Packaging Guide

PRISM DEPLOY®

Version 13

Prism Suite® Packaging Guide published June, 2012

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. New Boundary Technologies may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

Trademarks

The following trademarks apply to this volume:

NEW BOUNDARY TECHNOLOGIES, the New Boundary Technologies logo, Prism Suite, Prism Patch Manager, and Prism Asset Manager are registered trademarks of New Boundary Technologies, Inc.

LANOVATION, Smart Update, Prism Deploy, and the Prism Deploy logo are trademarks of New Boundary Technologies, Inc.

Express Inventory technologies are © Express Metrix, LLC 2012

Microsoft and Active Directory are registered trademarks of the Microsoft Corporation.

Windows, Windows NT, Windows 2000, Windows XP, Windows Vista, and Windows 7 are trademarks of the Microsoft Corporation.

All other products and companies are trademarks or registered trademarks of their respective companies.

Patent

Prism Suite is protected under US patent number 7,707,571, 7,568,018, and 6,564,369.

Additional Notes

Unless otherwise noted, all names of companies, products, and persons contained herein are part of a completely fictitious scenario or scenarios and are designed solely to document the use of the product.

New Boundary Technologies, Inc.
1300 Godward Street N.E. Suite 3100
Minneapolis, MN 55413

Phone (toll free): 800-747-4487
Phone (local): 612-379-3805
Fax (local): 612-378-3818

URL: *www.newboundary.com*

6-13-2007

Table Of Contents

Packaging Changes with the Editor

Welcome to the Editor

Prism Deploy® Editor is the program for building deployment packages. Prism Deploy saves you a great deal of time by allowing you to capture software changes into a Package that can be distributed to computers across your organization to make those same changes, without requiring you to visit each computer to manually install or update software.

Prism Deploy Editor builds Prism files (Packages, Executable Packages, and Pictures). These files are the basis for Prism Deploy Tasks that are distributed through the Prism Console.

Unlike complex scripting tools, the Prism Deploy Editor is amazingly easy to learn and use. But that’s not all. Packages built with Prism Deploy install with near 100% reliability—every time. This means that you’re not spending days or weeks scripting deployment tasks and you’re not visiting computers fixing problems post-deployment. Prism Deploy makes your company more efficient and productive.

What Kind of Packages can I Build with Prism Deploy?

Whether you need to rollout a complex application suite, delete some troublesome software, install a new printer, or change users’ permissions on a registry key or file, Prism Deploy can package and deploy those changes.

How the Prism Deploy Editor Works

Prism Deploy Editor has an Expert that takes you through the simple steps to building a deployment Package:

1. Take a baseline picture of a clean PC
2. Make changes (install software, delete software, update the registry, and so on.)
3. Name the Package
4. Find changes to create the Package

Please refer to this documentation for important best practices for building Packages. This documentation includes information on using the advanced features such as customizing Packages to install in diverse environments, digitally signing a Package so only authorized computers install it, setting requirements for target computers to check before installing a Package and much more.

Contacting Technical Support

If you are unable to locate answers to your questions about Prism Deploy within this document, please use the following resources to receive assistance:

Web site: www.newboundary.com

Here you will find the online New Boundary Technologies Support Forum, knowledge base articles, and responses to frequently asked questions. The Support Forum is an interactive discussion tool that will bring you in touch with other users of New Boundary Technologies software.

By registering, you can stay up-to-date with the forum(s) of your choice. Automatic e-mails will be sent to you as new messages are posted.

Phone: 612-379-1851
or 800-747-4487

Available 8:30 A.M. to 5:00 P.M. Central Time, Monday through Friday

What is a Prism File?

Prism Deploy lets you create several types of files to suit a variety of needs and situations. Pictures, Packages, and Prism Executable Packages are the three unique types of files created by the Prism Deploy Expert.

All three types of files:

- Bring multiple files together in one place.
- Are automatically compressed prior to being stored.
- Can be edited and customized.
- Can be installed on — and uninstalled from — any computer.
- Can be created manually.

The Prism files differ in the amount of information stored in each and how each file is installed on a computer.

- Package
- Prism Executable Package
- Picture

Package file

Packages are stored as a single, compressed files called Portable Windows Change (.PWC) files. Packages can:

- Install and uninstall software on any computer(s), anywhere.
- Fix Windows and application problems.
- Set or change registry and NTFS permissions.
- Evaluate target systems for administrator-defined installation requirements.
- Transport Web downloads or applications from one computer to another.
- Distribute and repair applications for mobile users.
- Install patches, upgrades, and incremental changes enterprise-wide.
- Install printer drivers on-the-fly.

Packages contain changes, additions, and deletions to the software on the computer, including:

- INI file lines or sections
- registry keys
- registry and NTFS permissions
- sections of special text files (such as AUTOEXEC.BAT and CONFIG.SYS)
- single desktop icons
- single items in the Start menu
- files and folders

The easiest and most complete way to create a Package is by using the Prism Deploy Expert. You can also create Packages by comparing two existing Packages or by building a customized Package.

Packages typically include the contents of folders and files. However, if you create a Package by comparing two Pictures that store only file information (not the file contents), the resulting Package will not contain the contents of files.

When a Package is installed, the contents of the Package is added and merged to that on the target computer.

Prism Executable Package

Prism Executable Packages are actually standalone executables (.EXE).

A Prism Executable Package will be larger than the Package used to create it, because the Prism Executable Package contains the software that the target computer needs to run it. Prism Executable Packages are ideal for updating remote computers, traveling laptops, and other computers over which you do not have full control.

