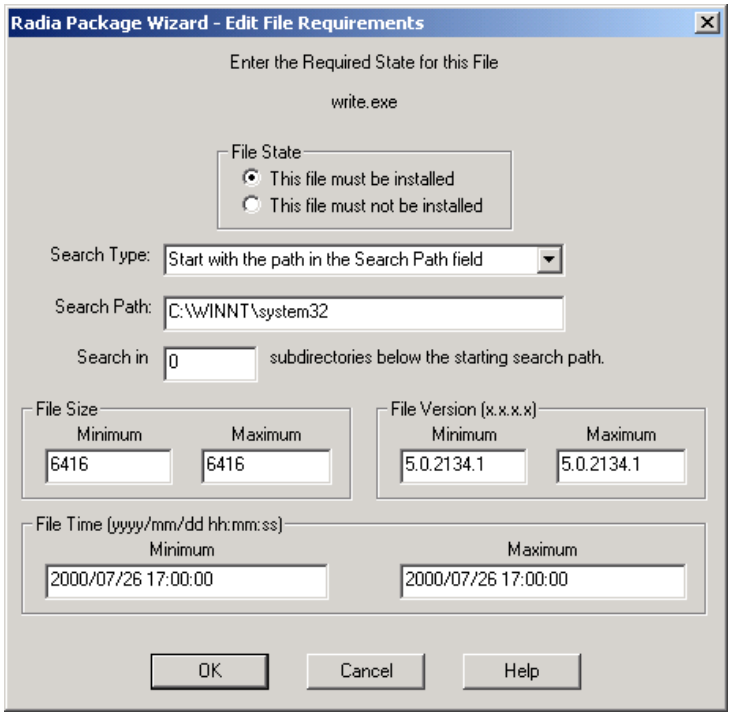

Figure 5.101 ~ Radia Package Wizard – Edit File Requirements.

13. In the **File State** area, select **This file must be installed**.

The Radia Package Wizard pre-fills the remaining information for WRITE.EXE based on the version that exists on the machine.

14. Click **OK** to return to the **Package Software Requirements** dialog box. WRITE.EXE is listed as a software requirement for the package. If it does not exist on the target machine, the package will not be installed.

Figure 5.102 ~ Radia Package Wizard – Package Software Requirements.

15. Click **OK** again to return to the **Advanced Options** dialog box.

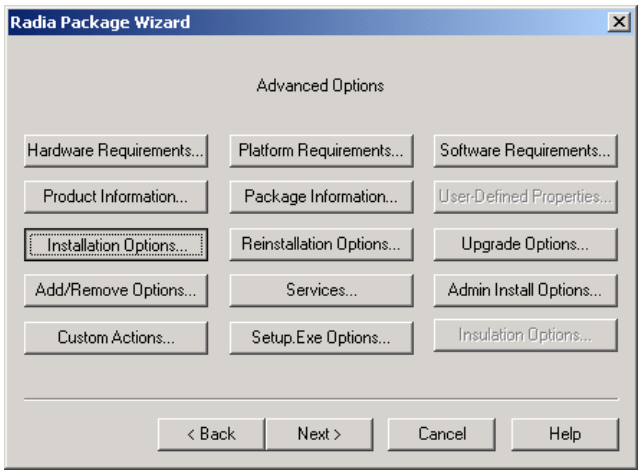

Figure 5.103 ~ Radia Package Wizard – Advanced Options.

16. Click **Installation Options**.

The **Installation Options** dialog box opens.

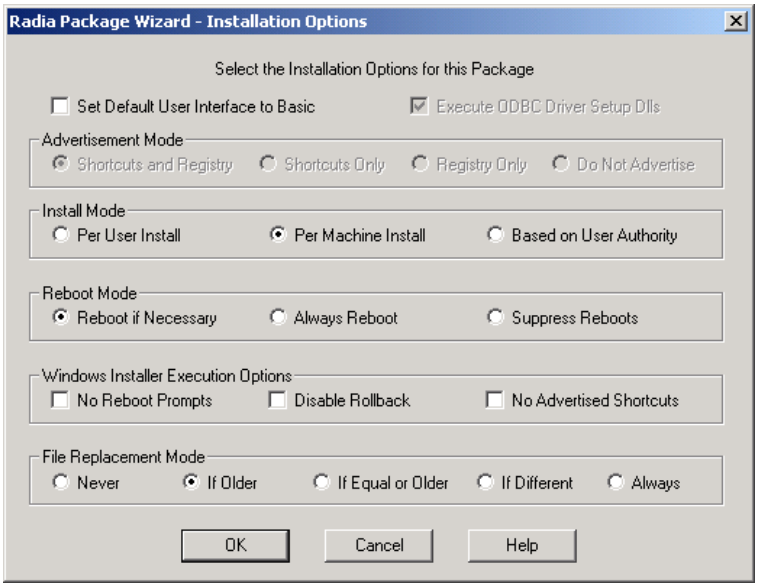

Figure 5.104 ~ Radia Package Wizard – Installation Options.

17. In the **Install Mode** area, select **Per Machine Install**. This option installs the application for all users. When another user runs the application on the same machine, only the user specific settings will be installed.
18. Click **OK**.
19. Click **Next**.

    The **Select Finish to change the package information** dialog box opens.

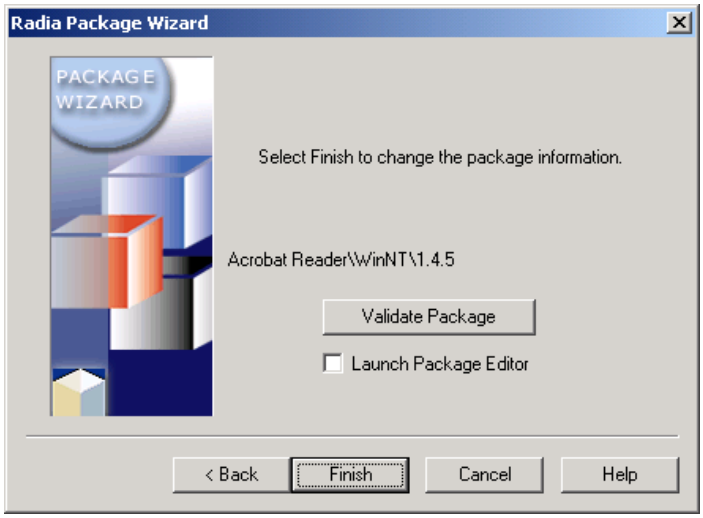

Figure 5.105 ~ Radia Package Wizard – Select Finish to change the package information.

20. Click **Finish** to complete the package.

Upgrading Multiple Products

Sometimes you need to update a certain resource in several products, and you do not want to sort through all your products to find the applications that use the resource. The Radia Package Wizard will find the packages that use a resource, and update the resource in any of these products. To upgrade multiple products, you will need multiple Products to upgrade and a Library that contains the resources you wish to update.

To upgrade multiple products

1. Start Radia Packager for Windows Explorer and select the custom option **Package components.**

   OR

   From the Radia Packager for Windows Installer Program Group, start the **Radia Package Wizard**.

   The **Select an Action** dialog box opens.

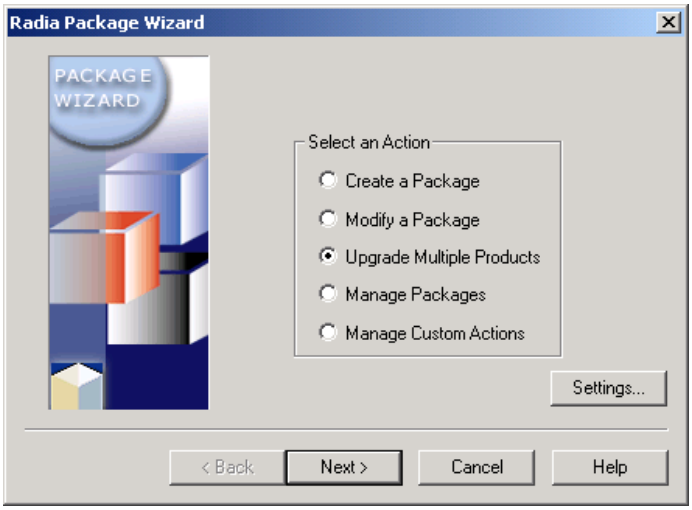

Figure 5.106 ~ Radia Package Wizard – Select an Action.

2. Select **Upgrade Multiple Products**.
3. Click **Next**.

   The **Select the Products to Search for Packages with Matching Files** dialog box opens.

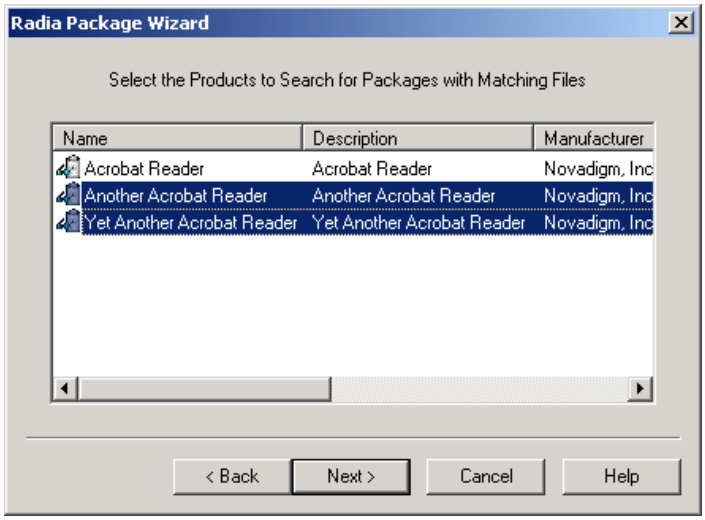

Figure 5.107 ~ Radia Package Wizard – Select the Products to Search for Packages with Matching Files.

4. Select the products to search for files to upgrade and then click **Next**.

The **Select the Library that will Upgrade the Product(s)** dialog box opens.

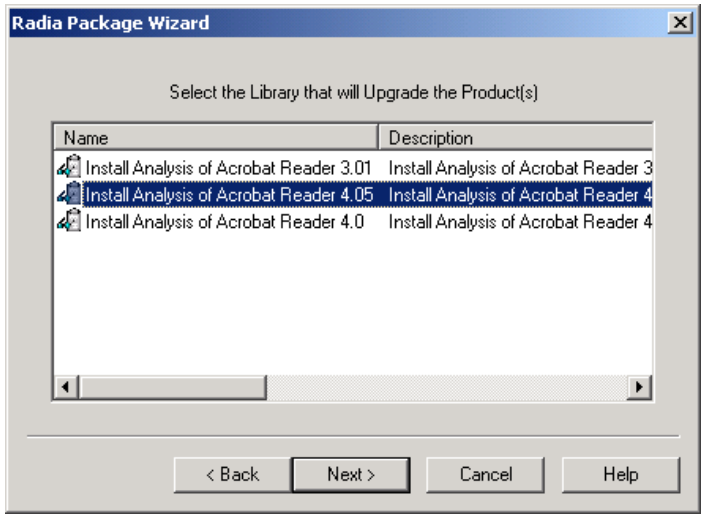

Figure 5.108 ~ Radia Package Wizard – Select the Library that will Upgrade the Products.

5. Select a library from which the products will be upgraded.
6. Click **Next**.

   The **Set the Package Search Options** dialog box opens. Use this dialog box to set limits on the Search Criteria.

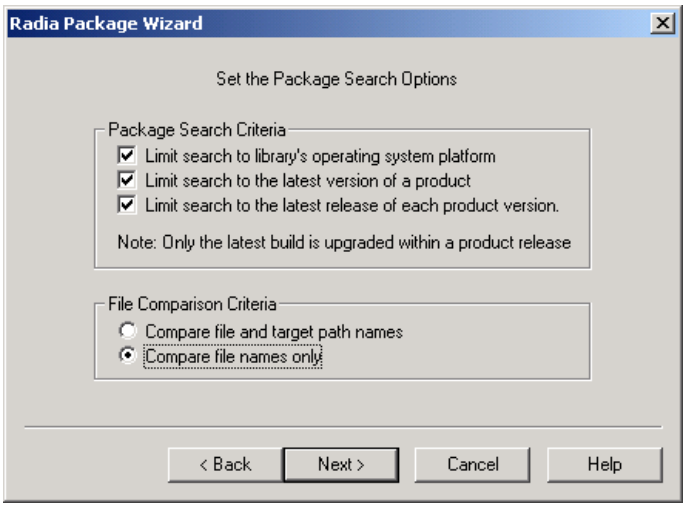

Figure 5.109 ~ Radia Package Wizard – Set the Package Search Options.

7. In the **Package Search Criteria** area, select **Limit the search to library's operating system platform**.
8. In the **File Comparison Criteria** area, select **Compare file names only**.
9. Click **Next**.

   The Radia Package Wizard searches for matching files to upgrade.

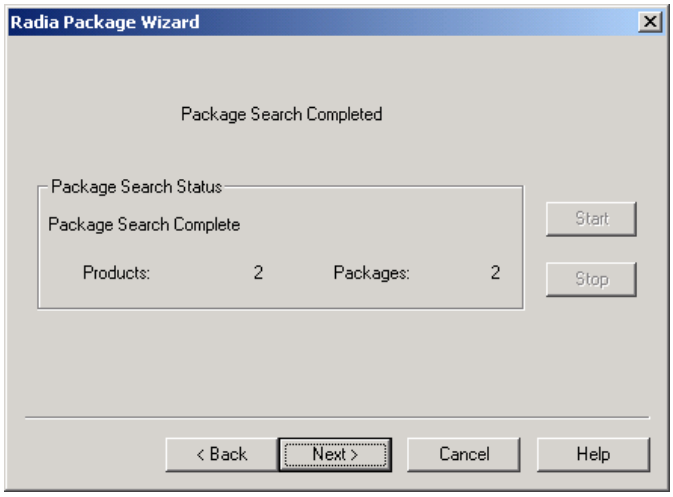

Figure 5.110 ~ Radia Package Wizard – Package Search Completed.

10. Click **Next**.

The **Select the Packages and Files to Upgrade** dialog box opens. You are given the ability to select which files to upgrade in the packages.

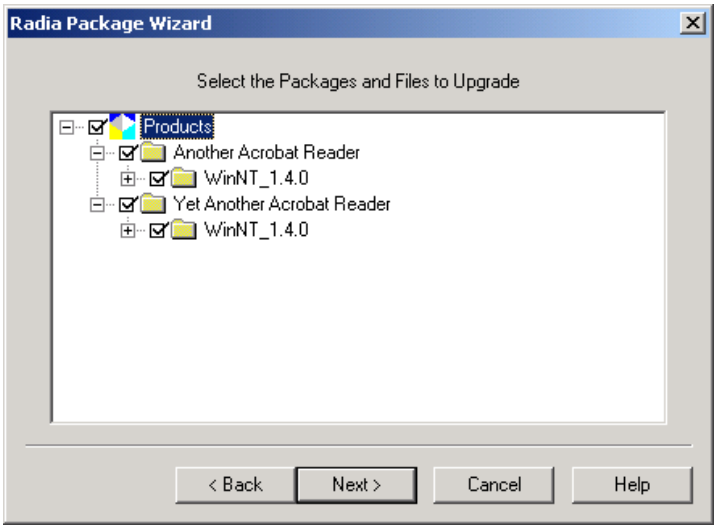

Figure 5.111 ~ Radia Package Wizard – Select the Packages and Files to Upgrade.

11. Select the packages and files to upgrade.

> **Note**
>
> For more information about any of the changes, right-click a file and select **Properties** from the shortcut menu. **Properties** displays information for the version of the file that currently exists in the package and the version to which it will be upgraded.

12. Click **Next**.

The **Package File Options** dialog box opens.

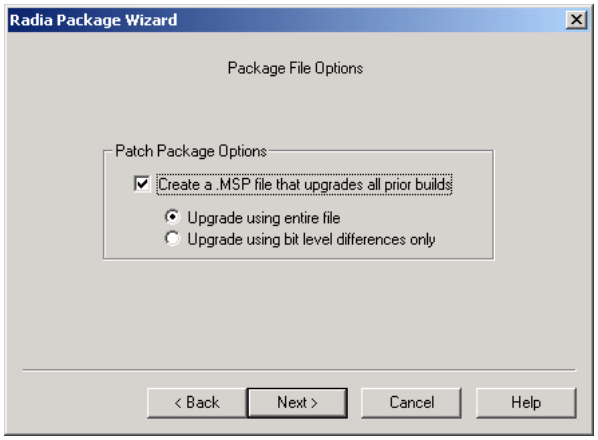

Figure 5.112 ~ Radia Package Wizard – Package File Options.

13. Click **Next** to upgrade the product.

> **Note**
>
> When you upgrade multiple products, a new build of the package is created in the Product directory, complete with Patch Package if you desire.

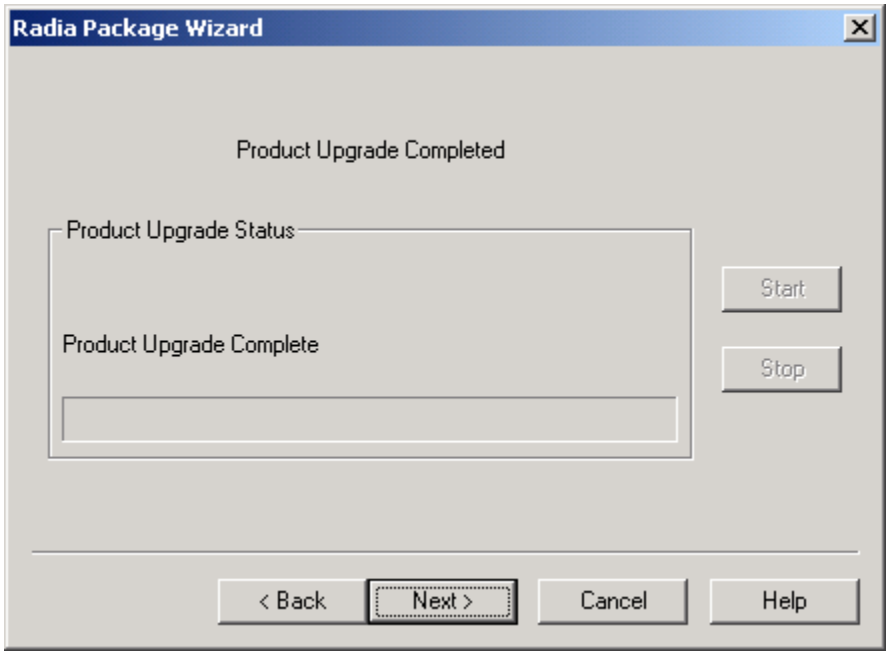

Figure 5.113 ~ Radia Package Wizard – Product Upgrade Completed.

14. Click **Next** when the product update is complete.

15. Click **Finish** to exit the Radia Package Wizard.

16. Use Windows Explorer to view the upgrade packages within the product.

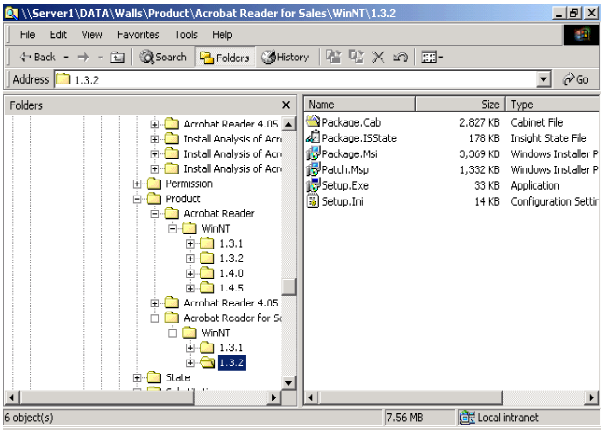

Figure 5.114 ~ Upgrade packages within a product.