Picture file

A Picture is a copy of the files and registry settings from the computer, compressed into a single file. Pictures can:

- Be used later as a baseline for finding changes.
- Backup and restore your customized software setup.
- Be compared to the current setup or another Picture to find changes.

The extension Prism Deploy uses for Picture files is .PWF.

A Picture contains information about the software on the computer, as well as the configuration of the computer, including:

- INI files
- registry sections (for example, all of the current user or hardware settings sections)
- registry and NTFS permissions
- special text files (such as AUTOEXEC.BAT and CONFIG.SYS)
- the desktop (all the icons that are displayed on the Windows desktop)
- shortcuts
- the Start menu
- information about files and folders

Even though individual files and registry sections are compressed into a larger file, they each retain their original path, size, creation date, and so on.

When you take a Picture of the computer, you may specify the information that will be stored in the Picture or use one of Prism Deploy’s predefined options. When Prism Deploy takes a Picture using its preset options, it does not store the contents of each file, only information about the file. Storing only file information decreases the time needed to take a Picture and takes less disk space than storing the contents. Customized Pictures can store the complete contents of folders and files. (See Customized Picture Options.)

When a Picture is installed on a computer, files and registry information are copied to the target. Shortcuts, files, and folders on the Desktop, in the Start menu, and in Application Data that are not part of the Picture are removed from the target.

How Prism Deploy Stores Files and Folders in Pictures

The information stored in a Picture affects how you can use the Picture later on. If the registry or INI files are included in a Picture, Prism Deploy stores all the information about these files, including their contents. For all other files, Pictures can store:

- just information about the files
- the entire contents of the files it finds on your computer

Contents of a Baseline Picture Taken with the Expert

When you take a Picture using the Expert, the resulting Baseline Picture can be used as a basis for comparison when you install new hardware or software. The Expert takes a Picture of your entire system, recording the name, and time and date stamp of all files. However, it does not store the contents of files and folders, just information about them. Storing only the file information takes considerably less disk space and time, but still provides all the information Prism Deploy needs later to find the changes made to a computer.

Contents of a Customized Picture

Storing the contents of files in a customized Picture lets you restore entire files from the Picture. The files are available if you need to install the Picture on a computer or recover a file later on. You can either install the entire Picture or drag-and-drop a file from the Prism Deploy Editor window to the appropriate folder on your system. Installing the Picture does not remove files that have been added since the Picture was taken.

When Prism Deploy stores the contents of files in a Picture, the resulting Picture is rather large. Prism Deploy compresses its information about 40–70%. If you take a Picture containing 500 MB of content, the resulting Picture with file contents is approximately 250 MB.

A Picture alone cannot be used to install an entire operating system or restore a computer to its original state. Because Prism Deploy is a 32-bit application, it must have a Windows operating system loaded before it can run. Prism Deploy Pictures cannot return a computer to its original state because installing a Picture does not remove files or registry entries that were added after the Picture was taken. However, a combination of Pictures and Packages can be used to return a computer to its original state.

You can also configure Prism Deploy to capture version information. Check the appropriate box in the Options | Settings tab. This setting applies to both Pictures taken with the Expert and Pictures you create manually. It may affect which, if any, conflicts that Conflict Checker finds between two Prism files.

What Happens When You Install a Prism File

When a Prism file is installed, all the files and directories whose contents are stored in the Prism file replace existing files and directories with the same name on the computer.

Merging Registry Information

Registry information is merged into the computer’s existing registry. This means that:

- Registry keys that are in both the Prism file and in the computer’s current registry will contain the Prism file’s value after it is installed on the computer.
- Registry keys that are only in the Prism file are added to the computer’s registry.
- Registry keys that are only in the computer’s current registry are not affected.
- Permissions for registry keys in the Prism file are added to the key's existing permissions.

Adding Files, Folders, and Icons

When a Prism file is installed, only directories and files with the same name are replaced on the computer. The following information is merged into the existing system in the same way as registry keys are merged when a Prism file is installed on a computer:

- INI file lines
- Registry keys
- File and folder permissions
- Sections of special text files (such as AUTOEXEC.BAT, CONFIG.SYS, NET.CFG)
- Single desktop icons
- Single items in the Start menu
- Items that were deleted

Delete and Replace Items

If the Prism file contains deleted items, items with the same name on the target system are deleted.

You can define exactly when you want to replace or delete existing files when a Prism file is installed. For example, you can specify that Prism Deploy should only replace the file if the time/date stamp on the Prism file is later than on the target system. You can even set different replace and delete options for each Prism file **and** for each item contained in a Prism file.

Add/Remove Programs

For a Prism Executable Package, the File Properties let you add an entry to the Add/Remove Programs list through the Windows Control Panel. This option is implemented only if the Prism Executable Package is installed directly.

If the Prism Executable Package is installed or run:

- Through the Prism Client or from the Prism Deploy Editor (with), it is treated as a Package. The contents can be uninstalled via the Uninstall file. But, an entry is not added to the Add/Remove list in the Windows Control Panel.
- Directly (for example by double-clicking on the .EXE file) **and** the **Add an entry to Add/Remove Programs when the file is run** check box is selected on the File Properties | Add/Remove tab, the entry in Add/Remove Programs can be used to reverse or uninstall the file.

Installing a Picture versus Installing a Package

By default, when a **Picture** is installed, all the files whose contents are stored in the Picture replace existing files with the same name on the target computer. Files that exist on the target computer but not within the Picture remain unaffected. However, if the Picture contains **Start** menu items and/or desktop shortcuts and folders, the desktop and **Start** menu in the Picture replace the desktop and **Start** menu on the target computer.

Registry information in the Picture is merged into the computer's existing registry. This means that registry keys that are in both the Picture and in the target computer's registry contain the Picture's value after the Picture is installed on the computer. Registry keys that are only in the Picture are added to the target computer's registry. Registry keys that are only in the computer's registry are not affected.

When a **Package** is installed, only folders and files with the same name are replaced on the computer. All other information, including the Start menu, INI files, and desktop items, is merged into the existing system in the same way registry keys are merged when a Picture is installed on a computer. If the Package contains deleted items, the items on the target computer with the same name are deleted.

You can define exactly how Prism Deploy should replace or delete existing files when it installs a Prism file. For example, you can specify that Prism Deploy should only replace an existing file if the time/date stamp on the file within the Prism file is newer than the existing file on the target computer. You can even set different replace and delete options for each item contained in a Prism file. For example, you may want Prism Deploy to replace older files with newer files except for one DLL file that you want to keep at an older version of the DLL to ensure that your printer continues to work.

Caution: Pictures can be used to back up customizations and other changes. However, using a Picture as a backup file *does not* restore your computer to exactly the state it was in when you took the Picture.

Best Practices

To take full advantage of Prism Deploy's powerful, flexible packaging and editing technology, please follow the Best Practices listed below. These suggestions come directly from the technical support department and preview some of the powerful features and concepts of Prism Deploy, which are thoroughly explored in other parts of the Help and User Guides.

Use the Expert to Lead You through the Correct Steps in Package Creation

Prism Deploy is not like other packaging tools for software distribution. There is no complex scripting required. For nearly all situations, the Expert makes the right choices. Once you become more familiar with Prism Deploy, you can experiment with custom settings, taking custom pictures, manually creating Packages, and so on. But start with the Expert.

Build Packages on a Clean Machine

It is important to build on a clean machine because Prism Deploy uses differencing technology (that is, what has changed on this system since the Baseline Picture was taken?). A clean machine insures that Prism Deploy pulls all the necessary files, folders, shortcuts, and settings into the Prism file.

For example, if you want to build a Package that includes a desktop shortcut to a system utility such as Notepad, be sure to build it on a system that does not already have a desktop shortcut to Notepad. Otherwise, Prism Deploy will not detect that you have added a new item on the desktop, so it will not include the shortcut in the Package.

Build Operating System-Specific Packages when Necessary

Many Packages will work on any version of Windows. However, some applications install platform-specific changes. You need to create platform-specific Prism files in the following situations:

- User-specific files and folders are stored in different locations on the various operating systems. Most paths are handled automatically by Prism Deploy (user Desktop, Start menu, and Application Data).
- Applications such as MS Office, MS Internet Explorer, and Netscape have many hooks integrating them tightly into the operating system.
- Printer drivers and drivers in general are typically operating system-specific.

Be Mindful of Reboots

- If you have run a setup program in preparation for capturing its changes with Prism Deploy and the setup program prompts you to reboot, reboot before capturing the changes with Prism Deploy.
- Add reboot options to Packages to match the behavior of the native setup program. If the program's native setup prompted you to reboot after installing the software, that is an indication that the Prism Deploy Package should also be configured with a reboot property.

Launch and Configure the Application before Finding Changes

Before capturing changes to create a Prism file, launch the software program to be sure that it is installed correctly and runs satisfactorily. Then, configure settings such as responses to licensing or registration prompts, response to the user name prompt, a default Save As location, and so on.

Stop Newly Installed Services before Finding Changes

If you are building a Prism file for software that installs services, stop the newly installed services before finding changes and building the Prism file. Prism Deploy may not be able to capture the contents of a file that is in use because of a loaded service. The resulting Prism file will contain a reference to the file but since the file contents are not stored, the file will not be copied to the target computer when the Prism file is installed.

Check the Paths Embedded in the Package

With the Package open in the Editor, click **C:** in the toolbar. The Editor will show you every path that is referenced anywhere in the Package (including targets of shortcuts, registry data, INI file entries, and so on).

Sometimes you will discover there is a path that needs to be changed. For example, there may be a pointer to C:\DOCUMENTS AND SETTINGS\ADMINISTRATOR\FAVORITES. You probably want to change the "Administrator" portion of the path to a variable so that the logged-in user’s name is inserted into the path when the Package is installed. The Edit Paths dialog box is the place to make that change.

Set Variables as Necessary to Customize the Deployment on Each Target Computer

In the example given in Check the Paths Embedded in the Package, you can change the "Administrator" entry to the "username" system variable or change the entire path C:\DOCUMENTS AND SETTINGS\ADMINISTRATOR to the system variable userprofile within Prism Deploy’s Edit Paths dialog box.

There are many ways to use Prism Deploy’s powerful smart variables to ensure successful software distribution in even the most mixed environments.

Please see Working with Variables for more details.

Clean Extraneous Entries Out of Your Package Before Distribution

You may notice that a Package contains an entry for a virus-scanning log file. Or, perhaps, you mapped a network drive between the time you took the baseline Picture and when you captured changes. The log file and the network mapping are included in the Package, but these items can (and should) be deleted. With time, you will get a sense for which entries are relevant and which are not. If you find you are deleting the same entries over and over, there is a way to configure Prism Deploy to always ignore certain files, folders, and settings.

Test, Test, Test!

Always do some testing with your Packages in a non-production environment to make sure you understand how the Packages interact, how they function on different operating systems, how the prompting messages are displayed, and so on. If you are trying to build and distribute one Package that will work on all operating systems, be sure to test it on all platforms! And while you are testing, always use Uninstall files in your testing environment.

Always Use Uninstall Files in Your Test Environment

Uninstall files give you a quick and easy way to remove or uninstall Prism files from target computers.