Editing a Package Template

You can use package templates to create packages that contain the same basic information. If you are using any kind of user interface, you can also modify that to give your packages a common appearance. In this section, you will modify a copy of the existing Radia Package Template.

To modify a package template

1. Start Radia Packager for Windows Explorer and select the custom option **Package components**.

   OR

   From the Radia Packager for Windows Installer Program Group, start the **Radia Package Wizard**.

   The **Select an Action** dialog box opens.

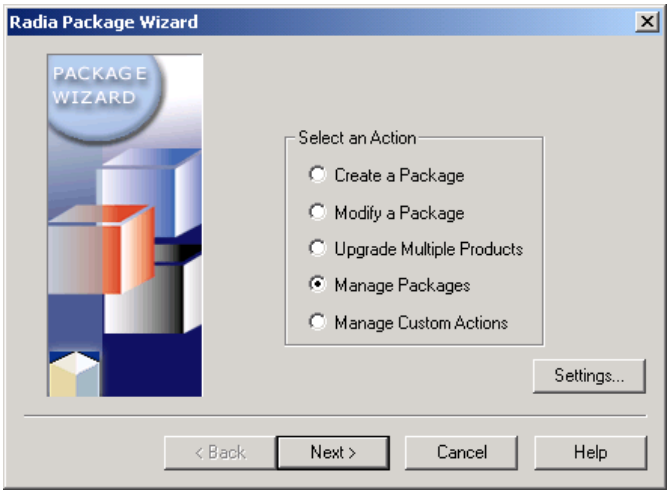

Figure 5.115 ~ Radia Package Wizard – Select an Action.

2. Select **Manage Packages**.
3. Click **Next**.
   The **Package Management Action** dialog box opens.

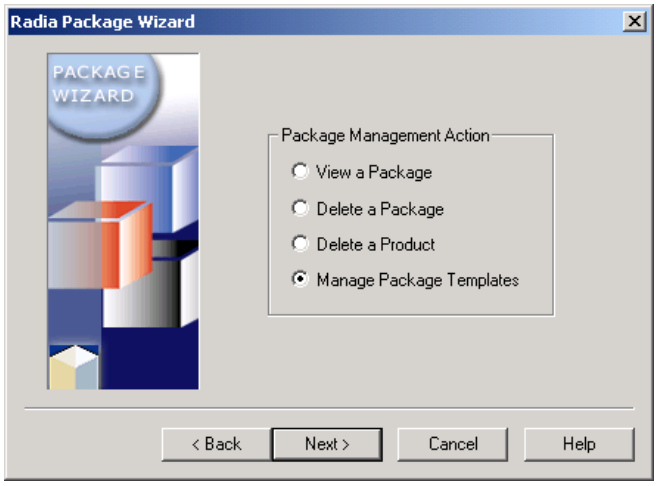

Figure 5.116 ~ Radia Package Wizard – Package Management Action.

4. Select **Manage Package Templates**.
5. Click **Next**.

   The **Select the Package Template File to be Managed** dialog box opens.

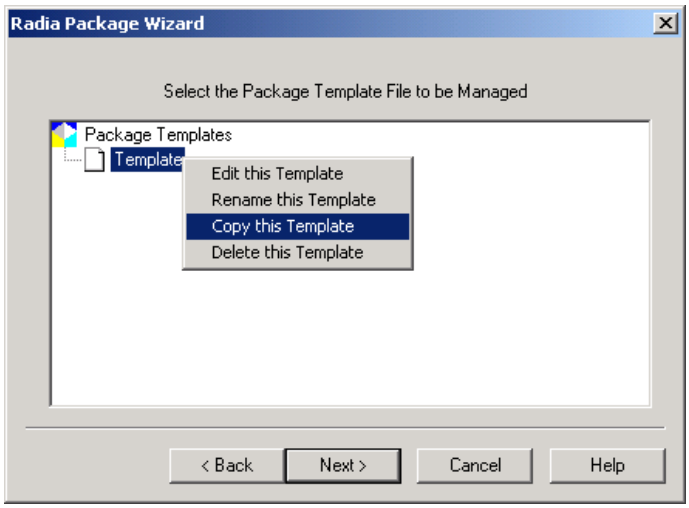

Figure 5.117 ~ Radia Package Wizard – Select the Package Template File to be Managed.

6. Right-click **Templates** and select **Copy this Template**. If you want to edit an existing template, right-click the template and select **Edit this Template**.

   The **Copy Template** dialog box opens.

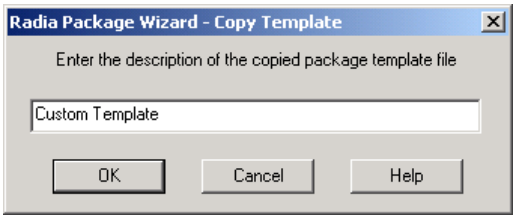

Figure 5.118 ~ Radia Package Wizard – Copy Template.

7. In the text box, type a description for the new template.
8. Click **OK**.

The **Edit Template** dialog box opens. See the following procedures for more information.

To modify a bitmap

Note

See the procedure *To modify a package template* on page 190 for instructions on how to access the **Modify the package template settings** dialog box.

1. In the **Modify the Package Template settings** dialog box, right-click the name of the bitmap image that you want to modify, such as **bannrbmp**.

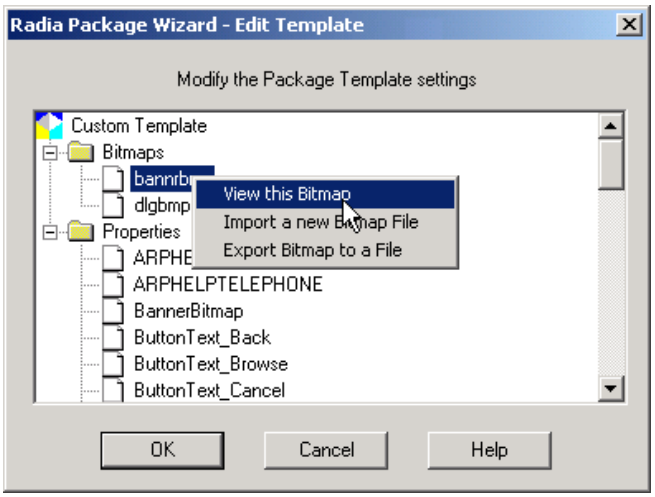

Figure 5.119 ~ Radia Package Wizard – Edit Template – modify package template settings.

Note

If you want to preview the bitmap, right-click the name of the bitmap and select **View this Bitmap**. Click the close button to exit the preview of the image.

2. Select **Export Bitmap to a File**.

   The **Bitmap Export** dialog box opens.

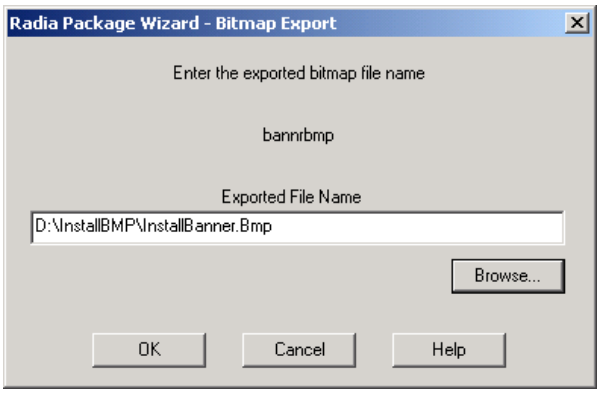

Figure 5.120 ~ Radia Package Wizard – Bitmap Export.

3. In the **Exported File Name** text box, type the path and filename for the file that you want to export.
4. Click **OK**.
5. Leave the Radia Package Wizard open and use an image editor, such as Paint, to modify the bitmap.
6. Save the file when you are done with your changes.
7. Go back to the Radia Package Wizard, where the **Modify the package template settings** dialog box is open.

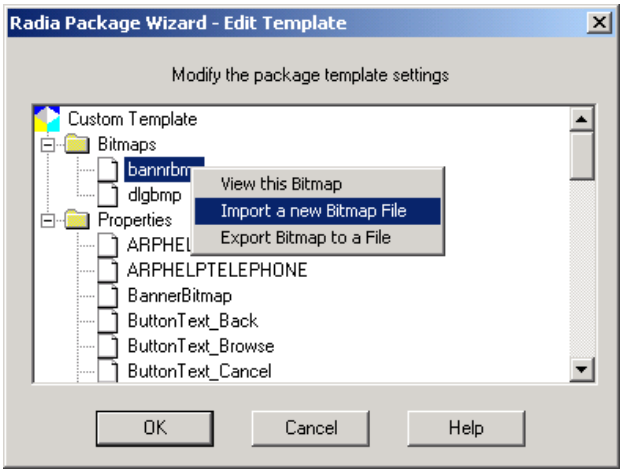

Figure 5.121 ~ Radia Package Wizard – Edit Template – Modify the package template settings.

8. Right-click the name of the bitmap that you want to modify, such as **bannrbmp**.
9. Select **Import a new Bitmap File**.
   The **Bitmap Import** dialog box opens.

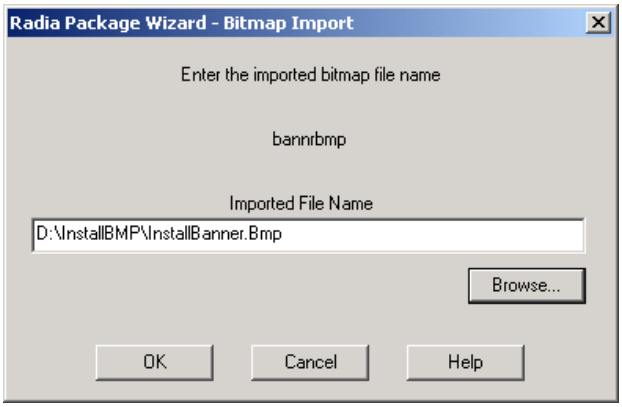

Figure 5.122 ~ Radia Package Wizard – Bitmap Import.

10. Click **Browse** and select the bitmap that you modified.
11. Click **OK** to return to the **Modify the package template settings** dialog box.

To modify a property

Note

See the procedure *To modify a package template* on page 190 for instructions on how to access the **Modify the package template settings** dialog box.

1. In the **Modify the package template settings** dialog box, right-click the name of the property that you want to modify, and click **Edit this Property**.

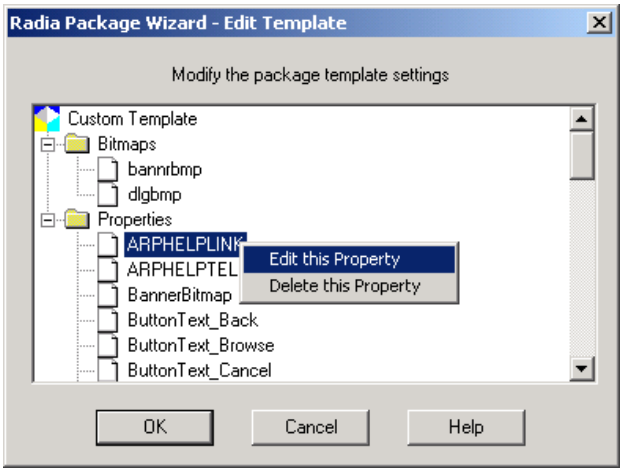

Figure 5.123 ~ Radia Package Wizard – Edit Template – Modify the package template settings.

The **Edit property** dialog box opens.

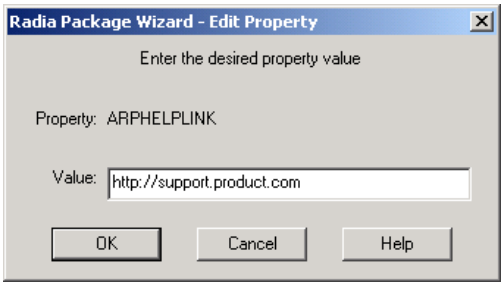

Figure 5.124 ~ Radia Package Wizard – Edit Property.

2. Type the value that you want included in this property for all of your packages.
3. Click **OK**.

To add a property

Note

See the procedure *To modify a package template* on page 190 for instructions on how to access the **Modify the package template settings** dialog box.

1. In the **Modify the package template settings** dialog box, right-click **Properties**. The **Edit Template** dialog box opens.

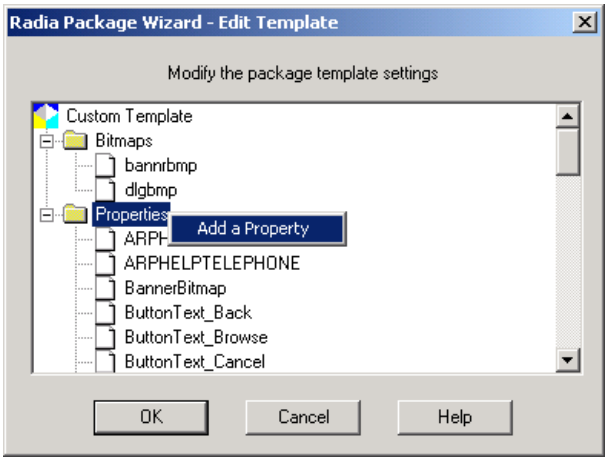

Figure 5.125 ~ Radia Package Wizard – Edit Template.

2. Select **Add a Property**.
3. The **Add Property** dialog box opens.

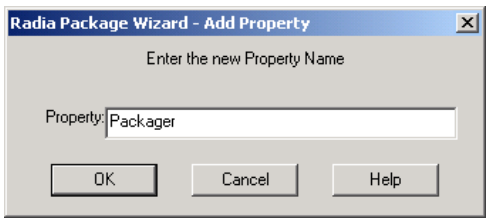

Figure 5.126 ~ Radia Package Wizard – Add Property.

4. Type a name for the property.
5. Click **OK**.

   The **Edit Property** dialog box opens.

Figure 5.127 ~ Radia Package Wizard – Edit Template.

6. Type the value that you want in this property for all your packages.
7. Click **OK**.
8. Click OK again to close the **Edit Template** dialog box.
9. In the **Radia Package Wizard**, click **Next**.
10. Click **Finish** to exit the Radia Package Wizard.

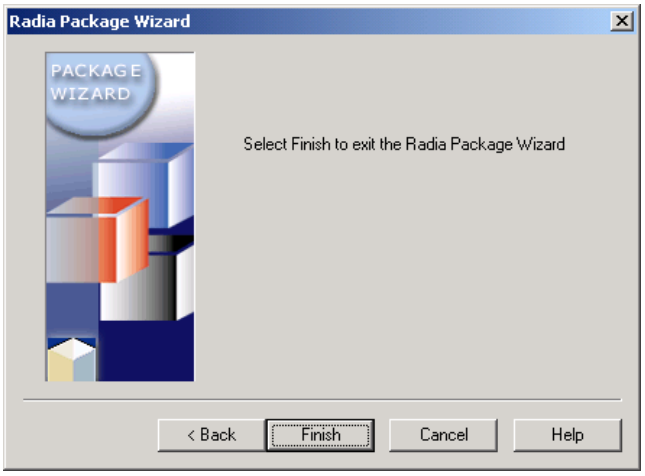

Figure 5.128 ~ Radia Package Wizard – Exiting the Wizard.

Enabling Radia Advanced Features Machine/User Support

In order to take advantage of the Radia Advanced Features machine/user support, the Client Components of the Radia Extensions for Windows Installer must be installed on your client computers. Refer to the *Radia Extensions for Windows Installer Getting Started Guide* for more information.

Note to I-Lab Users

Radia Advanced Features are only licensed in the Radia Extensions I-Lab product for testing purposes within a laboratory environment.

Prerequisites

Machine/user support requires the following prerequisite components:

- Radia Client version 3.1 or above
- Radia Administrator Workstation version 3.1 or above

Enabling Machine/User Support

To Publish a Package to Radia with Machine/User support enabled

1. Publish a package to Radia using the Radia Publisher (refer to the *Radia Publisher Guide* for details).
2. At the **Radia Machine/User Support** window, select the check box. This will enable the four previously unavailable check boxes (notice the first three check boxes are selected by default - this is the recommended configuration for enabling machine/user support). An explanation of each check box option follows.
   - **Enable shortcuts when the User MSI package is installed.**
     During the machine install, all shortcuts are removed from the **All Users** profile. Then during the user connect or install these shortcuts are placed in the user's profile.
   - **Enable HKCR registry values when the User MSI package is installed.**
     During the machine install, all HKEY_CLASSES_ROOT registry information is removed from the Local Machine (HKLM) hive. During the user connect, this information is placed in the Current User (HKCU) hive. This enables features such as right-click options available only when the package is installed.
   - **Enable private Exe execution when the User MSI package is installed.**
     Permissions will be placed on non-shared Exe files in the package to deny access to all

non-administrators (administrators can change permissions, so no restriction is given to them). When the user install is run, these deny permissions will be removed. This prevents direct access to the executable.

- **Enable private Dll execution when the User MSI package is installed.**
  This is the same as the previous option, except for Dlls. Shared Dlls are not denied access to prevent issues with other applications that might use the resource.

3. Finish promoting the package as you would normally.
4. Once the package is promoted to the Radia Database, use the Radia System Explorer to create a new Service (see the *Radia System Explorer Guide* for details).
   - Create the new Service with the Application Target Type set to **Application Manager.**
   - After the Service is created, make sure the ZSYSACCT is set to N. It should not be set to Y. If the ZSYSACCT is set to Y, you will not be able to deploy the user portion of the install.
   - Make sure the Service's ZSVCMODE is set to MU. This will enable Machine/User installations of the Service.
5. Use the System Explorer to attach the new Service to a policy instance. The next time a client connect takes place, the appropriate files will be installed.
   - When the client computer connects using the **Machine** context, the machine portion of the application is installed. At this point, the application is not yet available to any users on the machine.
   - When a connection is made in **User** context, the application is enabled for the user designated.

Summary

- Use the Radia State Wizard advanced functions to combine and manage state files.
- Use the Radia Library Wizard advanced functions to compress or decompress a library.
- Use the Radia Package Wizard advanced functions to rebuild a package, upgrade multiple products, or edit a package template.
- Select the Custom option **Identify resources to package** from Radia Packager for Windows Installer to open the Radia State Wizard.
- Select the Custom option **Refine selected components** from Radia Packager for Windows Installer to open the Radia Library Wizard.
- Select the Custom option **Package components** from the Radia Packager for Windows Installer to open the Radia Package Wizard.

6

Filtering

At the end of this chapter, you will:

- Be able to set default filters.
- Be able to view filters.
- Be able to create filters.
- Be able to use filters.

Filters are one of the most important ingredients in building application installation packages. You can define filters that allow you to limit the amount of data gathered during a machine scan or that are written to a Radia library and installation package. Filters allow you to include or exclude the following:

- A specific drive or set of drives, including network drives
- A specific directory or set of directories
- Specific files
- A specific Windows registry key or set of keys

When running the Radia Packager for Windows Installer Unified package creation process, as well as the Radia State Wizard or Radia Library Wizard, you can create a new filter, apply additional inclusion or exclusion filters to the existing filter settings, save your filter, reset the filter to the last saved or selected filter, or load a previously saved filter.

Using Default Filters

The Radia Packager for Windows Installer provides default filters. Installations can select their own default filters on the **Radia Packager for Windows Installer Setting** dialog box. On the **Install Analysis**, **Machine Scan**, and **Library** tabs of this Setting dialog box, a **Default Installation Filter File** entry specifies installation-defined default filtering options for the various types of processes. If you do not define an installation filter, a default filter is applied.

For example, you can build a base machine default scan filter that excludes network directories from being scanned when performing an install analysis, since no resources are to be installed to network locations.

The dialog box used to view and modify the filter allows you to expand/collapse the branches or the filter and select/remove an item from the filter set.