- If your tests involve using Prism Executable Packages, be sure to configure the Prism Executable Packages to add an entry to Add/Remove Programs. (See Add/Remove Programs tab.)
- During the test phase do not use the /NoRollback or /NoRollbackInfo options when you are installing Packages. (See "/InstallPackage" in the *Deployment Guide* or the Console online help for information on these options.)

Windows Vista

Security and Usability Features of Vista

Windows Vista introduces many changes intended to improve the security and usability of the operating system over previous versions of Windows. These changes include:

- **User Account Control** (UAC) - UAC is Vista's method of reducing the chance for an attack on the operating system by forcing all users to run in standard mode and requiring explicit consent to do administrative tasks. Standard users are not allowed to perform administrative tasks such as installing software or changing system settings. When administrative tasks need to be run, the user will be prompted for administrator credentials. UAC impacts a number of applications and user paradigms.
- **Fast User Switching** - This Vista feature (also available in Windows XP) allows multiple users to run simultaneously and provides the ability to switch between users without logging off.
- **Windows Resource Protection** (WRP) **-** WRP, which extends Windows File Protection of earlier Windows versions, is a security feature that prevents unauthorized applications from modifying various system files, folders, and registry keys.
- **Special Folders:** Special Folders represent predefined folders in Windows, such as Desktop and Start Menu. The actual location of the special folder can vary from one machine to another. While present in many previous Windows versions, Vista greatly increases the number of special folders available. On Vista, they are called "Known Folders".

Prism's Support for Vista

Prism Deploy Editor and the packages it generates are designed to work seamlessly with the Windows enhancements listed above. The Editor provides the following features and benefits for Windows Vista users:

- Prism Deploy Editor properly recognizes, scans, and captures changes to all aspects of Vista, including Known (Special) Folders and Junction Points.
- Full support for Fast User Switching - so a package containing user and machine-specific portions can be deployed to a machine with multiple active users.
- The Editor operates almost identically to previous releases. And except for some changes in the Editor's tree view and how special folders are organized in the tree, it looks very similar to previous versions.
- New packages built on non-Vista machines can be safely deployed to Vista (assuming the package contains only Vista-compatible applications).
- Packages built with earlier versions of the Editor can be converted to make them Vista-compatible.

Considerations for Vista

While the Editor's support of Vista hides from the user much of the complexity associated with the changes introduced by Vista, consideration should be given to the following points:

- **Administrator Account:** As with previous versions of the Prism Deploy Editor, New Boundary Technologies recommends that on Vista you launch the Editor when you are logged in as an **Administrator**, not as a Standard User. The Editor will display the Vista UAC Consent Prompt when the administrator initiates an operation that requires full system access, after which no further UAC Consent Prompts will be displayed.
- **Old Packages:** Packages built using previous versions of the Editor must be converted to work on Vista. When opening such a package, the Editor will automatically open the Upgrade Package Format dialog to prompt you to convert the package. If necessary, conversion can also be done later (see Changing File Path Names).
- **Non-Vista-Compliant Applications:** Packages that contain non-Vista-compliant applications should not be expected to work on Vista.

64-bit Operating System Support

Prism Deploy Editor supports both 32-bit and 64-bit operating systems:

- A Prism file generated by building a package on a 32-bit machine renders a 32-bit file.
- A Prism file generated by building a package on a 64-bit machine renders a 64-bit file.
- A 32-bit Prism file can be deployed on *either* 32-bit or 64-bit operating systems.
- A 64-bit Prism file can be deployed *only* on 64-bit operating systems.
- The operating system bitness can be used in installation rules.

There are a just a few considerations to keep in mind relative to the above points:

- If you manually create a Package or Prism Executable Package, you will be asked whether to render a 32-bit or 64-bit file.
- When upgrading an existing (32-bit) Prism file, the upgrade must be performed on a 32-bit OS.
- If, on a 64-bit operating system, you create a package or executable package containing exclusively 32-bit applications, the resulting file *cannot* be deployed on a 32-bit OS. This is because it was built on a 64-bit operating system. For the resulting file to be deployable to 32-bit operating systems, it must be built on a 32-bit OS.

Installing Prism Files on Multiple Operating Systems

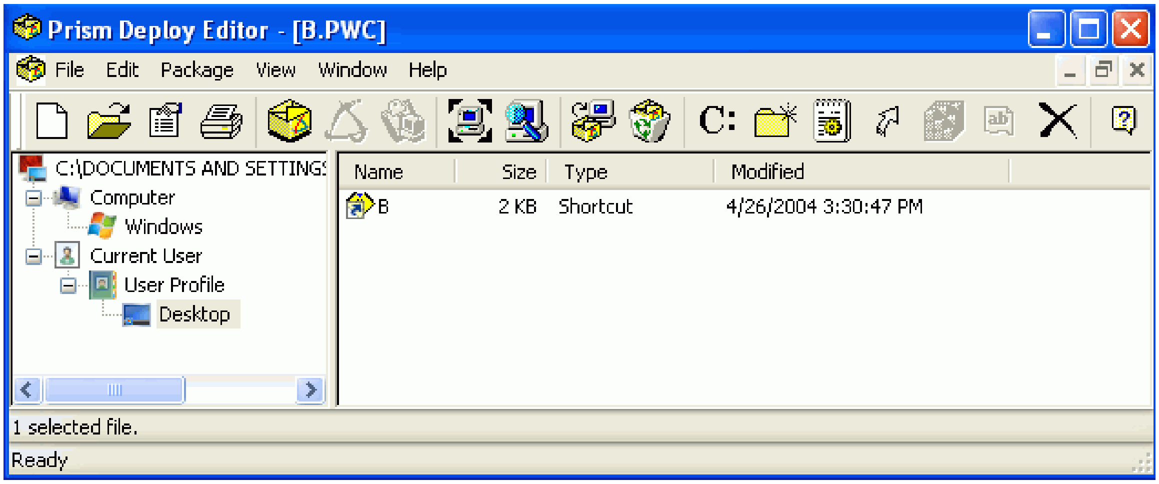
Many Packages will work on any version of Windows. However, some applications install platform-specific changes. You need to create platform-specific Prism files in the following situations:

- User-specific files and folders are stored in different locations on the various operating systems. Most paths are handled automatically by Prism Deploy (user Desktop, Start menu, and Application Data).
- Applications such as MS Office, MS Internet Explorer, and Netscape have many hooks integrating them tightly into the operating system.