Viewing Filters

You can use standard Windows Explorer techniques to expand and collapse the branches of the path, files, and registry tree:

⊞ Click on a collapsed branch indicator to expand the first level below the item.

⊞ Right-click on a collapsed branch indicator to fully expand all levels below the item.

⊟ Click on an expanded item indicator to collapse it to that level.

Including Resources in a Filter

A check box to the left of an entry indicates whether the entry, and any of its children, will be included in the scan.

☑ A check mark in a white box indicates the entry will be included in the scan. For entries with children, a check mark in a white box indicates all children of the entries are included in the scan.

☑ A gray box with a check mark indicates that the entry and at least one, but not all, of the children of the entry are selected.

☐ A gray box without a check mark indicates that the entry itself is not included in the scan, but at least one of its children is selected.

☐ A white box indicates the entry is not included; nor are any of its children.

Creating Filters

Filters can be created and modified in both the Radia State Wizard and Radia Library Wizard. You can manually create filters or dynamically create them from other States, Libraries, or Packages. Filters can also be created during the State or Library creation process, as well as during the Typical Unified package creation process.

To manually create filters

1. Start the Radia Packager for Windows Installer and select the custom option **Identify resources to package.**

   OR

   From the Radia Packager for Windows Installer Program Group, start the **Radia State Wizard**.

   The **Select an Action** dialog box opens.

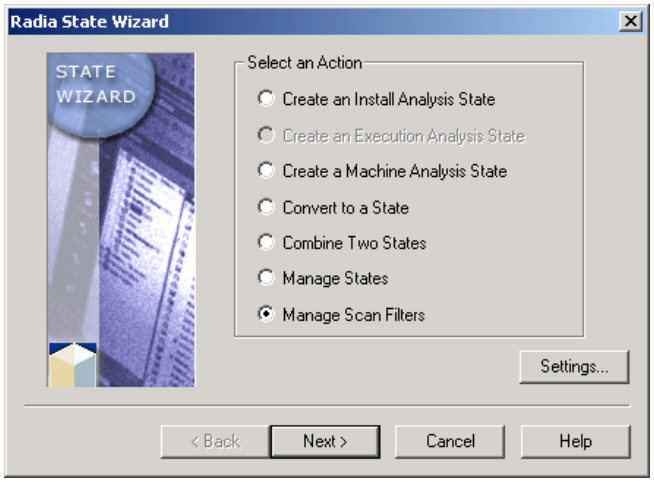

Figure 6.1 ~ Radia State Wizard – Select an Action.

2. Select **Manage Scan Filters**.
3. Click **Next**.

   The **Select a Filter Management Action** dialog box opens.

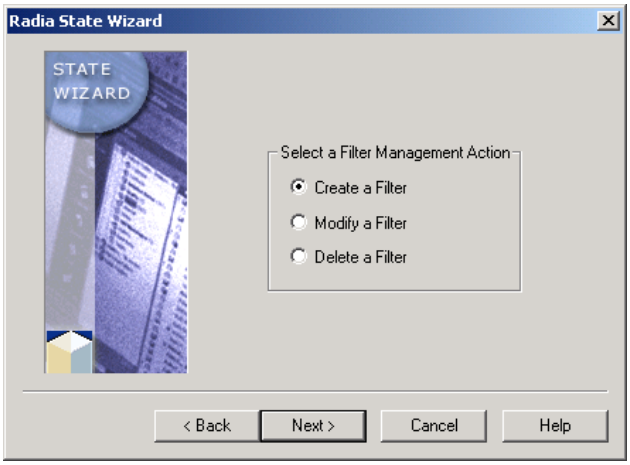

Figure 6.2 ~ Radia State Wizard – Select a Filter Management Action.

4. Select **Create a Filter**.

5. Click **Next**.
   The **Select the Filter Source** dialog box opens.

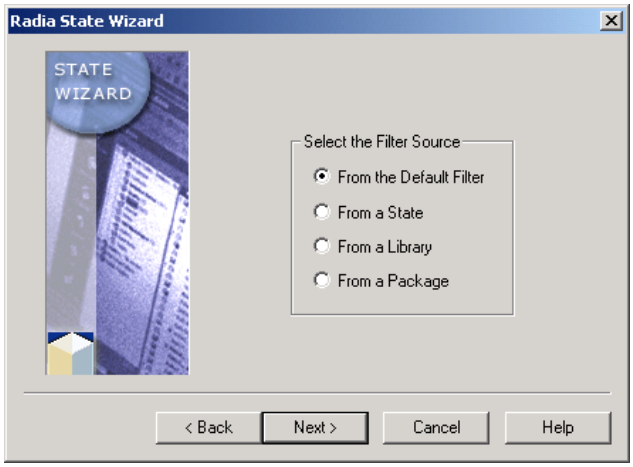

Figure 6.3 ~ Radia State Wizard – Select the Filter Source.

6. Click **From the Default Filter**.
7. Click **Next**.
   The **Select the Filter Type** dialog box opens.

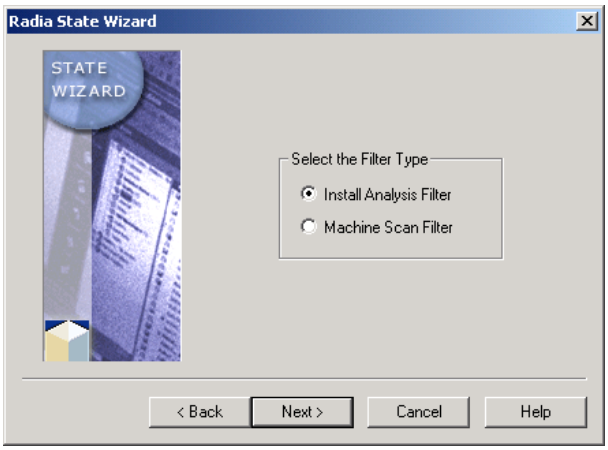

Figure 6.4 ~ Radia State Wizard – Select the Filter Type.

8. Select **Install Analysis Filter**.
9. Click **Next**.

   The **Select the paths, files and registry entries to scan** dialog box opens.

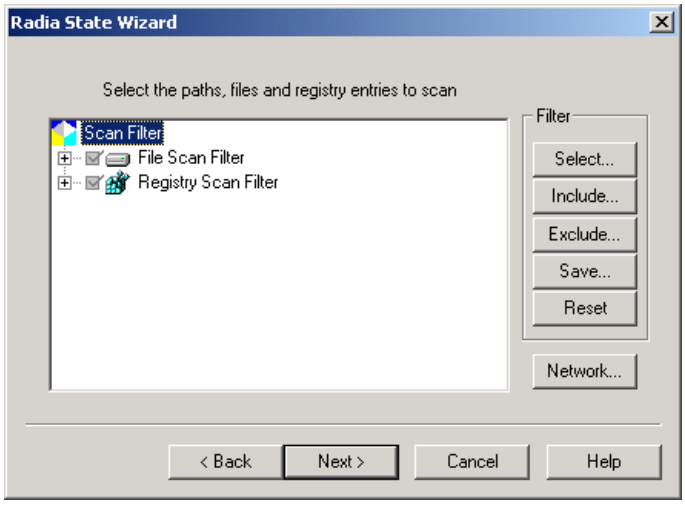

Figure 6.5 ~ Radia State Wizard – Select the paths, files, and registry entries to scan.

10. Modify the filter using the following techniques to select/remove entries from the scan.

> **Note**
>
> Press the **F10** on the keyboard to expand this dialog box.

- Select a collapsed branch to include all children of an entry in the scan.
- To select individual entries in a branch, expand the branch and select each entry to be included in the scan.
- Click **Include** or **Exclude** to select one or more filters from which to add or subtract resources from the current filter. This is an extremely powerful feature for eliminating known resources from a state, library, or package file if the resources already exist on the workstations where the packaged software is to be installed. For example, you may have an OLE filter and an MDAC filter to eliminate resources from being included into distribution packages, if these resources are already on the machine.

11. Click **Next**.

The **Set the Description for the Filter** dialog box opens.

Figure 6.6 ~ Radia State Wizard – Set the Description for the Filter.

12. In the **Description** text box, type in a description, and then click **Next**.

13. Click **Finish** to save the filter.

To create a filter from a library

1. Start the Radia Packager for Windows Installer and select the custom option **Refine selected components.**

   OR

   From the Radia Packager for Windows Installer Program Group, start the **Radia Library Wizard**.

   The **Select an Action** dialog box opens.

Figure 6.7 ~ Radia Library Wizard – Select an Action.

2. Select **Manage Library Filters**.
3. Click **Next**.

   The **Select a Filter Management Action** dialog box opens.

Figure 6.8 ~ Radia Library Wizard – Select a Filter Management Action.

4. Select **Create a Filter**.
5. Click **Next**.

   The **Select the Filter Creation Mode** dialog box opens.

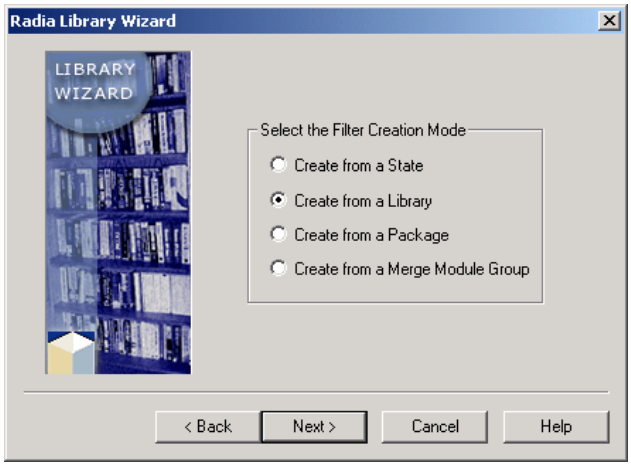

Figure 6.9 ~ Radia Library Wizard – Select the Filter Creation Mode.

6. Select **Create from a Library**.

7. Click **Next**.

The **Select the Library to Create the Filter From** dialog box opens.

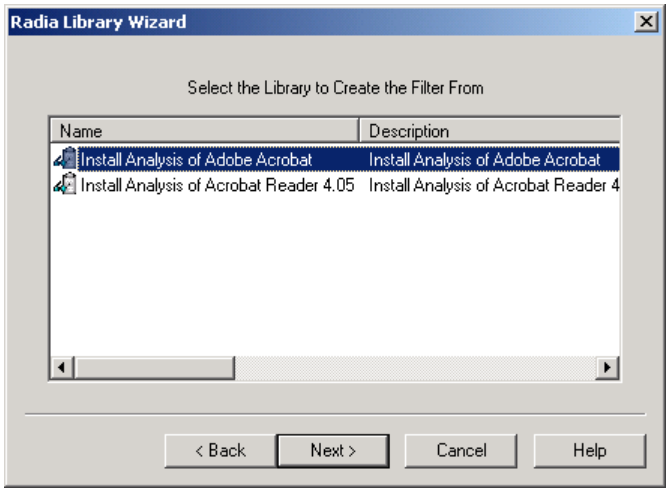

Figure 6.10 ~ Radia Library Wizard – Select the Library to Create the Filter from.

8. Select the library from which you want to create the filter.

9. Click **Next**.

The **Select the Paths, Files and Registry Entries to Include** dialog box opens.

Figure 6.11 ~ Radia Library Wizard – Select the Paths, Files and Registry Entries to Include.

10. If necessary, modify the filter. See *Using Filters* on page 214 for more information.

11. Click **Next**.

12. The **Set Description for the Filter** dialog box opens.

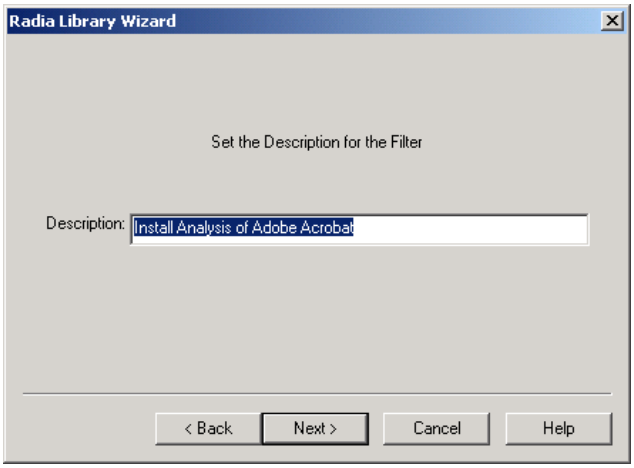

Figure 6.12 ~ Radia Library Wizard – Set the Description for the Filter.

13. In the **Description** text box, type a description of the filter.

14. Click **Next**.

15. Click **Finish** to exit the Radia Library Wizard.

Using Filters

When creating or modifying a filter using the Radia State Wizard or the Radia Library Wizard, you will notice a set of buttons on the right side of the **Select Paths, Files and Registry Entries to Include** dialog box. This section describes how to use these buttons.

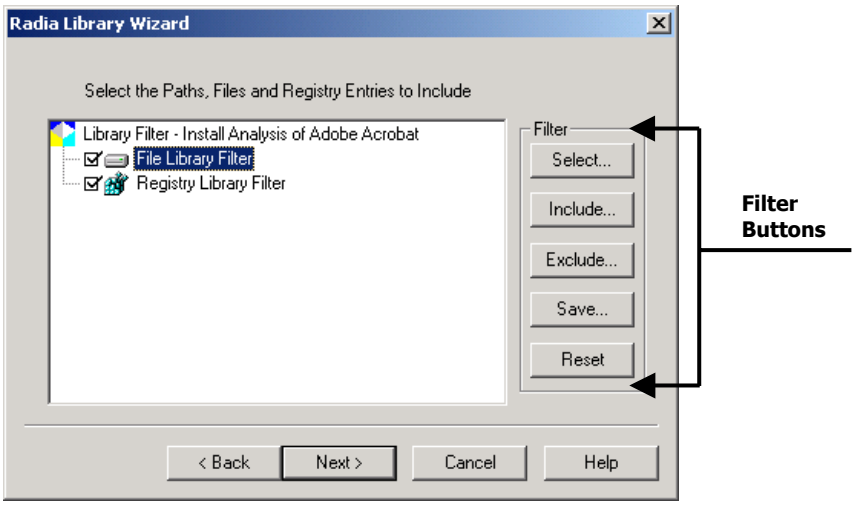

Figure 6.13 ~ Radia Library Wizard – Filter buttons in the Radia Library Wizard.

Table 6.1 ~ Filter buttons

Button	Description
Select	Click **Select** to select and load a previously saved filter from those listed in the Filter Directory. This will clear all other applied filters. **Note:** The Filter Directory is specified on the Paths dialog box of the Radia Default Settings dialog box, launched from the **Settings** button from any Radia Wizard.
Include	Click **Include** to include a previously saved filter's resources from those listed in the Filter Directory. The resources selected in the inclusion filter are selected for the current filter. If filters are multi-selected at the same time, they are applied in the order listed.

Table 6.1 ~ Filter buttons

Button	Description
Exclude	Click **Exclude** to exclude a previously saved filter's resources from those listed in the Filter Directory. The resources selected in the exclusion filter are cleared from the current filter. If filters are multi-selected, they are applied in the order listed.
Save	Click **Save** to save or replace the current filter with a user-assigned description. **Note:** Click **OK** to save the filter to the location specified as the Filter Directory in the Radia Settings. If the filter name is the same, it replaces the filter with the given name.
Reset	Click **Reset** to reset the current scan filter to the last saved or selected filter.
Network (available in the Radia State Wizard)	From the Radia State Wizard, click **Network** to select an external path on the Network to add to a state filter. Click **Network** to add remote paths to the File Scan Filter list. You can access any path available through Windows Network Neighborhood. Use the Browse Folder dialog box to locate the desired path. Click on the selected path (the item will be highlighted), and then click **OK** to add the path to the list of File Scan Filter locations. Add one path at a time.

Resetting to a Default, Previously Selected, Saved Filter

Click **Reset** to reset to the last saved, selected, or default filter for the analysis type.

Click **Select** to apply a previously saved filter from the list of filters saved to the **Saved Filter path**, as set in the **Default Settings** dialog box. Load a saved filter to ensure the same filter is used for a given analysis.

Saving Filters

Click **Save** to save the current filter to its own file, with a description you enter. Filters may also be created from the **Manage Scan Filters** or **Manage Library Filters** options of the Radia State and Radia Library Wizards, respectively.

Deleting Filters

To delete individual filters, click **Select** to view the list of filters. Then, right-click and select **Delete** from the shortcut menu.

Filters may also be deleted from the **Manage Scan Filters** or **Manage Library Filter** options of the Radia State and Radia Library Wizards, respectively.

Summary

- Use the Radia State Wizard to create filters. The Radia State Wizard opens when you select **Identify resources to package** from the Radia Packager for Windows Installer.
- Use the Radia Library Wizard to apply filters. The Radia Library Wizard opens when you select **Refine selected components** from the Radia Packager for Windows Installer.

7

Merge Modules

At the end of this chapter, you will:

- Understand Merge Modules.
- Know how to create a Merge Module.
- Know how to create a Merge Module Group.

Merge Modules are simplified Windows Installer packages that cannot be installed on their own, but are instead installed by merging them with a complete MSI. These simplified packages are designed to install shared components. Ideally, no two packages would install the same component. Instead, that component would be installed from one resource shared between them. This allows Windows Installer to always properly increment the resource counter for the component. If a component is identified by two different unique identifiers, then the reference count will not be properly incremented when the second application's dll overwrites the first application's.

Radia packages are built in such a way that this does not become a problem, but you may want to work with pre-existing Merge Modules or to create your own to pass to other administrators on your team. The Radia Merge Module Wizard provides you with a familiar interface (the process is very similar to the Radia Package Wizard) to create and maintain your Merge Modules.

HP supplies a group of basic Windows 2000 Merge Modules with the Radia Packager for Windows Installer installation. These Merge Modules and ones you create are easily applied to your packages through the Radia Library Wizard. Through this chapter, you will learn some of the basics for working with Merge Modules.

Creating a Merge Module

Merge Modules, just as packages, are created from libraries. In a fashion similar to Packages, Merge Modules are created in Merge Module Products. Within a Merge Module Product you might have several different versions of Merge Module packages. Use the Radia Merge Module Wizard to create Merge Modules. To launch the Radia Merge Module Wizard, select it from the Radia Packager for Windows Installer Program Group.

Merge Module Groups

To be integrated into a library, a Merge Module must first be part of a Merge Module Group. Merge Module Groups help to organize Merge Modules and provide a single location from which the modules are referenced.

Analyzing a Merge Module

If you receive a pre-built Merge Module, you will need to analyze it and create a Radia Merge Module product for it before you are able to include it in your libraries and packages.

Applying a Merge Module

Because Merge Modules deal with resources and how they are used in your package, they are integrated through the Radia Library Wizard. You may recall seeing a dialog box requesting

Merge Module information when creating a library. That is where Merge Module information is displayed (including Merge Modules that match resources in your library) and you are permitted to select Merge Modules to include in your final package. You can also add Merge Modules by modifying a library.

To apply a merge module

1. Start the Radia Packager for Windows Installer and select the custom option **Refine selected components.**

   OR

   From the Radia Packager for Windows Installer Program Group, start the **Radia Library Wizard**.

Figure 7.1 ~ Radia Library Wizard – Select an Action.

2. Select **Modify a Library**.
3. Click **Next**.

   The **Select the Type of Modification** dialog box opens.

Figure 7.2 ~ Radia Library Wizard – Select the Type of Modification.

4. Select **Modify Included Merge Modules**.
5. Click **Next.**

   The **Select the Library to Modify** dialog box opens.

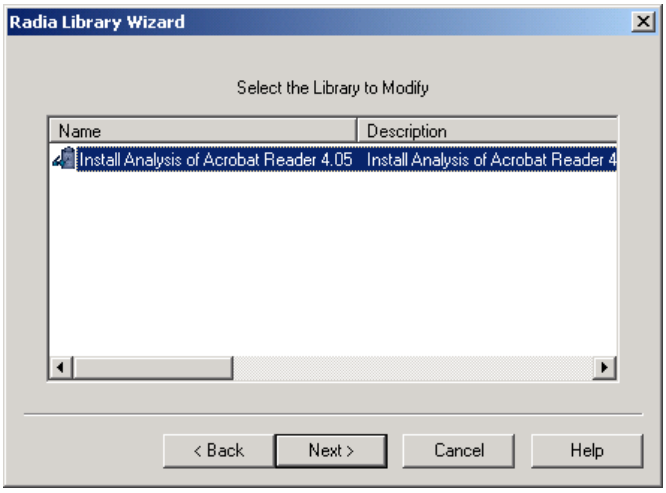

Figure 7.3 ~ Radia Library Wizard –Select the Library to Modify.

6. Select the library to modify and click **Next.**

The **Select the Merge Module Packages to include in the Library** dialog box opens.

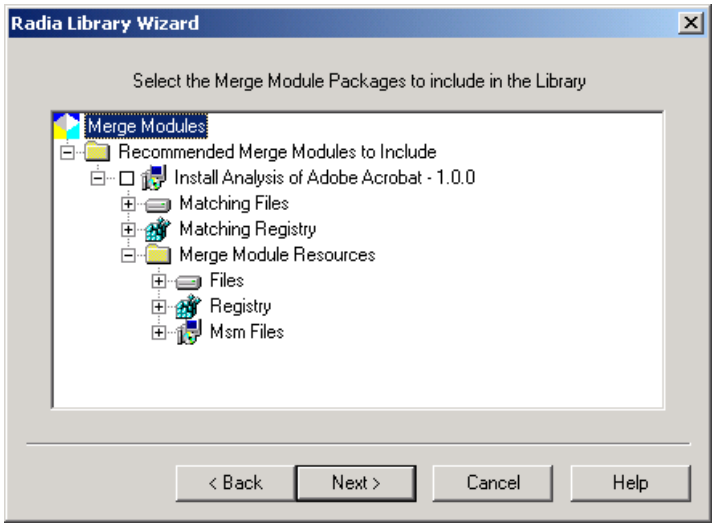

Figure 7.4 ~ Radia Library Wizard – Select the Merge Module Packages to include in the Library.

7. Right-click on **Merge Modules** and select **Select Merge Module Group**.

The **Select the Merge Module Group that contains the Merge Module to include** dialog box opens.

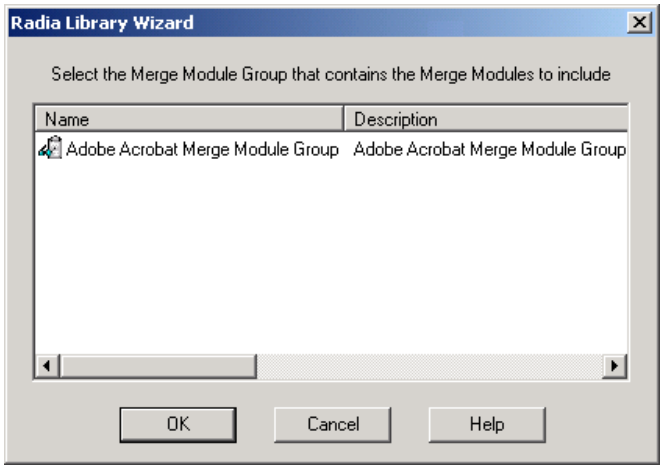

Figure 7.5 ~ Radia Library Wizard – Select the Merge Module Group that contains the Merge Modules to include.

8. Select the Merge Module Group with which you want to work. A default Merge Module group can be selected in the Radia Packager for Windows Installer Settings and recommended Merge Modules will be displayed automatically with every library you create.

The Radia Library Wizard quickly determines which Merge Modules in your Merge Module Group contain matching resources with your library and recommends Merge Modules to include. The exact matching resources are displayed to allow you to make an informed decision on whether or not to include the module.

9. Click **OK.**

The **Select the Merge Module Packages to include in the Library** dialog box opens.

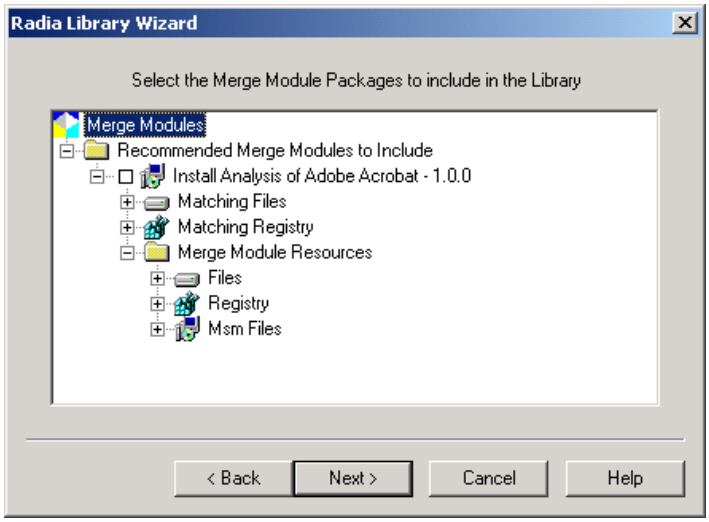

Figure 7.6 ~ Radia Library Wizard – Select Merge Module Packages to include in Library.

10. Select the check the box next to the modules you wish to include.

11. Click **Next**.

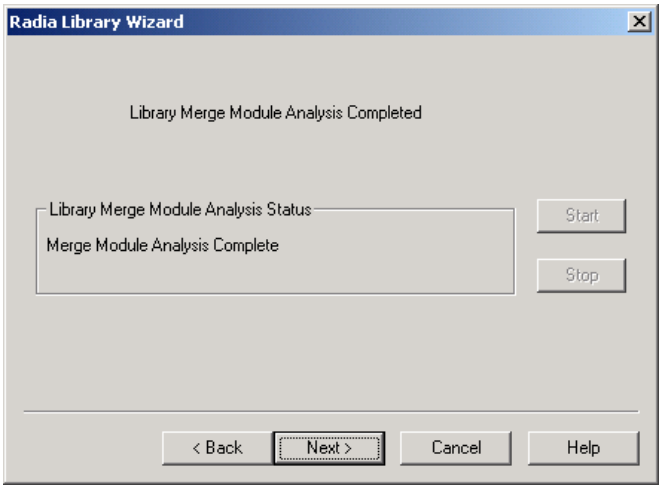

Figure 7.7 ~ Radia Library Wizard – Library Merge Module Analysis Completed.

The Radia Library Wizard performs a quick analysis to adjust the library to accommodate the Merge Module and its resources.

12. Click **Next**.

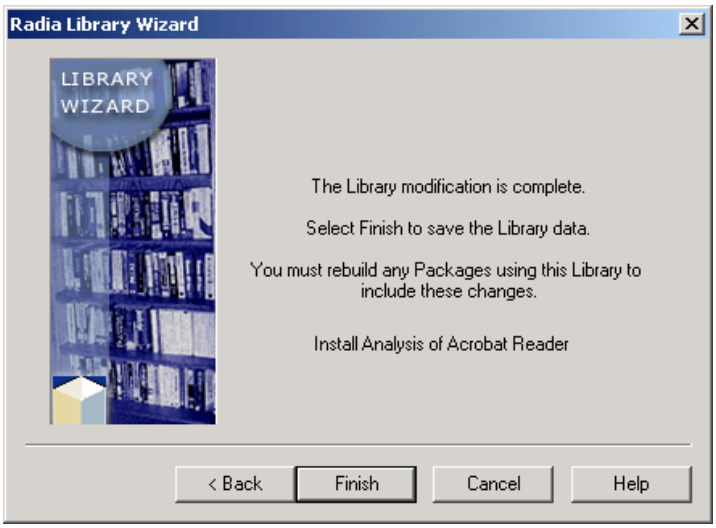

Figure 7.8 ~ Radia Library Wizard – The library modification is complete.

13. Click **Finish** to return to the Radia Packager for Windows Installer.

The Radia Library Wizard is finished. Next you need to build (or rebuild, as appropriate) your package. This process is the same as detailed in previous chapters with the exception of the **Package Results** dialog box, where you will see the included Merge Module.

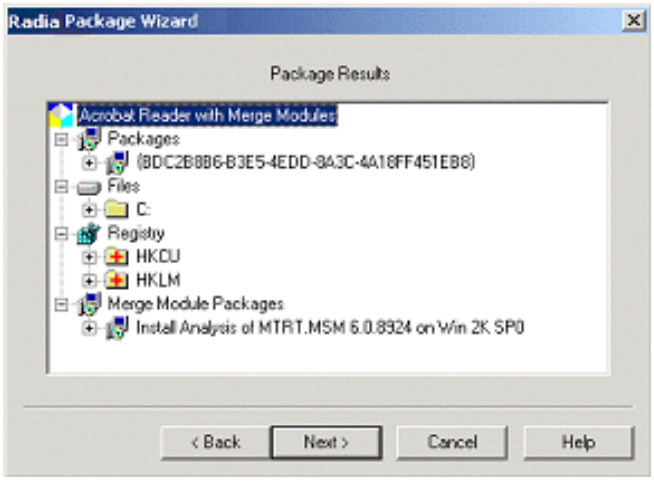

Figure 7.9 ~ Radia Library Wizard – Package Results dialog box with included Merge Module.

You will also notice some extra files in the Packages directory. Most of these files are for reference. If you are promoting a package or moving it to another location, the only additional files you will need to include are the **Merge#.Cab** files.

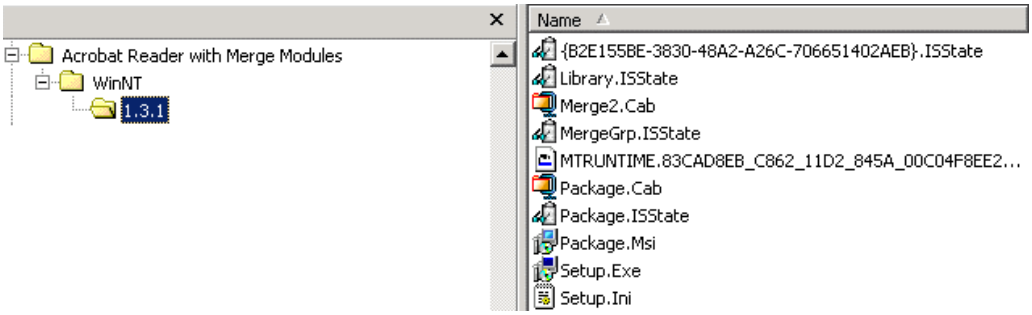

Figure 7.10 ~ Additional files in the package directory.

Summary

- Use the Radia Merge Module Wizard to create a Merge Module.
- Launch the Radia Merge Module Wizard from the Radia Packager for Windows Installer Program Group.
- In order to apply Merge Modules, they must belong to a Merge Module Group.
- Use the Radia Library Wizard to apply a Merge Module.

Radia Packager for Windows Installer Settings

To make your job easier, the Radia Packager for Windows Installer provides a way to set default filters, paths, and other settings. This appendix provides an overview of the different configurable settings.

To access the configurable settings, select any **Custom** action from the Radia Packager for Windows Installer and click the **Settings** button located on the next dialog.

The **Settings** button is also available on the Radia Packager for Windows Installer (Process) Menu and on the first dialog box of all Wizards.

Paths (1) Tab

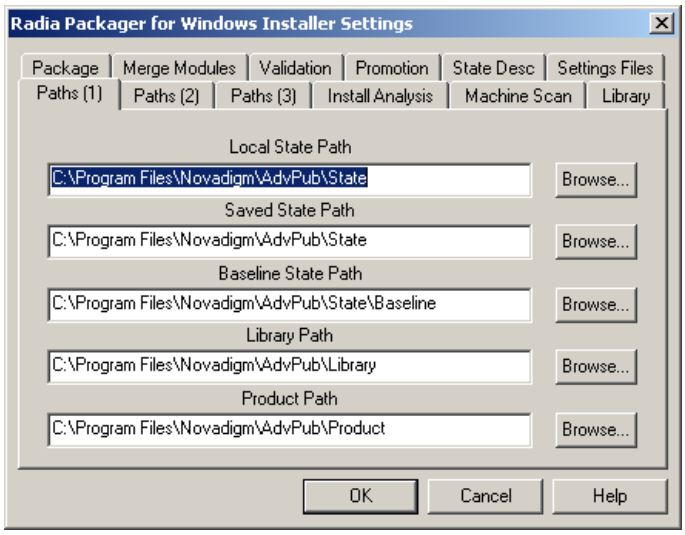

Figure A.1 ~ Radia Packager for Windows Installer Settings – Paths (1) tab.

The **Paths (1)** tab contains the default location for each of the paths. By default, each path points to the local C drive.

> **Note**
>
> When in a lab environment where machines are frequently reset to a new base, click **Browse** to redirect the path settings to permanent locations on an enterprise server.

- **Local State Path**
  Specifies the location of temporary state files when running the Radia State Wizard.
- **Saved State Path**
  Specifies the location where state files are saved by the Radia State Wizard.
- **Baseline State Path**
  Specifies the location of baseline state files (reusable "before states").
- **Library Path**
  Specifies the location of Radia libraries.

- **Product Path**
  Specifies the location of Radia Extensions for Windows Installer products. Packages are created with the following directory structure under the product path:

  ```
  Product Path / product_name / platform (WinNT or Win9x) / version number
  / Package
  ```

  Note

  Paths can be designated with UNC names or mapped drives.

Paths (2) Tab

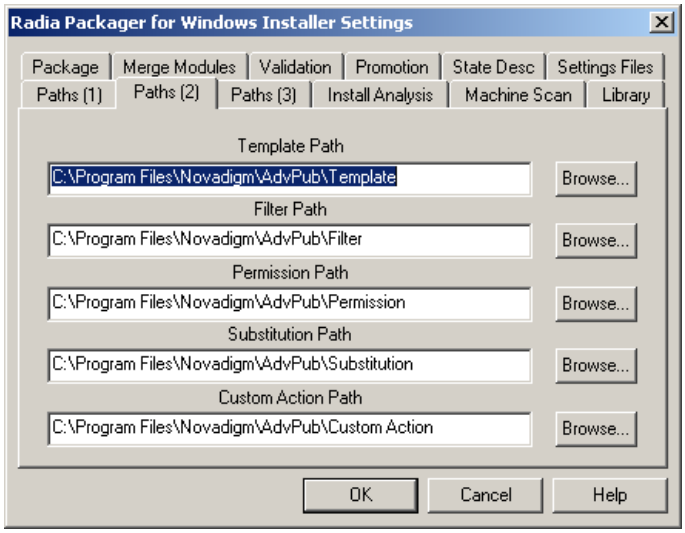

Figure A.2 ~ Radia Packager for Windows Installer Settings – Paths (2) tab.

The **Paths (2)** tab contains the default location for each of the paths. By default, each path points to the local C drive.

Note

When in a lab environment where machines are frequently reset to a new base, click **Browse** to redirect the path settings to permanent locations on an enterprise server.

- **Template File Path**
  Specifies the location of Windows Installer package template (*.MSI) file, used with the Radia Package Wizard. Two subdirectories of the template path are used to locate configuration files for the Radia Package Wizard: User Properties and Admin Properties on the **Package** tab.
- **Filter Path**
  Specifies the location of filter files, used with the Radia State Wizard and the Radia Library Wizard.
- **Permission Path**
  Specifies the location of permission files, used with the Radia Library Wizard.

- **Substitution Path**
  Specifies the location of substitution state files, used with the Radia Library Wizard to customize an installation through powerful search and replace functions.
- **Custom Action Path**
  Specifies the location of your Custom Action files, if any, used with the Radia Package Wizard.

Paths (3) Tab

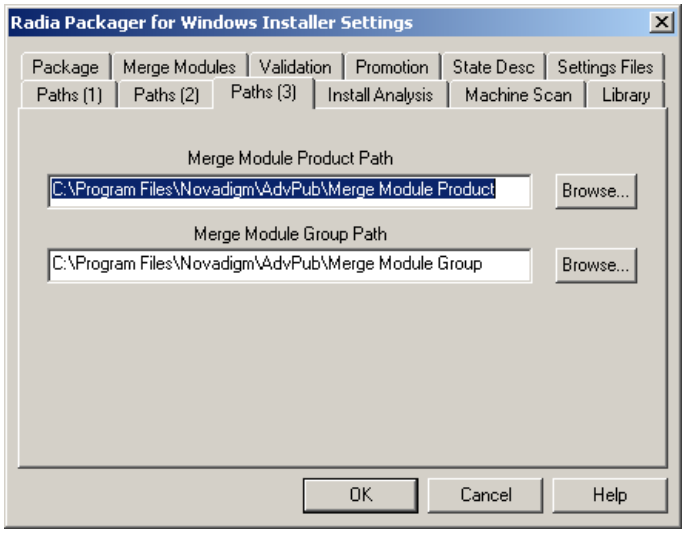

Figure A.3 ~ Radia Packager for Windows Installer Settings – Paths (3) tab.

The **Paths (3)** tab contains the default location for each of the paths. By default, each path points to the local C drive.

Note

When in a lab environment where machines are frequently reset to a new base, click **Browse** to redirect the path settings to permanent locations on an enterprise server.

- **Merge Module Product Path**
  Specifies the location of Radia Extensions for Windows Installer Merger Module products. Merge Module packages are created with the following directory structure under the Merge Module product path:

  ```
  Merge Module Product Path \ mergemodule_name \ platform (WinNT or Win9x) \
     version number \ Merge Module
  ```

- **Merge Module Group Path**
  Specifies the location of Merge Module Groups. Merge Modules are then applied to a library from a Merge Module Group.

Install Analysis Tab

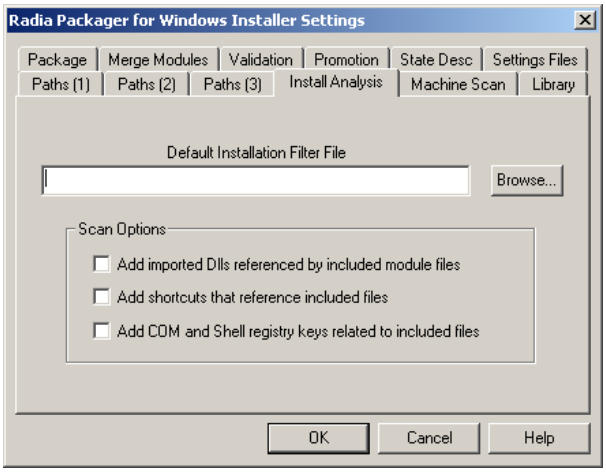

Figure A.4 ~ Radia Packager for Windows Installer Settings – Install Analysis tab.

The **Install Analysis** tab contains defaults that apply when the Radia State Wizard is used to create an install analysis. For creating an install analysis, you can specify:

- the default filter file
- a scan option to **add imported DLLs** referenced by included module files
- a scan option to **add shortcuts** that reference included files
- a scan option to **add COM and Shell registry keys** related to included files

The defaults specified on the **Install Analysis** tab can always be overridden by selecting a different filter or set of scan options when creating an install analysis.