Working in the Editor Main Window

Editor Main Window

When a Picture, Package, or Prism Executable Package is open, the Prism Deploy Editor window displays the file in much the same way as Windows Explorer. The contents of a Prism file include many types of items, including files, folders, special folders, registry keys, and drives.



Name	Size	Type	Modified
B	2 KB	Shortcut	4/26/2004 3:30:47 PM

1 selected file.

Ready

Layout

The Editor window consist of a **toolbar** at the top, a **tree view** of the file on the left side, a **detail pane** on the right side, and a **status bar** at the bottom of the page.

- **Toolbar**: The toolbar provides quick access to commonly used functions. If a toolbar function is not applicable to the currently selected item, its icon will be disabled. Click here for a complete list of icons in the toolbar and their meaning.
- **Tree view:** The left side of the Editor window displays a tree view of the file. Prism Deploy organizes the tree into a hierarchical structure. Click the **+** or **-** signs to expand or collapse the items in the tree. Items in the tree may be special folders, registry branches, folders, or other item types described below.
- **Detail pane:** The right side of the Editor window displays the contents of the selected item in the tree view. Click on a column label to sort the items by name, size, type, or date modified.
- **Status bar:** The Status Bar provides status on your current selection and actions.

Contents

There are a variety of item types that constitute a Prism file:

- **Computer**: Contains items specific to the computer. It is a "virtual" folder in that it does not correspond to any specific folder on the computer. Items in this virtual folder are common across users.
- **Current User**: Contains items specific to the current user. Like **Computer**, this is a virtual folder.
- **Deleted Items**: This virtual folder defines files, folders, and registry keys to delete. These are partitioned into **Computer** and **Current User** virtual folders.

- **Special folders:** Special Folders represent common directories on a Windows computer, such as Windows, Desktop, Application Data, Common Programs, and Start Menu. Each special folder is associated with either the Computer (common) or Current User (per user) virtual folder. Special folders may contain folders, variable folders, files, and shortcuts. Some special folders may also contain other special folders. For example, the Windows special folder contains the System special folder.
- **Registry branches:** Registry branches contain registry keys and values. These appear directly under the **Computer** and **Current User** virtual folders. For example, the HKEY_LOCAL_MACHINE branch belongs to the **Computer** virtual folder, and the HKEY_CURRENT_USER branch belongs to the **Current User** virtual folder.
- **Drives:** A computer's drives appear directly under the **Computer** virtual folder. This is the only place they may appear.
- **Folders:** Folders denote folders on the computer, and may appear within drives, special folders, and other folders.
- **Variable folders:** Variable folders represent a variable that has been defined for the environment.
- **Files:** Files on the computer. Files appear only in the detail pane. Icons associated with files have special interpretations in the Editor:

  In Picture files, grayed items indicate that only file information is stored, not the file contents.

  Packages and Prism Executable Packages always contain file contents unless you created them by comparing two Pictures that contained only file information or a file was busy when it was created.

  If the item is not grayed, the file contents and file information are stored in the Prism file.

  A red X on an item in a Prism file means that Prism Deploy discovered that the item has been deleted from the system. The file is removed from the target computer when the Prism file is installed.
- **Shortcuts:** Shortcuts to files on the computer. These appear only in the detail pane.
- **Registry keys:** Registry keys defined in a registry branch or within another registry key.
- **Registry values:** The registry value associated with a registry key. These appear only in the detail pane.

Editor Toolbar

The toolbar in the Editor main window includes these icons:

Create a new Package

Open Prism file

Properties

Print contents of Prism file

Create a Prism Package

Open the Prism Console

Deploy the Prism file

Take a Picture

Find changes

Install the Prism file

Uninstall the Prism file

Scan the Prism file for path references

Add a folder

Add a file

Add a shortcut

Add a registry key

Add a registry value

Delete the highlighted item

Defining the Editor Options

Prism Deploy lets you customize the way the Editor operates on your computer, or on other computers where you use the Editor. Use the **Options** dialog box to define Prism Deploy Editor settings. Once you define a setting, Prism Deploy uses that setting for all Prism files that you create or install on that computer only. These settings do not apply to the target computers you manage.