Machine Scan Tab

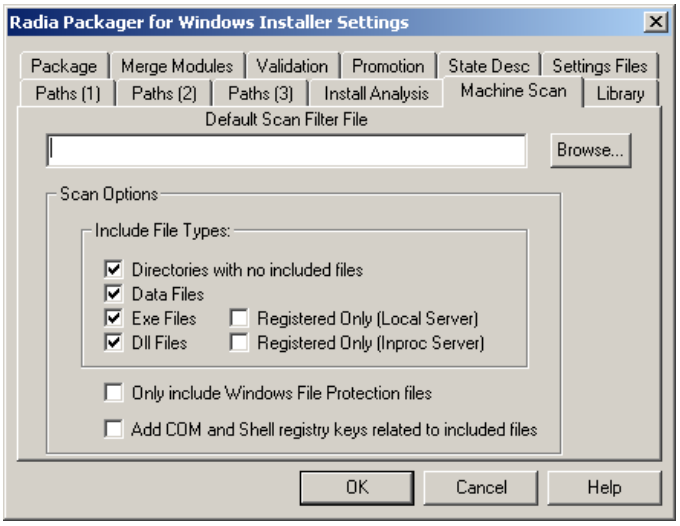

Figure A.5 ~ Radia Packager for Windows Installer Settings – Machine Scan tab.

The **Machine Scan** tab contains default options that apply when you create a Machine Scan from the Radia State Wizard.

- **Default Scan Filter File**
  Browse for the filter to be used when creating a machine scan.
- **Scan Options area**
  Use the Scan Options area to set defaults for including or excluding various file types from a machine scan. The default settings can be overridden whenever a machine scan is run.

Library Tab

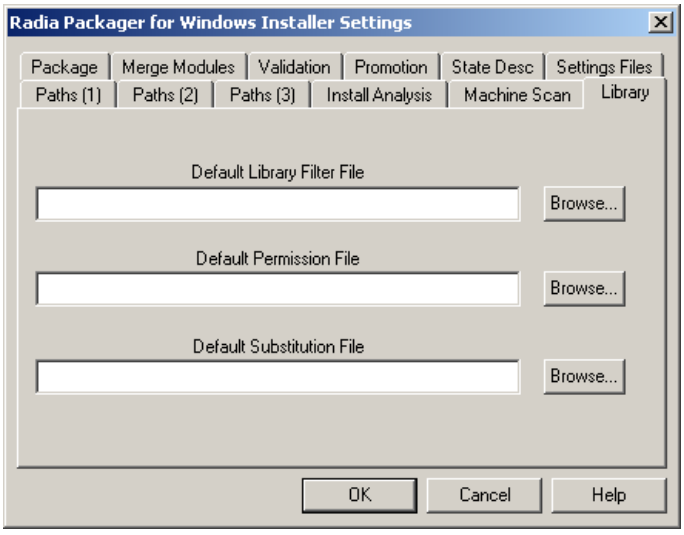

Figure A.6 ~ Radia Packager for Windows Installer Settings – Library tab.

The **Library** tab contains installation or session defaults to be applied when building a Radia library. You can specify:

- a **Default Library Filter File**, to be used when creating a library
- a **Default Permission File**, to be used when assigning permissions to library resources
- a **Default Substitution File**, to be used to assign a default set of substitutions to the library for installation customization, but does not enable any of the resource substitutions

Package Tab

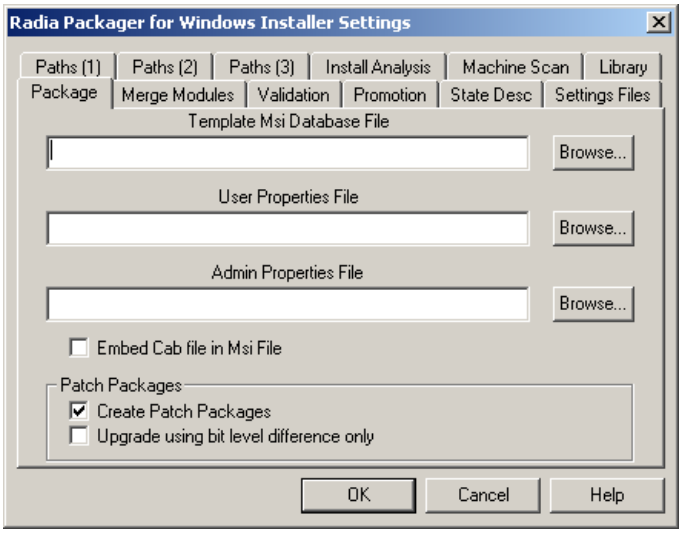

Figure A.7 ~ Radia Packager for Windows Installer Settings – Package tab.

The **Package** tab contains set installation or session packaging defaults.

- **Template MSI Database File (required)**
  Set the *template.msi* database to be used for packaging. The first time a package is created, or if a template is not available, a default template is generated automatically.
- **User Properties File (optional)**
  You can define an .ini file that contains user-defined properties that are to be included into the .msi package. User-defined properties are a very good way to include useful and consistent information in each package. For example, you can designate that inventory and chargeback properties be included in each package. User-defined properties can be edited during the packaging process. A template of the properties and their default values are defined in an .ini file. A sample *SampleUser.ini* file is contained in the **\Novadigm\AdvPub\Template\User Properties** directory. Use this file as a model to create your specific overrides. Place your .ini file in the **\Novadigm\AdvPub\Template\UserProperties** directory, or the template directory you defined. This directory is scanned when **Browse** is clicked. Choosing an .ini file automatically embeds the user-defined properties into the .msi package.
- **Admin Properties File (optional)**
  If the package is going to be installed as an administrative installation to a server from which end users will install the product, you can define an .ini file that contains *administrative property overrides* that are to occur when the end user installs the product from the

administrative installation copy. A sample *SampleAdmin.ini* file is contained in the **\Novadigm\AdvPub\Template\AdminProperties** directory. Use this file as a model to create your specific overrides. Place your .ini file in the **\Novadigm\AdvPub\Template\AdminProperties** directory. This directory is scanned when **Browse** is clicked. Choosing an .ini file automatically embeds the AdminProperty overrides into the .msi package.

- **Embed Cab file in MSI File**
  You can select **Embed the package.cab file in the .msi file** as the default on the **Package File Options** dialog box. This option is generally used for packages that are going to be administratively installed to a server.
- **Patch Packages Options**
  When creating a new build for existing products, these options set the defaults presented to the user on the **Package File Options** dialog box of the Radia Package Wizard for creating patch packages (.msp files). Users can override the default. Note that patch packages are created *in addition to* full .msi files.

Select **Create Patch Packages** as the default. When patch packages are selected, you can have the upgrade default set to **Upgrade using bit-level difference only** (as opposed to **Upgrade using entire file**).

Merge Modules Tab

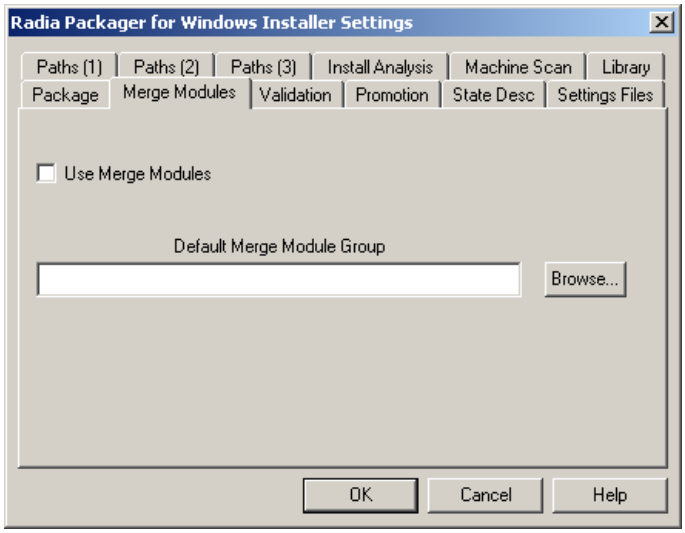

Figure A.8 ~ Radia Packager for Windows Installer Settings – Merge Modules tab.

- Select whether or not to use Merge Module by selecting **Use Merge Modules.**
- Define the location of the default Merge Module Group when using Merge Modules in the **Default Merge Modules Group** text box.

Validation Tab

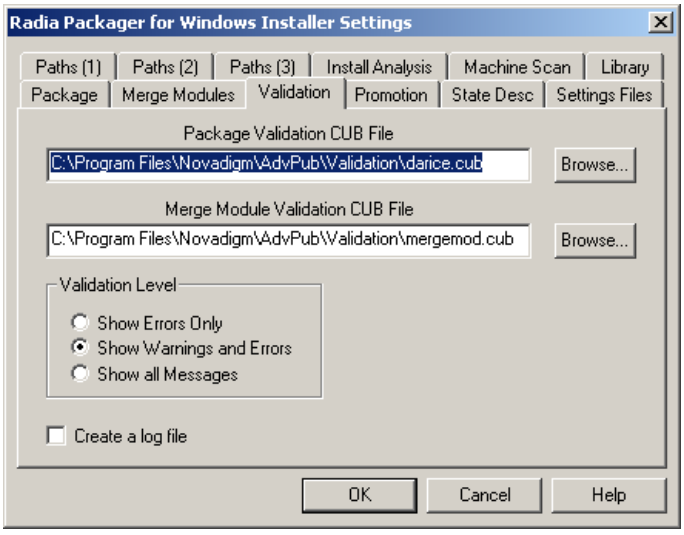

Figure A.9 ~ Radia Packager for Windows Installer Settings – Validation tab.

Windows Installer packages may be validated using an installation-defined validation file after the package has been created. Validation files can be customized by the administrator to enforce installation-specific validation criteria.

- **Package Validation CUB File**
  This is the default validation file name and path for validating packages such as, Installation_Default_Ice.cub. This file will be used for validation when you click **Validate Package** in the **Radia Package Wizard Completion** dialog box (see Figure A.10 on page 240).

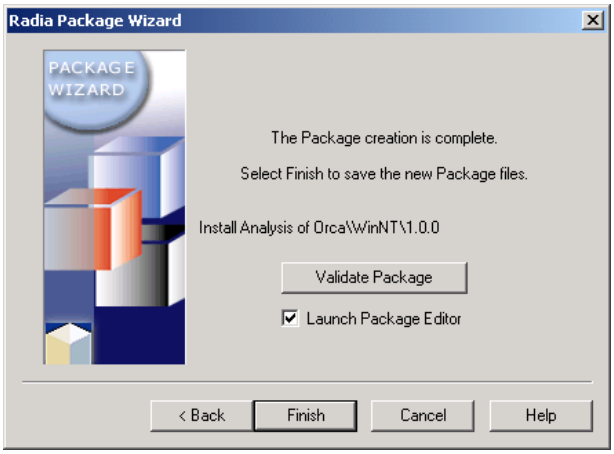

Figure A.10 ~ Radia Package Wizard – Package creation is complete.

- **Merge Module Validation CUB File**
  The default validation file name and path. This file will be used for validation when you click **Validate Merge Module** in the **Merge Module Wizard Completion** dialog box.

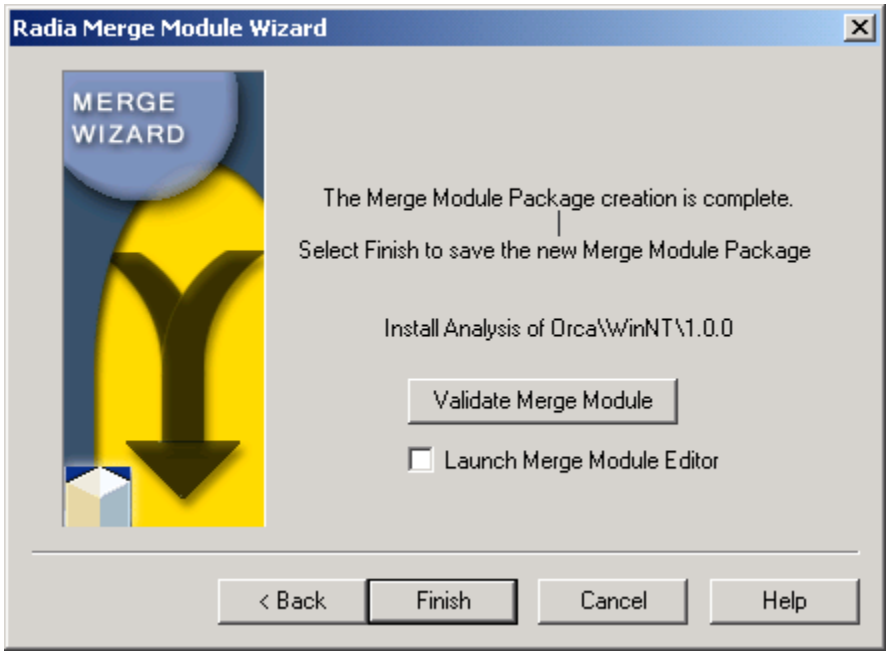

Figure A.11 ~ Radia Merge Module Wizard – Merge Module Package creation is complete.

- **Validation Level**
  Select the type of messages that you want to see during the validation process:
  - Errors only
  - Warnings and errors only
  - All messages
- **Create a log file** check box
  Creates a log file in the same directory where the .MSI file resides, and is named **validate.log**.

Promotion Tab

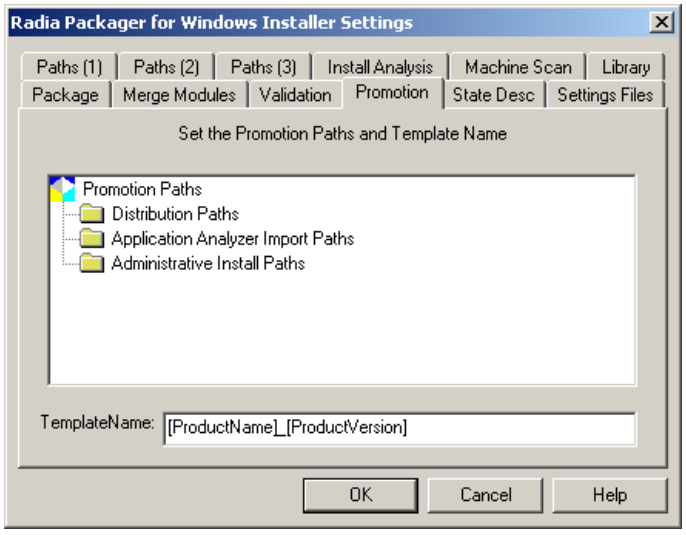

Figure A.12 ~ Radia Packager for Windows Installer Settings – Promotion tab.

Use the **Promotion** tab to set default Distribution Paths and Radia Configuration Analyzer Import Paths. These default entries create drop-down selection lists for the corresponding entry in the Radia Install Wizard promotion functions, from which the user can choose a path. Setting default paths are optional.

TemplateName is used to define the resulting MSI name and directory for the AIP, and the package name and description when publishing to Radia.

To add a default distribution path

1. Right-click the **Distribution Paths** folder.
2. Select **Add a Distribution Path**, and complete the selections.

To remove a default distribution path

1. Right-click an entry from the tree.
2. Select **Delete**.

To add a default import path

1. Right-click the **Configuration Analyzer Import Paths** folder.
2. Select **Add a Configuration Analyzer Import Path**, and complete the selection.

To remove a default import path

1. Right-click the import path from the tree.
2. Select **Delete**.

To add an administrative install path

1. Right-click the **Administrative Install Paths** folder.
2. Select **Add an Admin Install Path**, and complete the selections.

State Desc Tab

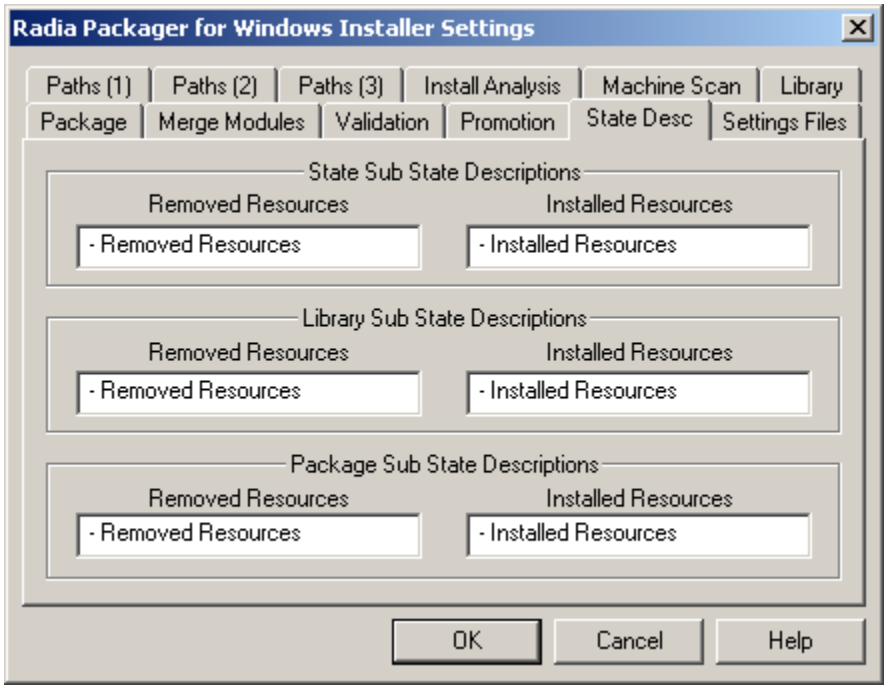

Figure A.13 ~ Radia Packager for Windows Installer Settings – State Desc tab.

The **State Desc** tab contains the descriptions appended to sub-states as they appear when imported into the Radia Configuration Analyzer.

For example, if you had a library description of **My Library** and imported it into the Radia Configuration Analyzer, you would have two sub-states:

- **My Library- Removed Resources,** which lists items that would be removed by the final package.
- **My Library- Installed Resources,** which lists items installed by the final package.

Settings Files Tab

The **Settings Files** tab allows you to save the current, or reload previous, Radia Packager for Windows Installer environment settings. Settings should be saved to a server, but may be saved to the local machine. Once saved, these settings can be reloaded to restore the current settings to a previously saved state.

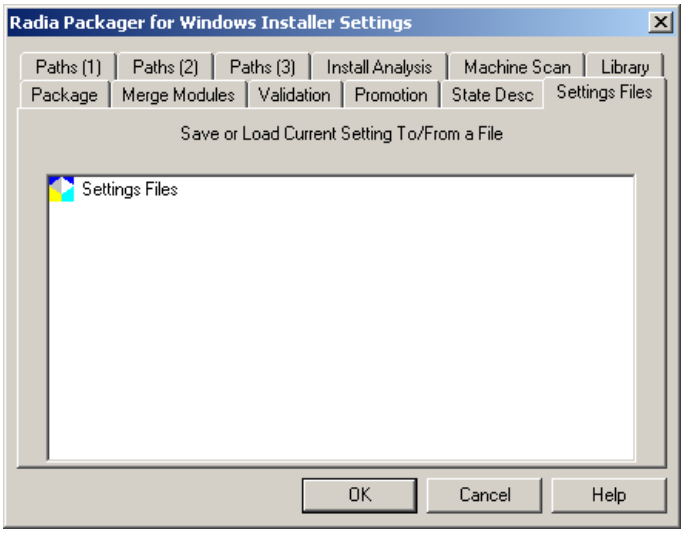

Figure A.14 ~ Radia Packager for Windows Installer Settings – Settings File tab.

The operating settings may be set and saved based on any criteria that meets your packaging needs, such as by corporate division, department, or line of business.

To save settings

1. Right-click the **Settings Files** icon.
2. Select **Save Settings to a File**, and enter a path and name for the settings file (.ini).

To load settings

1. Right-click the **Settings Files** icon or the desired entry from the tree.
2. Select **Load Settings from a File**, and enter a path and name for the settings file (.ini) to load.