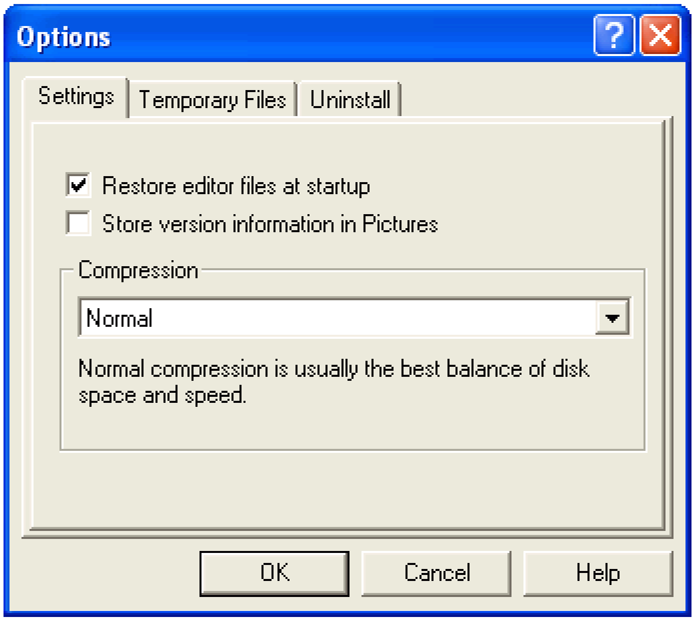
The **Options** dialog box lets you define these settings for using Prism Deploy:

- Set Prism Deploy to open the most recently edited Prism file when you start the Editor.
- Define a directory for storing temporary Prism files.
- Define a directory for storing uninstall information.

To open the **Options** dialog box, select **Package | Options** from the menu bar. The tabs available are:

- Settings tab
- Temporary Files tab
- Uninstall tab
- Checkpoint/Restart tab

Settings Tab



☑ Restore editor files at startup
□ Store version information in Pictures

Compression

Normal

Normal compression is usually the best balance of disk space and speed.

The Options | Settings tab opens when you choose **Package | Options**. Use the choices on this tab to set options for your Prism files and how you work with Prism Deploy.

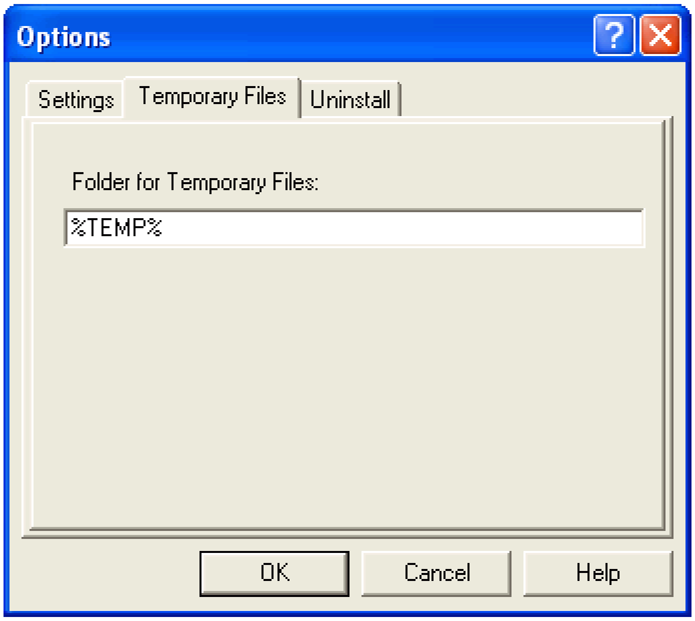
- **Restore Editor Files at Startup**: Automatically open the most recently edited Prism file when you start the Prism Deploy Editor.

  If this check box is cleared, Prism Deploy closes all open files when you quit.
- **Store version information in Pictures**: Store the version information for files within the Picture. If you take a Baseline Picture, file version information is always stored. However, when you use the Expert to take a Picture, the default Picture does not contain file version information.

  If you want to use Conflict Checker Professional to check for version conflicts in Pictures, you should select this option to store version information. However, it can increase the time required to take a Picture.
- **Compression**: Choose an option in the drop-down list to change the default compression percentage for the Prism files.
  - **Normal compression:** Prism Deploy offers an optimum level of compression at the Normal setting. This option will meet your needs in almost all cases. It is the default option.
  - **Low compression:** Choose this option if the size of the Prism file is not important, but speed is an issue. For example, choose Low compression if you plan to use the Find Changes feature, you have a huge number of files, and you need to locate the changes quickly. The resulting Prism file may be large.
  - **High compression:** This option offers the maximum level of compression when space is at a premium. Use High compression only when the size of the Prism file is critical. Using High compression dramatically increases the time it takes to save the file, yet provides a relatively small gain in the percent compression.

If you change this setting, the percent compression is reflected in the **Compression (%)** field on the File Properties | Information tab after you edit the Prism file.

Temporary Files Tab

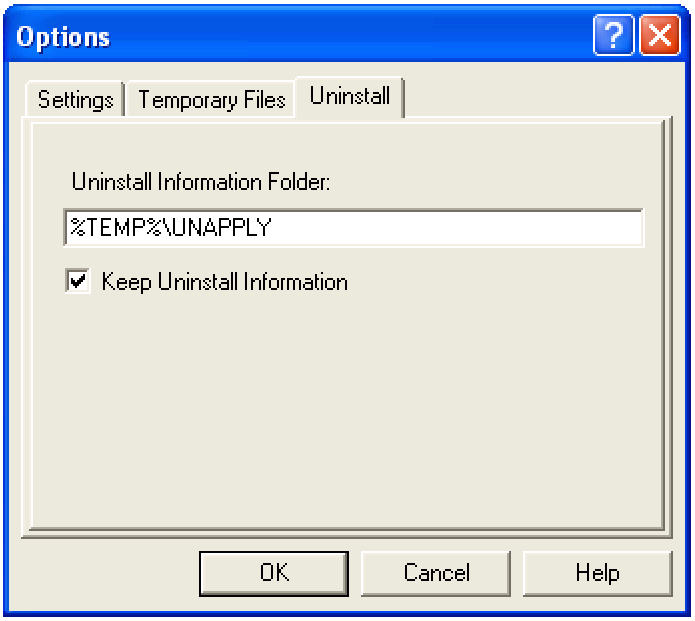


Folder for Temporary Files: %TEMP%

The Options | Temporary Files tab lets you define a location for Prism Deploy temporary files. This folder is where Prism Deploy stores file components as you are editing them and where it stores temporary (.TMP) files when finding changes on the computer.