To delete settings

1. Right-click an entry from the tree.

2. Select **Delete this Settings File**.

B

Radia Package Wizard Advanced Options

The Radia Packager Wizard includes an **Advanced Options** menu that gives you easy access to the common settings and various types of information placed in your Windows Installer packages, as shown in Figure B.1 on page 248.

These same settings are also available from the **Windows Installer Options** dialog of the Radia Packager for Windows Installer, as shown Figure B.2 on page 248; this dialog is presented at the end of a Unified packaging session.

The only difference between the Advanced Options menu and the Windows Installer Options dialog is the grouping of the options.

The following Appendix describes each of the options available from either menu. However, many of the options can be preset through your package template, thereby removing the need to edit the information each time you reach the **Advanced Options** or **Windows Installer Options** dialogs.

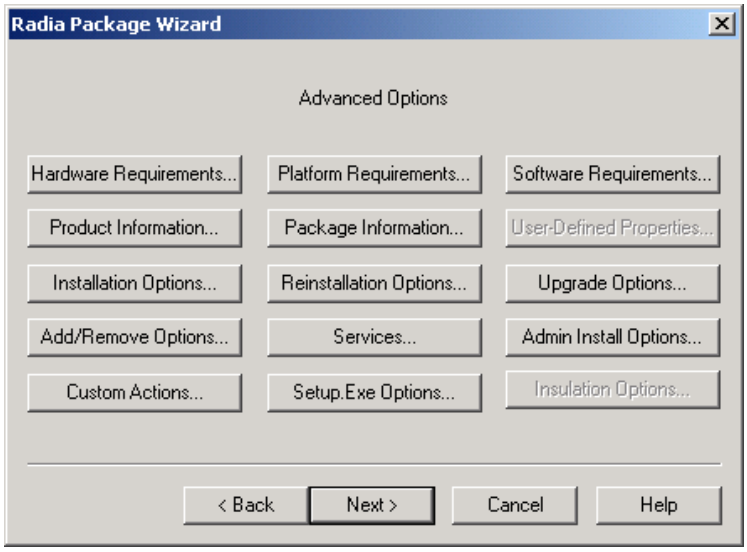

Figure B.1 ~ Radia Package Wizard – Advanced Options.

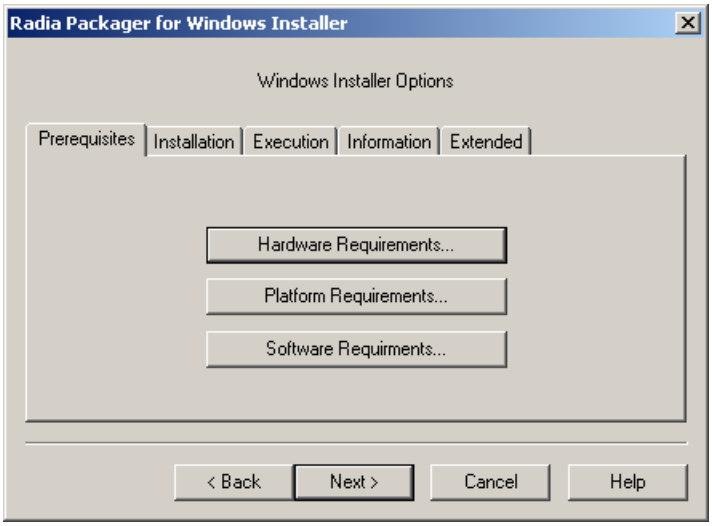

Figure B.2 ~ Radia Packager for Windows Installer, Windows Installer Options.

Table B.1, below, lists the options found under each tab on the Windows Installer Options dialog of the Radia Packager for Windows Installer. To obtain the details on using each option, see the topics that follow.

Table B.1 ~ Windows Installer Options, by Tab Location

Option	Tab Location
Hardware Requirements Platform Requirements Software Requirements	Prerequisites Tab
Add/Remove Options Administrative Install Options Installation Options Reinstallation Options	Installation Tab
Custom Actions Services	Execution Tab
Package Information Product Information User Defined Properties	Information Tab
Insulation Options Setup.Exe Options Upgrade Options	Extended Tab

Hardware Requirements

Hardware requirements set the criteria for the hardware. If the destination machine does not meet these requirements, the package will not download and install to the machine.

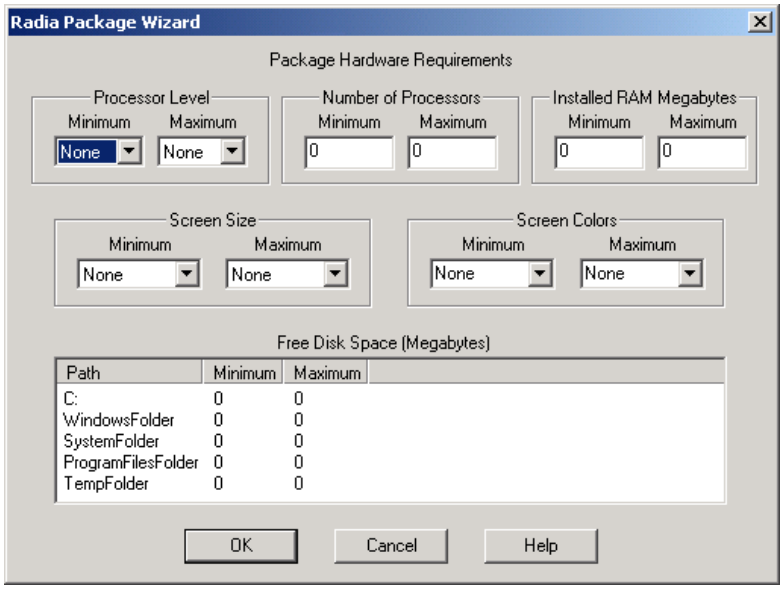

Figure B.3 ~ Radia Package Wizard – Package Hardware Requirements.

Platform Requirements

Platform requirements define the Windows platform version requirements that should be met for the package to install. By default, all packages are set to install only if these platform requirements are met, and the **Minimum Platform Version** (see Figure B.4 on page 251) is set to the platform on which the original application was analyzed.

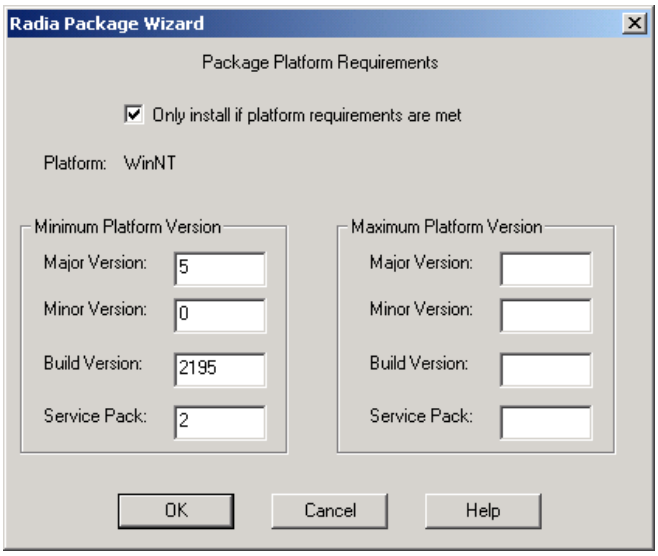

Figure B.4 ~ Radia Package Wizard – Package Platform Requirements.

Software Requirements

Software requirements can be set to verify that certain applications, files, or registry values are, or are not, on the machine before your package is installed.

Figure B.5 ~ Radia Package Wizard – Package Software Requirements.

Product Requirements

You can set product requirements to ensure a Windows Installer package has been previously installed or not previously installed.

To set product requirements

1. On the **Advanced Options** dialog box (see Figure B.1 on page 248), click **Software Requirements.**

   OR

   On the **Windows Installer Options** dialog box (see Figure B.2 on page 248), click the **Prerequisites** tab, and then click **Software Requirements**.

2. On the **Package Software Requirements** dialog box**,** right-click **Product Requirements** and select **Add Product Requirement** from the shortcut menu.

   The **Add Product Requirement** dialog box opens.

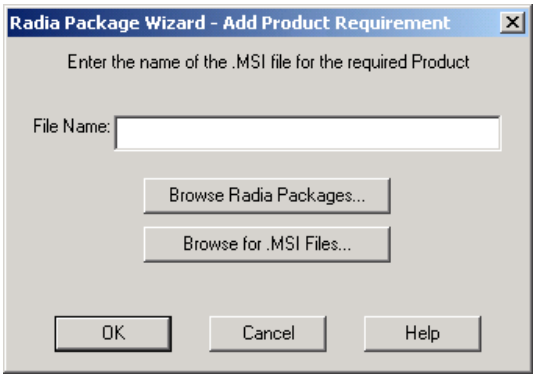

Figure B.6 ~ Radia Package Wizard – Add Product Requirement.

3. Enter the appropriate file name or click one of the **Browse** buttons to manually locate the file.
4. Click **OK.**

   The **Edit Product Requirements** dialog box opens.

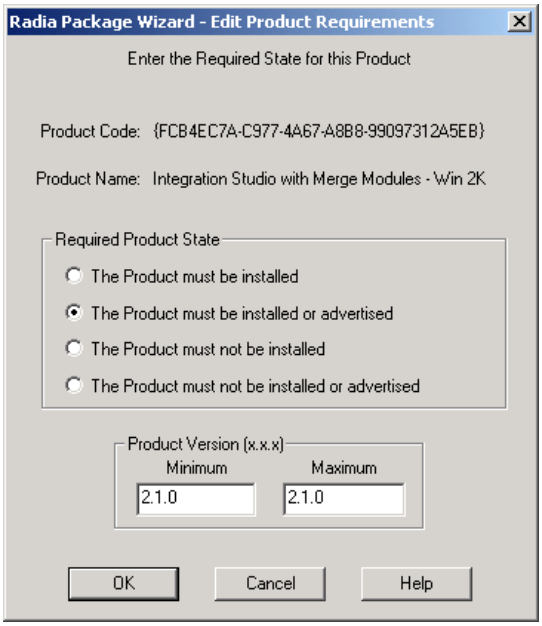

Figure B.7 ~ Radia Package Wizard – Edit Product Requirements.

5. Enter the required state for the product and click **OK** to return to the **Package Software Requirements** dialog box.

File Requirements

You can set the file requirements to verify whether files are or are not on the machine before your package is installed.

To add a file requirement

1. On the **Advanced Options** dialog box (see Figure B.1 on page 248), click **Software Requirements.**

   OR

   On the **Windows Installer Options** dialog box (see Figure B.2 on page 248), click the **Prerequisites** tab, and then click **Software Requirements**.

2. On the **Package Software Requirements** dialog box, right-click **File Requirements** and select **Add File Requirement** from the shortcut menu.

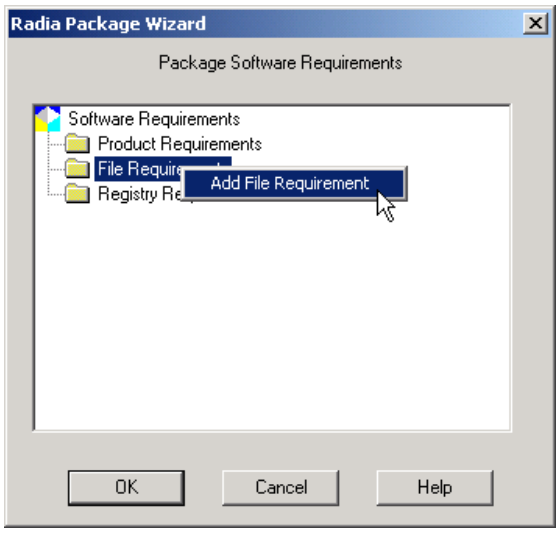

Figure B.8 ~ Radia Package Wizard – Add File Requirement.

The **Add File Requirement** dialog box opens.

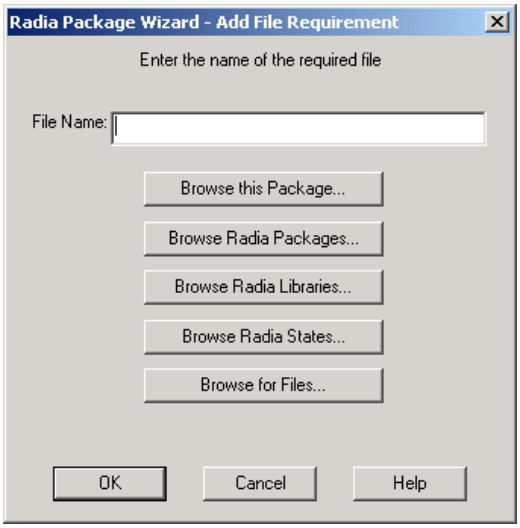

Figure B.9 ~ Radia Package Wizard – Add File Requirement.

3. Enter the name of the file, or click one of the **Browse** buttons to manually locate the file.
4. Click **OK**

   The **Edit File Requirements** dialog box opens.

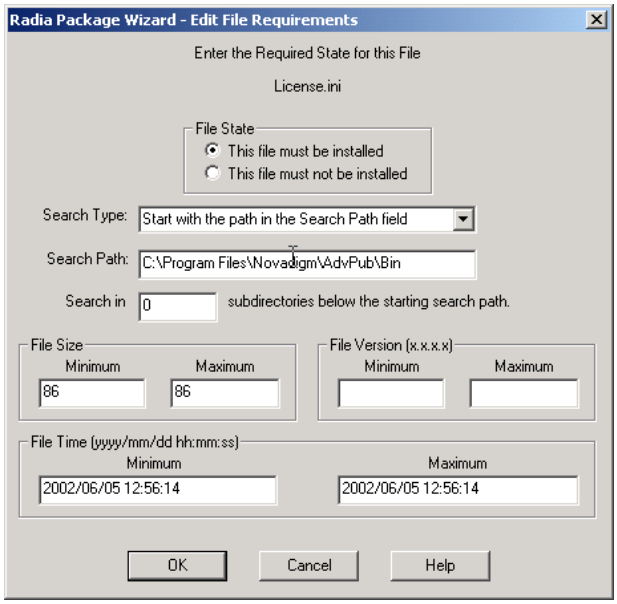

Figure B.10 ~ Radia Package Wizard – Edit File Requirements.

5. Edit the appropriate file requirements, and click **OK** to return to the **Package Software Requirements** dialog box.

Registry Requirements

You can set the registry requirements to verify whether a registry entry is or isn't on the machine before your package is installed.

To add a registry requirement

1. In the **Advanced Options** dialog box (see Figure B.1 on page 248), click **Software Requirements.**

   OR

   On the **Windows Installer Options** dialog box (see Figure B.2 on page 248), click the **Prerequisites** tab, and then click **Software Requirements**.

2. In the **Package Software Requirements** dialog box, right-click **Registry Requirements** and select **Add Registry Requirement** from the shortcut menu.

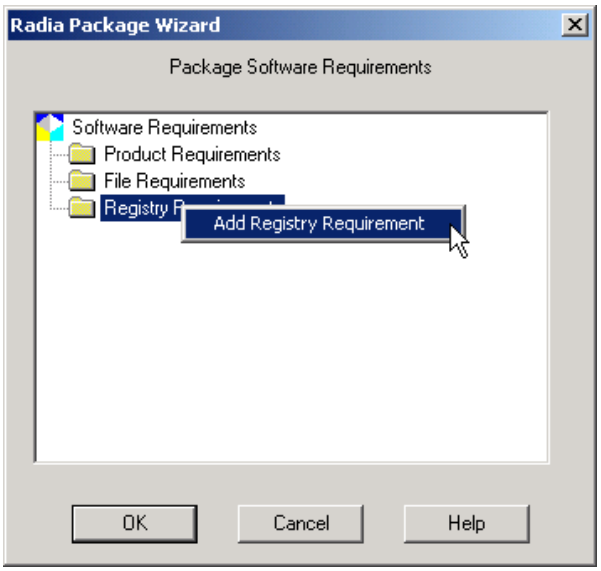

Figure B.11 ~ Radia Package Wizard – Package Software Requirements.

The Add Registry Requirement dialog box opens.

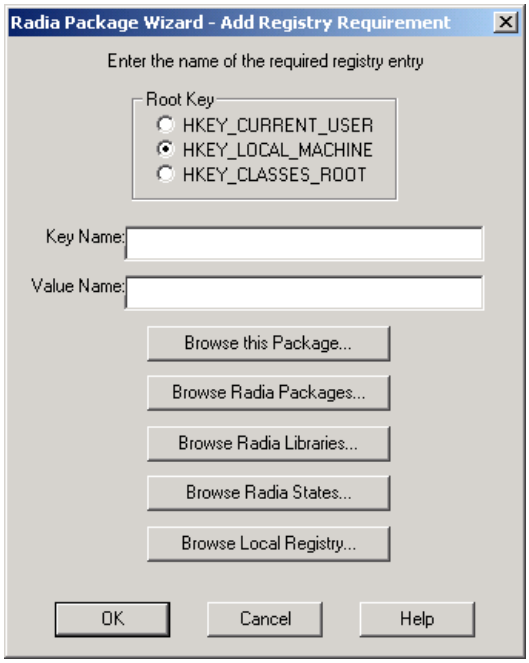

Figure B.12 ~ Radia Package Wizard – Add Registry Requirement.

17. Enter the required registry entry.

- Select the appropriate Root Key.
- Enter the registry Key Name and Value Name, or use one of the **Browse** buttons to manually locate the Key and Value Names.

18. Click **OK.**

The **Edit Registry Requirements** dialog box opens.

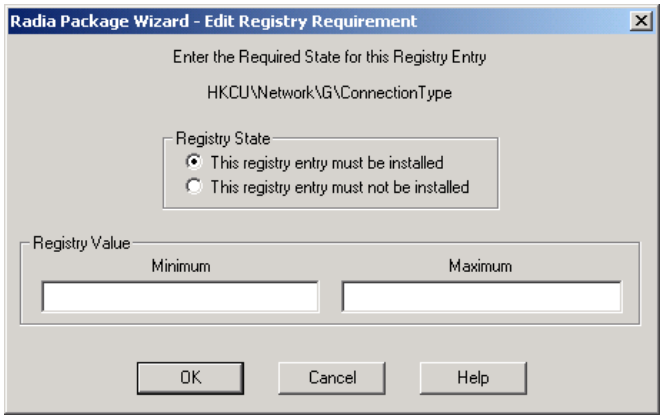

Figure B.13 ~ Radia Package Wizard – Edit Registry Requirement.

19. Edit the appropriate file requirements, and then click **OK** to return to the **Package Software Requirements** dialog box.

Product Information

Product information contains your **Company Name** and a **Description** of the product.

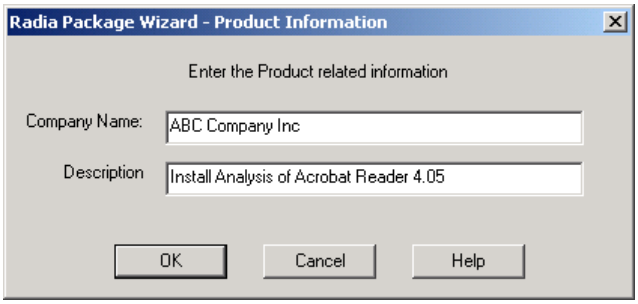

Figure B.14 ~ Radia Package Wizard – Product Information.

Package Information

Package information allows you to enter specific comments about a package. 1033 represents English in the **Language** option.

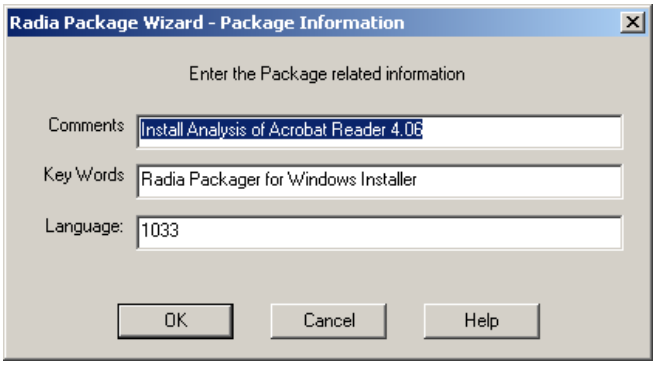

Figure B.15 ~ Radia Package Wizard – Package Information.

User-Defined Properties

User-defined properties is information you want embedded into the MSI database that you may want to change with each package. Any information that remains consistent from package to package should be added to your MSI package template (see *Editing a Package Template* on page 190). Examples of instances where you may want to use user-defined properties:

- Track the packager of an application.
- Enter inventory control numbers into the package.

> **Note**
>
> For this button to be available, select a User Properties .ini file in the **Package** tab of the Radia Packager for Windows Installer Settings dialog box. See *Package Tab* on page 236 for more information.

To modify a user-defined property

1. On the **Advanced Options** dialog box (see Figure B.1 on page 248), right-click **User-Defined Properties**.
2. The **Set user-defined property values** dialog box opens.

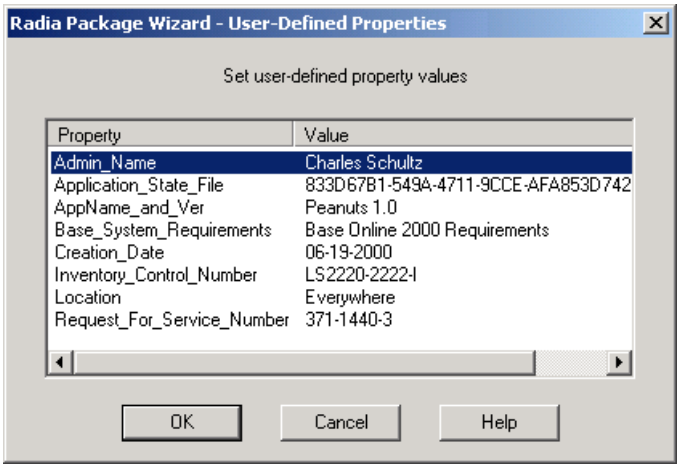

Figure B.16 ~ Radia Package Wizard – User-Defined Properties.

3. Right-click the property you would like to modify, and select **Edit** from the shortcut menu.

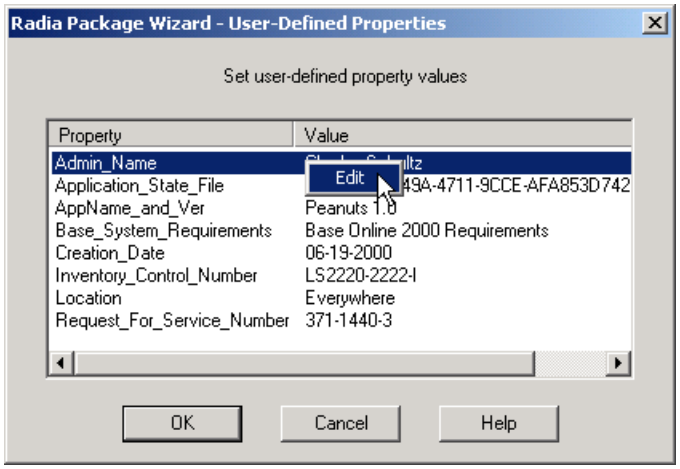

Figure B.17 ~ Radia Package Wizard – User-Defined Properties.

The **Radia Package Wizard – Property Value Edit** dialog box opens.

4. In the **Value** text box type a new value.

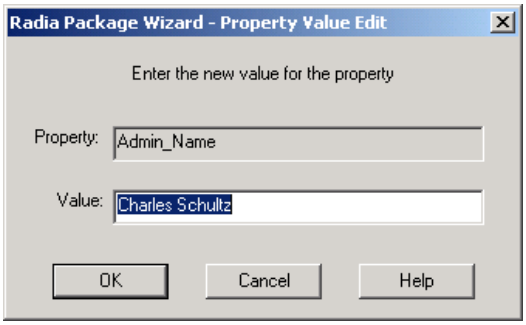

Figure B.18 ~ Radia Package Wizard – Property Value Edit.

5. Click **OK** to return to the **User-Defined Properties** dialog box.
6. Click **OK** to return to the **Advanced Options** dialog box.

Installation Options

Installation Options allow you to modify how your application is installed on the target machine. If you wish your application to be installed with only a progress bar (and no other interface), and your users to install by double-clicking the MSI, select **Set Default User Interface to Basic**.

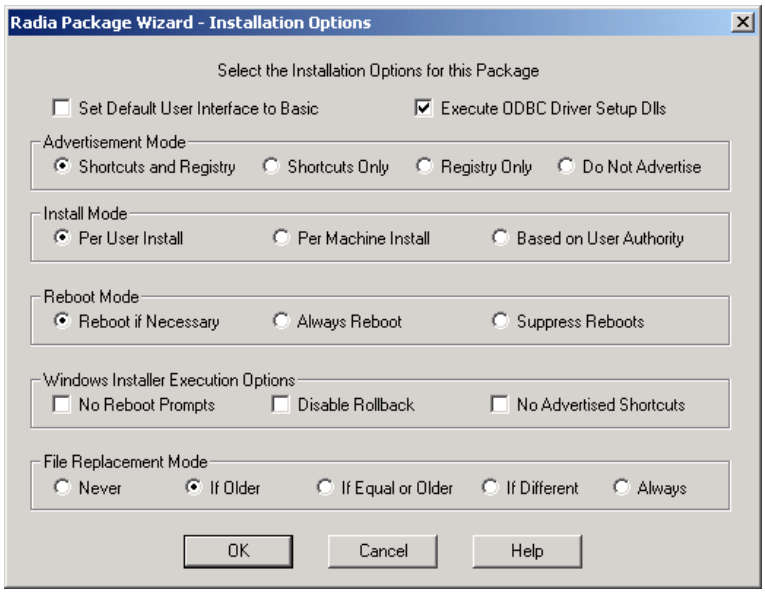

Figure B.19 ~ Radia Package Wizard – Installation Options.

If your platform of choice is Windows NT, Windows 2000, or Windows XP, then you can choose to install the application in one of three ways:

- **Per User** (only the user who installs the application will have access to it).
- **Per Machine** (all users will have access to the application)
- **Based on User Authority** (if the installing user is an administrator, the application will be installed Per Machine, otherwise the application will be installed Per User).

There are multiple options available in the **Installation Options.**

- Select the **Reboot Mode** you would like your Windows Installer package to use on installation.

Set options for the execution of your Windows Installer package:

- **File Replacement Mode** allows you to set the criteria Windows Installer use to determine if a file that already exists on the machine should be overwritten when installing the package. You can overwrite an existing file:
  - never,
  - if the existing file is an older version,
  - if the existing file is the same version or older than the one in the package,
  - if the existing file is simply a different version than the file on the machine,
  - or you can always replace the file.

Reinstallation Options

Review and modify any reinstallation options to be used if a reinstallation of the product is required and performed on the client machine by Windows Installer.

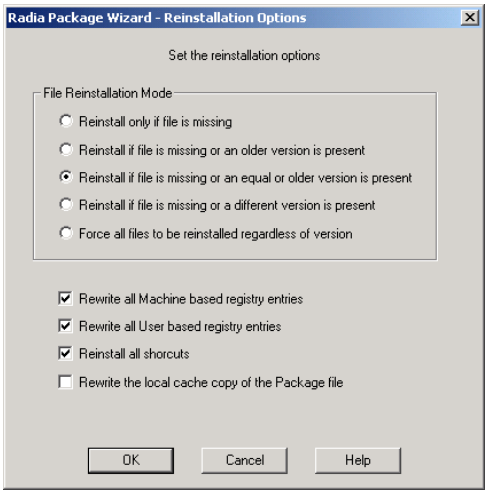

Figure B.20 ~ Radia Package Wizard – Reinstallation Options.

File Reinstallation Mode determines when files are reinstalled. You can choose to reinstall:

- only if the file is missing,
- if the file is missing or the existing version is older than the version in the package,
- if the file is missing or the existing version is older or the same version as the version in the package,
- if the file is missing or the version on the machine is different than the version in the package,
- or you can always reinstall the files.

Upgrade Options

Upgrade Options allow you to enter Upgrade Codes for packages to be removed before new packages are installed.

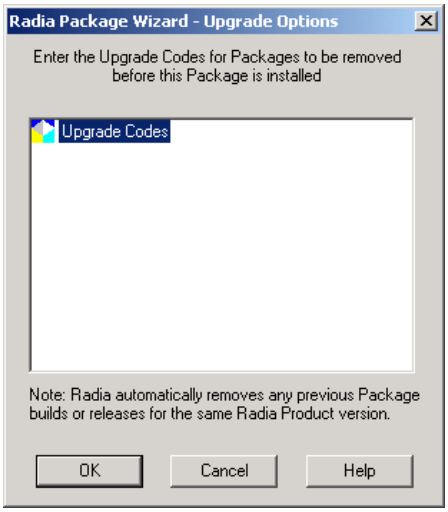

Figure B.21 ~ Radia Package Wizard – Upgrade Options.

Use **Upgrade Options** to define previous versions of packages that will be upgraded automatically when the package is installed. Because versions and upgrades are maintained within the product once it is created, this option is only available when creating a new product.

Note

An Upgrade Code is a GUID identifier within the MSI that defines what packages can be used to upgrade one another. It is used by Windows Installer in conjunction with the ProductVersion variable. Take three (simplified) packages:

pkg 1	UpgradeCode=123	ProductVersion=1.1.1
pkg 2	UpgradeCode=123	ProductVersion=1.2.0
pkg 3	UpgradeCode=234	ProductVersion=1.2.0

pkg2 will upgrade pkg1 because the UpgradeCode is the same and the ProductVersion is higher. pkg3 will not upgrade either, because its UpgradeCode is different. pkg1 will not update the others because they both have a higher ProductVersion (and pkg 3 has a different UpgradeCode).

- Right-click **Upgrade Codes** to add a new Upgrade Code.

Add/Remove Options

Add/Remove Options allow you to adjust how your package interacts with Windows 2000 or later operating system's improved Add/Remove Programs Control Panel Applet. In this Advanced Option, you can set whether the application will be displayed in the list of applications in the Control Panel, and whether to allow the application to be repaired, modified, or removed from the Control Panel. You can also set help contact, phone, and Web address information for the application in the Control Panel.

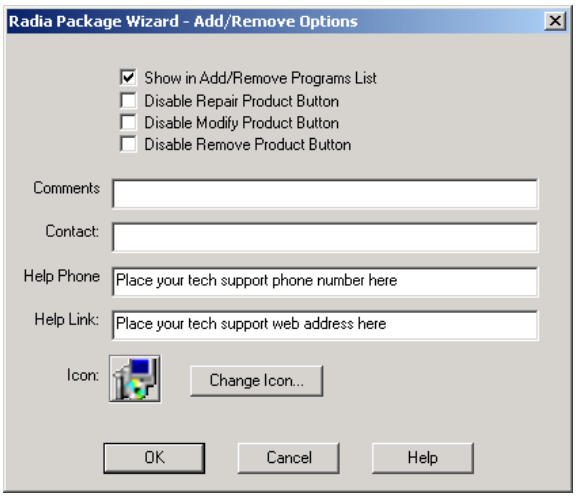

Figure B.22 ~ Radia Package Wizard – Add/Remove Options.

Services

The Services Option allows you to perform different service related tasks with your installation. If you had to stop a service on installation, you would add that option here. Any services installed with the application would be displayed here as well.

To add a service

1. Open the **Services** dialog box.

   Access **Services** from the Radia Package Wizard Advanced Options dialog or from the **Execution** tab of the Windows Installer Options dialog of the Radia Packager for Windows Installer.

2. Right-click **Services,** and select **Add a service** from the shortcut menu.

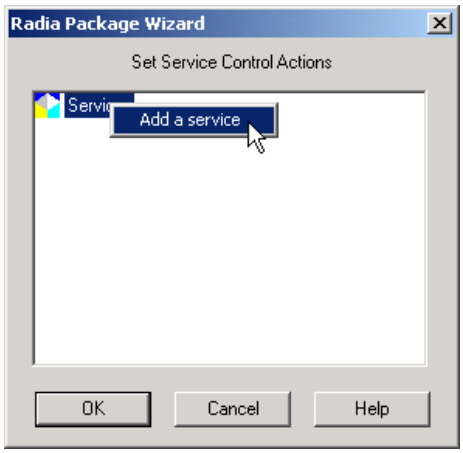

Figure B.23 ~ Radia Package Wizard – Add a service.

3. Enter the service name and click **OK.**

   The **Service Control Actions** dialog box opens.

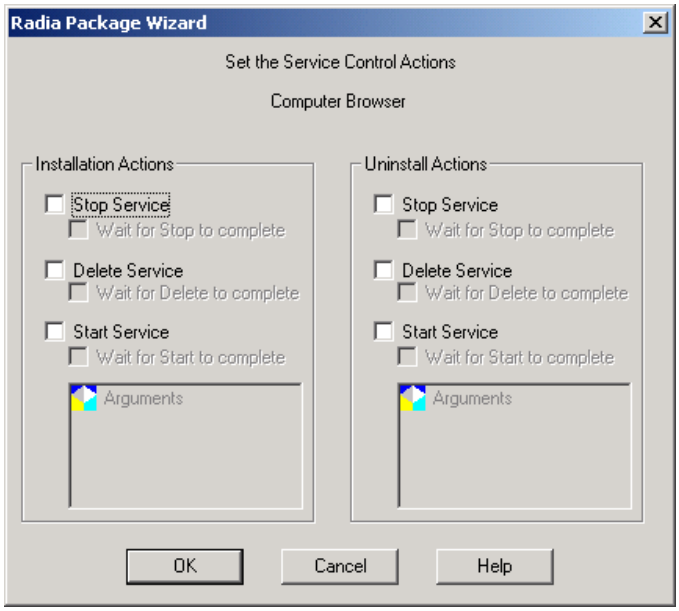

Figure B.24 ~ Radia Package Wizard – Set service control actions.

4. Select the appropriate installation and un-installation actions for the service and click **OK.**
5. Click **OK** to return to the **Advanced Options** dialog box or the **Windows Installer Options** dialog box.

Admin Install Options

Admin Install Options let you set information that you may want to differ if the application is installed from an Administrative Installation.

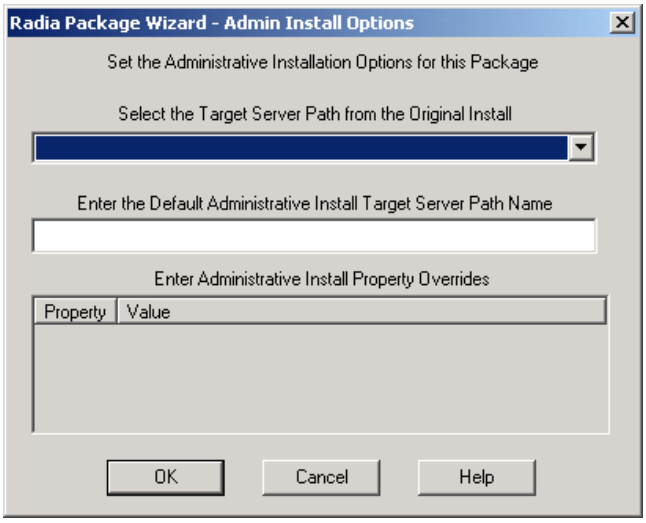

Figure B.25 ~ Radia Package Wizard – Admin Install Options.

Custom Actions

Custom Actions may be used during the installation process to provide functionality that is outside the scope of the standard Radia Packager for Windows Installer and Windows Installer functionality.

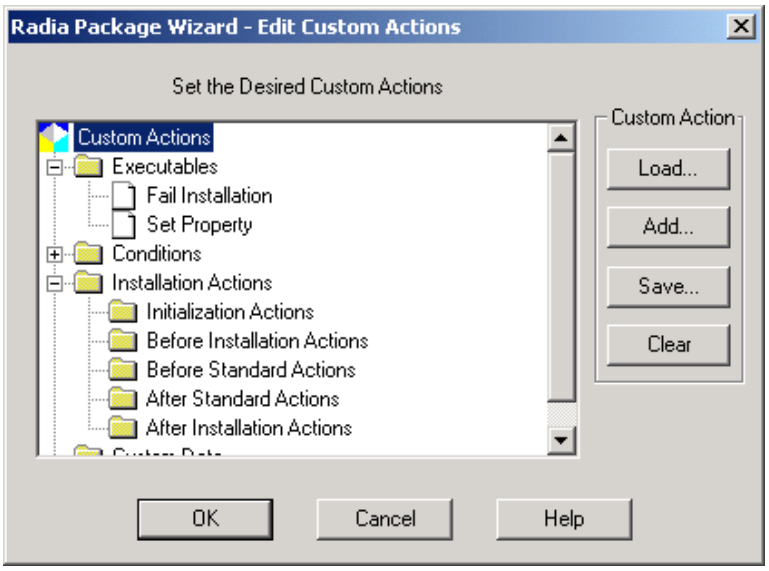

Figure B.26 ~ Radia Package Wizard – Edit Custom Actions – Set the desired custom actions.

Custom Actions allow you to expand the functionality of Windows Installer package processing to include your own custom written code. For example, you can use Custom Actions to extend the user interface, edit install components prior to installation, validate the target install environment, or for a myriad of other functions.

You can execute custom actions at each of the following Windows Installer points:

- Initialization Actions
- Before Installations Actions
- Before Standard Actions
- After Standard Actions
- After Installation Actions

Radia Packager for Windows Installer packaging treats custom actions as reusable components, which allows the administrator to create and save a related set of custom actions and then imbed these custom actions into any number of installation packages. Therefore, custom action sets

should be created with functionally related actions. One or more custom action sets may be added to the installation package.

Windows Installer supports the following Custom Action file types:

- Visual Basic Scripts
- Java Scripts
- EXE (Executables)
- DLLs
- .MSIs

Creating a Custom Action

Custom Action executable code is created by an administrator, or developed using one of the supported executable file types: VB Script, Java Script, EXE, DLL, or MSI. As such, each custom action is associated with the appropriate executable file type.

Use the Radia Package Wizard's **Manage Custom Actions** option to create, edit, and delete Custom Action sets or files. The wizard steps you through the process of loading executables, setting conditions, adding actions, and accessing custom data.

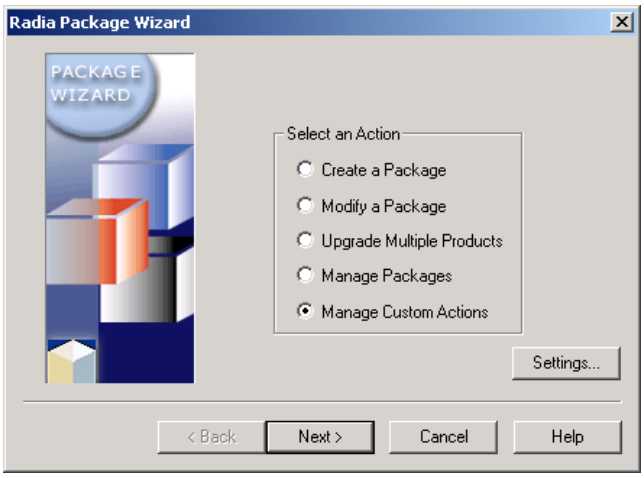

Figure B.27 ~ Radia Package Wizard – Select an Action.

Your set of custom actions are saved as a Custom Action State File (*.ISState). Included in this state file are the actual executables as well as any custom data binary files. Therefore, each time a custom action executable is modified, it must be imported into the Custom Action state file again, so it can be embedded into Windows Installer packages.