The default location for the temporary files depends on the operating system and individual computer settings. You may want to enter a different directory if the current location is low on space or if it is automatically emptied upon logout.

Uninstall Tab



Uninstall Information Folder: %TEMP%\UNAPPLY

☑ Keep Uninstall Information

The Options | Uninstall tab lets you define the directory where Prism Deploy stores the Uninstall files. The Uninstall files are created when it installs a Prism file and read when it uninstalls Prism files.

- The default location for the uninstall files depends on the operating system and individual computer settings. If you want the uninstall files stored in another location, type the path to the directory in the text box.
- **Keep Uninstall Information**: If you have disk space concerns, you can clear the **Keep Uninstall Information** check box so no permanent uninstall information is stored.

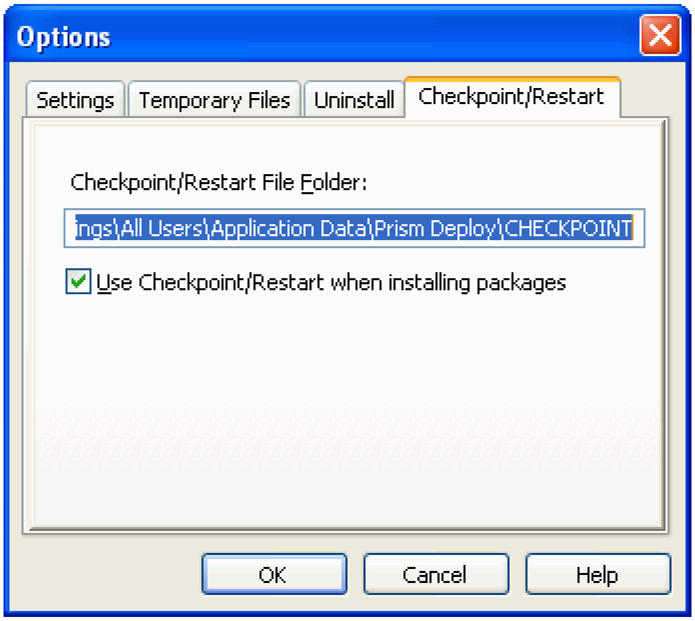
Note: You can only uninstall or rollback software that was installed with the **Keep Uninstall Information** check box selected.

Remember this setting affects your current computer only. To change the default location where PTCLIENT.EXE stores uninstall files on target computers, you must add the following registry entry on those computers:

HKCU\SOFTWARE\Lanovation\PictureTaker\Settings\Directories; name = "Unapply Folder", data = <path to folder>

The path and folder must exist before you make the registry change.

Checkpoint/Restart tab



Checkpoint/Restart File Folder: ings\All Users\Application Data\Prism Deploy\CHECKPOINT

☑ Use Checkpoint/Restart when installing packages

The Checkpoint/Restart option can save bandwidth when a partially installed Prism file is interrupted due to a communication break. When installation of a remote Prism file fails before it is complete, Prism Deploy rolls back all of the changes and saves them to a checkpoint file (.PKGCHKPT) on the target computer. When the connection is re-established, only the items that were not installed initially need to be transferred over the network or other connection. Installation takes place more quickly and uses less bandwidth than starting over and re-transmitting the contents of the entire Prism file. The benefit of this feature is greatest for large Prism files being installed over a slow connection, such as a dial-up modem.

This tab is part of the Options dialog box, which is opened by selecting **Package | Options**.

- **Checkpoint/Restart File Folder**: Enter the location on the target computer where the checkpoint file should be stored. The default location for this file depends on the operating system and individual computer settings.

  To change this information, type the path to the new directory in the text box.

☑ **Use Checkpoint/Restart when installing Packages**: With this option selected, **if** the Prism file installation is interrupted, a checkpoint file is created on the target computer. When the connection is re-established, or this Prism file is reinstalled after the interruption, the Prism Client uses the checkpoint file to resume the installation where it left off.

□ **Use Checkpoint/Restart when installing Packages:** With this option turned off and the connection is re-established or this Prism file is reinstalled after the interruption, the contents of the entire Prism file are transferred to the target computer and installed.

Taking a Picture

Prism Deploy provides preset options for taking a Picture or you can specify exactly which files you want included in the Picture. You can take a Picture of the entire hard drive or just part of it, such as the desktop and Start menu.

Using the Expert versus Taking a Customized Picture

When you take a Picture using the Expert, the Expert takes a Baseline Picture. You can use a baseline Picture as the basis for comparison when you install new hardware or software. Because the Expert takes a Picture of the entire system, the Baseline Picture can be rather large and time-consuming.