When setting the Advanced Options for a package, you can use the Custom Actions dialog box to apply the custom actions to the package, as necessary.

Setup.Exe Options

Radia Packager for Windows Installer provides you with a Setup.exe with every package you create. This executable will start Windows Installer with the options you provide. To set the options, use the **Setup.Exe Options** button in the **Advanced Options** dialog box of the Radia Package Wizard. Alternatively, use the **Setup**.Exe Options button on the **Extended** tab of the Windows Installer Options dialog of the Radia Packager for Windows Installer.

- **User Interface Level** area
  Radia Packager for Windows Installer allows you to easily set the user interface level on the install to anything from completely silent (None) so the user doesn't even know the application is being installed (even if there are user interface dialog boxes to respond to) to Full, with which the user is always given dialog boxes to respond to, even if there is no user interface to the installation.
- **Logging Options** area
  You can also modify the logging options that are run when you install your application using Setup.exe.

Setup.exe uses the information in the file Setup.ini, also included with the package. As with any INI file, Setup.ini can be modified by hand if you wish to run transforms or run other Windows Installer options.

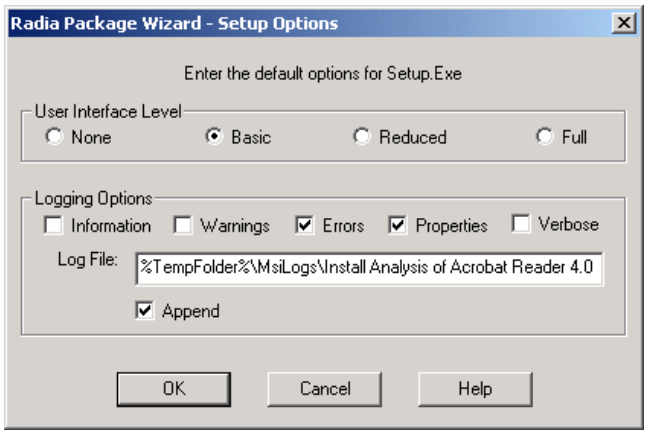

Figure B.28 ~ Radia Package Wizard – Setup Options.

After making changes to the Advanced Options, be sure to finish the Radia Package Wizard to apply the changes to your package.

C

EDM-Radia Compatibility Mode for the Radia Configuration Analyzer

State File Generator (IsState)

A state file (.ISState) contains metadata of the components of a Radia package or EDM service, as it would look on the desktop after the Client Connect process. This is done by starting at the specified Radia package or EDM service, reading its components from the Radia Configuration Server in real-time, and resolving any symbolics that replace standard paths during the packaging process. The symbolics include registry keys/values and (Radia/EDM) object/variables. The State File Generator batch mode process connects to the Radia Configuration Server as an administrator to read the component classes, instances and resource data of the Radia package and chained Radia packages or EDM resources to get the component information to generate the state file container.

Setting Symbolic Variable Values

The Radia Client performs symbolic variable substitution for well known logical directories on the Windows computer during installation of resources. The following logical directories are symbolically processed based on settings in the following registry key on the local machine:

HKLM \ Software \ Microsoft \ Windows \ CurrentVersion \ Explorer \ ShellFolders

with values:

- Common Startup
- Common Desktop
- Common Programs
- Common Start Menu
- Startup
- Desktop
- Programs
- Start Menu

During the packaging process, if shortcuts are discovered to have the same data as the key/value for the items noted above, those data strings are replaced with a Radia tag for the data value. The tags are:

- $COMMON STARTUP$
- $COMMON DESKTOP$
- $COMMON PROGRAMS$
- $COMMON START MENU$
- $STARTUP$
- $DESKTOP$
- $PROGRAMS$
- $START MENU$
- $NTUSER$
- $ALLUSERS$

Radia/EDM object variables are in the form of *&(object.variable)* or *&(variable)*. In the second case, the object name is defaulted to ZMASTER. Since the State File Generator would usually run on the administrator machine, it has access to the current registry values on that machine and the object stored in EDM/IDMLIB, namely ZMASTER. This symbolic substitution must be honored when creating state files also. Since the package associated with a state file is resolved at an instance in time, the current registry and (Radia) object values of the administrator might not be the correct values with which to replace the symbolics.

The State File Generator supports a schema to override the current machine values. This is done with an INI file. Thus, the default behavior is not to use the local machine's registry values or object in IDMLIB, but the value stored in the noted INI file. This can be altered with a runtime option.

The format of the INI file section names are the object names; [ZMASTER] would be the pseudo value for ZMASTER and the values in the section are attributes and values.

There is a special section, [REGISTRY] which is the override values for the registry items noted above.

Note

If the State File Generator cannot find the value for a symbolic replacement value in the INI file, the process will stop and the INI file must be updated with the associated symbolic value.

If *UseLocalValues* was specified, then it will then look in the registry and EDM/ IDMLIB for objects and their values. If the value still can't be found, the process stops, unless the State File Generator output type is set to *discover*. When running in *discover* mode, a random value is generated for the unknown value to permit the process to continue. The contents of the INI file may look similar to Figure C.1 (the case for the section names and the keys —left of the equal sign) is case-neutral. The values are case-sensitive.

```
[registry]
$Common   Startup$            =   C:\Windows\Profiles\All   Users\Start
Menu\Programs\Startup
$Common Desktop$    = C:\Windows\Profiles\All Users\Desktop
$Common Programs$   = C:\Windows\Profiles\All Users\Start Menu\Programs
$Common Start Menu$ = C:\Windows\Profiles\All Users\Start Menu
$Startup$                          =  C:\Windows\Profiles\JoeUser\Start
Menu\Programs\Startup
$Desktop$           = C:\Windows\Profiles\JoeUser\Desktop
$Programs$          = C:\Windows\Profiles\JoeUser\Start Menu\Programs
$Start Menu$        = C:\Windows\Profiles\JoeUser\Start Menu
[zmaster]
zuserid   = JoeUser
localuid  = JoeUser
zwindrv   = C:
zwindir   = \Windows\
zdrive    = S:
zmsysdir  = \radia\sys
zmlibdir  = \radia\lib
zos       = WIN2K
zsysdrv   = c:
zsysdir   = \novadigm\
Zdomname  = MyDomain
Zmgrname  = ThisMgr
[desktop]
zobjid    = 00000000000
zobjpcls  = dummy
[dummy]
zobjname  = dummy1
```

Figure C.1 ~ Contents of sample INI file.

Output Types

The output format is controlled by the **Type=** option. When **Type=State** is specified, the output state file is written to the file named in the **Filename=** keyword. With the addition of the **Mode=** keyword, the output can be directed to a special directory as described by a registry value:

HKLM\Software\Novadigm\Radia\Settings\ApplicationAnalyzerExportPath

The output modes are **autoname** and **export**.

> **Note**
>
> **Type=Package** will force export mode.

In both **autoname** and **export**, the output state file name of the state files are controlled by the State File Generator and are used to define the state file output formats that are compatible with the Radia Knowledge Base Manager automated import process. **Type=Package** is used exclusively for generating Radia packages. For more information on the Knowledge Base Manager refer to the topic *The Knowledge Base Manager*.

The State File Generator (IsState) is a command line driven utility. All input is specified by keywords on the command line in the form **keyword=value keyword=value keyword=value**. In addition to the keywords, there are control switches which are in the form of -x or =X. The common switches are -Uuserid for the Radia userid, and -Ppassword for the password.

> **Note**
>
> There are no blanks after the -U and the text of the userid.

Keywords

The valid keywords for IsState are:

- Argsdump
- Description
- DiscoverINI
- EDMServiceName
- Filename
- Inifile
- Logonly
- MaxErrors
- Mode
- OS

- Packagename
- Replace
- Saveedr
- Showkeys
- Type
- UseLocalValues
- Validate

The keywords can be abbreviated to the least number of characters that makes the keyword unique. Thus **PackageName** can be shortened to **P=,** but **SaveEDR** would be specified as **Sa=. ShowKeys** would be shortened to **Sh=.** The keywords are case neutral, but their values are case sensitive. If the value for the keyword contains blanks or special characters, then use quotation marks around the keyword and value; "D=This is my state file", where **D=** is short for **Description**. The Keyword=Value pairs are separated by one or more blanks on the command line, IE K1=V1 K2=V2 K3=V3.... Kn=Vn. Table C.1 below, contains a description of the keywords and their values.

Table C.1 ~ IsState Keywords

Keyword	Values	Description
Argsdump	Yes No True False 1 0	Used for debugging. Argsdump shows the values being passed into the State Create API. Argsdump=Yes
Description	Descriptive text	State file description text that describes its content and is viewed by the state analyzer. If not specified, a string with the package name is used as the description. When **Type=Package** is specified, the description is the value saved in the DESCRIPT attribute of the PACKAGE instance.
DiscoverINI	*Output INIFile.INI*	When running with **type=discover**, all the symbolics used (registry / objects), will be written to this INI file, including the random values generated for the unknowns. The contents of this INI file can be edited with new values replacing the randomly generated values. The values are in the form %<digits>%. The number of digits is 12 and the first digit is 0. If the file name is not fully qualified, the current directory is used.
EDMServiceName	*ServiceName*	Name of the EDM service to process. If only the instance name is specified, then the default location is **Primary.SystemX.ZService**. To specify a different domain, type the full FDCI.

Table C.1 ~ IsState Keywords

Keyword	Values	Description
FileName	*MyState.ISState*	File name of the state file to create. If omitted, the state file name is the same as the package (instance) name. If the name is not fully qualified, then the current directory is used. When **Mode=Export** or **Mode=Autoname** is specified, the State File Generator controls the name and location of the state file using the value of a registry key for folder name.
INIFile	*MyINI.INI*	The name of the INI file of default symbolic values to use. The default name is ISState.INI, and is looked for in the current directory. When **Mode=Export** is specified, the location of the inifile is controlled by the *Radia registry value* ***ApplicationAnalyzer ExportPath***. This registry value points to a directory structure that is a series of subdirectories, and the INI files are stored in the ***Varsets*** subdirectory. If there is no path specified on the inifile name, then the default path is located by the value of *ApplicationAnalyzerExportPath* and *Varsets|ISState.INI.* If a path is specified, that value will be used, regardless of the *Mode* value.
LogOnly	Yes No True False 1 0	Logonly will redirect most of the console output to the log file. LogOnly=Yes
MaxErrors	*Number*	The number of download errors that will be tolerated if this value is exceed, the process stops. MaxErrors=5

Table C.1 ~ IsState Keywords

Keyword	Values	Description

Mode	Autoname [: StateFilePath]	In autoname mode the name of the state file is changed to reflect the package name and the INI file used to create the state file. The state file is placed in is the directory specified after **autoname:**. In addition to the created state file, a copy of the INI file used is also placed in the same directory. If no path is specified, then the current directory is used.
		The name of the **INI/state file** pair comes from the package **FDCI**. Since the **F** part is always **PRIMARY,** and the **C** part is always **PACKAGE** (ZSERVICE), just the domain and instance part of the name will be used to build the (file system) name of the state and INI file. Periods (.) are usually used to annotate an FDCI name, but in the case of the state/INI names, the periods are replaced with a hyphen (-). In addition to the domain and instance name, the package timestamp and the CRC of the INI file contents are added to the name. Lastly, if required, the operating system for which the state file was generated is added. If the state operating system is neutral, then the name ends with the CRC. The operating systems can be Windows 9x, NT, 2000, or XP. An example name would appear as:
		`domain-instance-date-hour-minute-second-crc-os. [ INI | ISState ]`
		For example, the package name
		`PRIMARY.SOFTWARE.PACKAGE.C_COMMUNICATOR_476_W2K`
		would generate the following two files:
		`software-c_communicator_476_w2k-20020213-17-33-21-4a32ed92-win2k.INI`
		`software-c_communicator_476_w2k-20020213-17-33-21-4a32ed92-win2k.ISState`
		If the state file is placed into an Automated Import Path, then the Knowledge Base Manager will automatically import the state file into the Application Knowledge Base.

Table C.1 ~ IsState Keywords

Keyword	Values	Description
	Export	The location of the state files is controlled by the value of the Radia registry value: `HKLM\Software\Novadigm\Radia\Settings\ApplicationAnalyzerExportPath` This registry value points to a directory structure that is a series of subdirectories. In addition to the **VarSets** subdirectory, there are **Package** and **Service**, subdirectories. If the state file is placed into an Automated Import Path, then the Knowledge Base Manager will automatically import the state file into the Application Knowledge Base.
OS	*OS Name*	This keyword permits the operating system name (i.e. WIN95, WIN98, NT, WIN2K, WINXP) to be preloaded and thus override the value stored in ZMASTER.ZOS.
PackageName	*PackageName*	The nName of the package to process. If only the package instance name is specified, then the default location is **Primary.Software.Package**. To specified a different domain. Specify the full FDCI. Packages can only be in a class named package.
Replace	Yes No True False 1 0	Replace controls if the generated output is replaced in the directory. If set to *True* or *Yes*, the state file is always replaced. When *Type=Package* and *Replace=False* is specified, the output is replaced if the package stored on RCS is newer than the existing one in the output area.
SaveEDR	Yes No True False 1 0	This is mainly for debugging. When specified, the EDR file that contains the registry keys is save in the TEMP directory. The State File Generator looks for a working temp directory in the environment variable "TEMP". The files with the extension "*.before" are the EDRs fetched from RCS and the files name "*.after" are the values after the symbolics are resolved.
ShowKeys	Yes No True False 1 0	When processing the EDR files which contain the registry keys, the active keys are not echoed to the screen. When set to true, all keys are echoed to the screen. The inactive and skipped keys are always echoed to the screen.

Table C.1 ~ IsState Keywords

Keyword	Values	Description
Type	S[tate] D[iscover] P[ackage]	Specifies the type of output: State, Discover, or Package (S, D or P, respectively). If **Type** is omitted, then a dump is generated which will annotate all the components of the package. **Type=Package** will generate a state file, but will force **Mode=Export**, which will place the state files in the directory specified by the Radia registry value. **HKLM\Software\Novadigm\Radia\Settings\ ApplicationAnalyzerExportPath.** With *Type=Discover,* ISState will continue to read the components of the package even if unknown symbolics are discovered. When an unknown symbolic is found, a random value in the form of **%<digits>%** is used in its place. The number of digits in this value is 12 and the first digit is 0. At the end of the process, all the values for the symbolics are written to the console, and if the *DiscoverINI* key was specified, then they are written there too.
Validate	Yes No True False 1 0	This will validate the resource data fetched from the manager. The first validation test is the size. The size of the resource as noted in ZRSCSIZE must be equal to the size of the resource when it's fetched (and exploded if compressed) from RCS. Then if a CRC of the data is stored in the instance, a CRC is computed for the data just fetched and the two values are compared. Lastly, if an MD5 signature exists, that is also checked in the same manor as the CRC was checked. Internal resource, like EDR files and Radia objects are always checked regardless of the value of **Validate.**

Glossary

Application Extension Framework (AXF)

AXF is an extendible framework that resides on each workstation or server to monitor, report on, and control selected applications.

Application Extensions

Application Extensions are loaded by the Application Extension Framework to control some aspect of application processing. Application Extensions may be simple or complex in nature and enhance the ability to manage the application, such as protecting an application from corruption.

AXF Log File

The Application Extension Framework maintains a log of all extensions that were loaded for each launch of an application. This is basically an audit file containing a history of the extensions and any extraordinary events produced for the Radia Packager for Windows Installer protected applications.

Comma-Separated-Value (.CSV) Formatted Files

CSV files are data files that have each record value separated by a comma. CSV files are created by Radia Packager for Windows Installer from a state file so the state file information can be used by other software applications.

configuration set

Contains the configuration information for an Application Extension. Each Configuration Set is given a user-allocated unique name.

custom actions

A type of state file that contains customized application processing to extend the functionality of the Windows Installer package. Custom action files include Installation Actions that execute a VBScript, Java Script, EXE, or DLL based on your defined conditions, and may use Custom Data.

delta state file

A type of state file that contains a definitive list of the differences between any two state files that are compared.

filter state file

A type of state file that contains the filtering criteria used during a machine scan or other state file build process.

impact state file

A type of state file that contains all resources changed by an application program during its execution.

install state file

A type of delta state file that combines a Machine Scan Delta State and a Radia Impact State to achieve the best of both techniques. Install states identify the resources needed for an install.

intersection state file

A type of state file that contains a definitive list of all the resources shared by any two state files.

package difference transform

A package difference transform creates a transform containing the differences between two package files. The packages must not have any file, registry, or shortcut differences.

packaging

The act of identifying and gathering the components of an application.

permission state file

A type of state file that contains a set of Access Control Groups, each of which has been assigned one or more group or user types with corresponding rights. Library resources can be assigned to the permissions of a specific Access Control Group. The assigned permissions are implemented when the Windows Installer package is installed, thus providing control over user access authority at installation time.

profile state file

A type of state file that contains a definitive list of all resources needed by an application program to execute properly.

project tree

The diagram showing the hierarchy of Radia' workspaces and projects, which is displayed when IStudio is executed.

promoting

When you **promote** a package that was created with the Radia Publisher, you are storing the package in the Radia Database.

publishing

The act of promoting the result of packaging to the Radia Database.

Radia Application Knowledge Base (AKB)

A historical database of application and machine state information.

Radia Configuration Analyzer

A console for viewing, storing, and comparing application data. Backed by an SQL or Oracle database, the Radia Configuration Analyzer allows you to import state files created by the Radia State Wizard or Radia Package Wizard of the Radia Packager for Windows Installer. The Radia Configuration Analyzer allows you to compare imported application information for conflicts before creating packages for deployment, and to compare these packages with applications that have been deployed.

Radia Configuration Analyzer Project

A Radia Configuration Analyzer project is a logical grouping comprised of the application and machine state files that have been exported to the Radia Application Knowledge Base.

Radia Configuration Analyzer Workspace

A Radia Configuration Analyzer workspace is the user's view into the Radia Application Knowledge Base based on the user's authority. Workspaces contain projects.

Radia Insulation Wizard

A wizard that guides the user in insulating Radia Libraries and Applications.

Radia Library Wizard

A wizard that guides the user in building Radia libraries.

Radia Package Wizard

A wizard that builds Windows Installer package files (.MSI) and patch package files (.MSP).

Radia Packager for Windows Installer (Process) Menu

A single menu that offers a process view as well as quick access to the Radia Default Settings dialog box, all Radia Wizards, and the Radia Configuration Analyzer.

Radia State Wizard

A wizard that guides the user in the creation of different state file types.

Radia/Insulate

This component of Radia insulates, protects, and ensures application reliability by providing a "wall" around an application, insulating and protecting it from other applications.

scan state file

A type of state file that contains selected directory, file, and registry information created by scanning a workstation.

self-registration

A task that adds a registry setting that indicates where to load the modules required for the component to execute properly.

state file

A highly-tuned file format that is used to store information about an application or workstation at a particular point in time. There are several different types of state files.

transform files

A user interface transform executes the package installation's user interface and saves all responses in a transform file. Although the product appears to be installed, it is not. A package difference transform creates a transform containing the differences between two package files. The packages must not have any file, registry, or shortcut differences.

union interface transform file

A user interface transform executes the package installation's user interface and saves all responses in a transform file. Although the product appears to be installed, it is not.

union state file

A type of state file that contains a definitive list of all the resources contained by adding each of the resources contained in any two state files.

ZSTOP

A ZSTOP expression is used to stop the resolution of an instance based on certain criteria. For example, create a ZSTOP expression to deploy a ZSERVICE instance only to client computers with a particular operations system.

Lists

Figures

Tables

Procedures

Index

A

B

C

D

E

F

H

I

K

L

M

N

O

P

R

S

T

U

V

W

Z