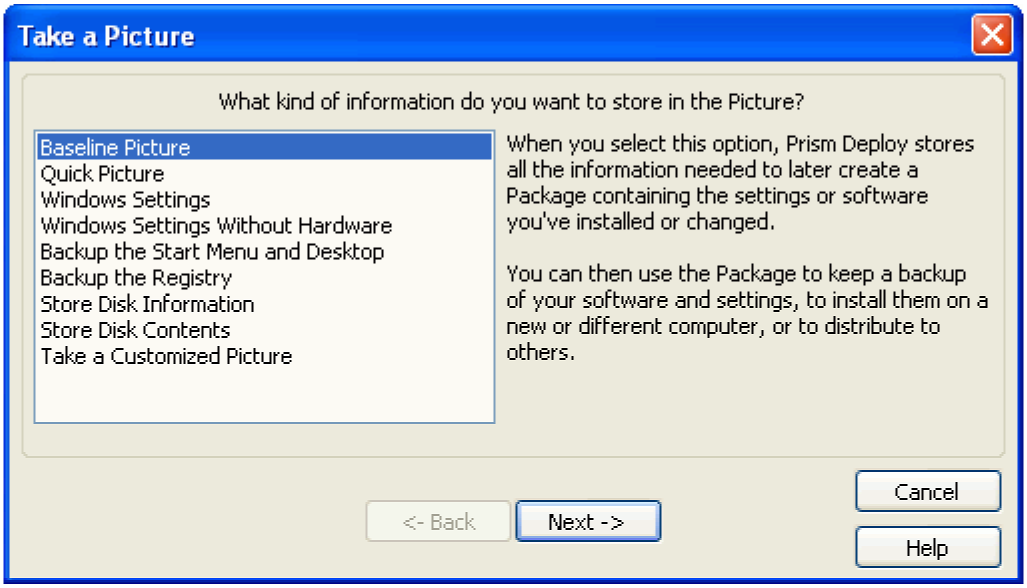
To save time and disk space, you do not always want a Picture of the entire system. Prism Deploy provides some preset options for customizing a Picture or you can specify exactly which files you want included in a Picture. Pictures can be copies of the entire hard drive or just part of it, such as the desktop and Start menu.

When the Expert takes a Baseline Picture, it includes only the file information, such as the time and date stamp and version number. When you take a customized Picture, you can include file contents in the Picture, which makes it ideal for backups.

- To take a Picture with the Expert, click in the toolbar.
- To specify what to include in the Picture, click in the toolbar to open the Take a Picture dialog box. Then you select the options for taking a Picture.

You can also create a Prism file manually.

Picture Options dialog



What kind of information do you want to store in the Picture?

Baseline Picture
Quick Picture
Windows Settings
Windows Settings Without Hardware
Backup the Start Menu and Desktop
Backup the Registry
Store Disk Information
Store Disk Contents
Take a Customized Picture

When you select this option, Prism Deploy stores all the information needed to later create a Package containing the settings or software you've installed or changed.

You can then use the Package to keep a backup of your software and settings, to install them on a new or different computer, or to distribute to others.

The **Take a Picture** dialog box presents some preset options for taking a Picture.

To open the **Take a Picture** dialog box, click in the toolbar.

Options

Select one of the following options and click **Next**.

- **Baseline Picture**: Select this option to use the Picture as the basis for a comparison, for example when you install new software or hardware. After installation, you can create a Prism file that contains the changes made by the installation.

  Prism Deploy does not store the contents of each file on the selected drives, only the file's information, such as the time and date stamp. When you do a comparison later, Prism Deploy includes new files and files that changed in the Prism file. This type of Picture is not good for backing up files.

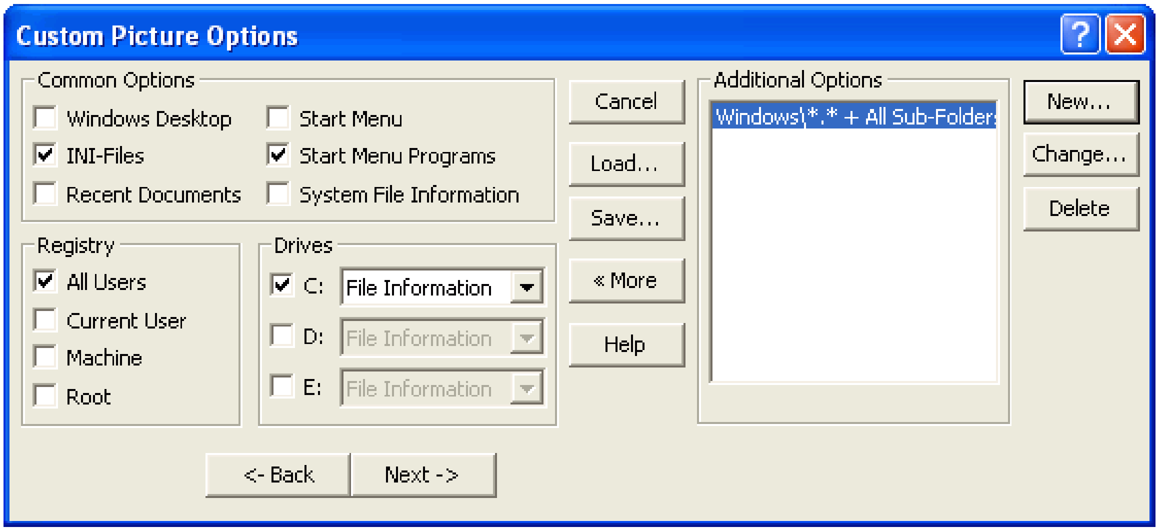
  If you have more than one drive and select this option, the Picture Drive Letters dialog box opens.
- **Quick Picture**: Stores the information needed to create a Prism file. However, the **Quick Picture** option assumes that new files will be installed only in a new folder, not in an existing folder. It does capture new files in any of the standard Windows folders such as \WINDOWS\SYSTEM. This option saves time when creating the Prism file, but the Prism file may be incomplete if the program files were added to an existing folder.
- **Windows Settings**: Stores all of the Windows settings in the Picture, including INI files, registry settings, driver information, and shortcuts on the desktop and in the Start menu.
- **Windows Settings Without Hardware**: Stores all of the Windows settings in the Picture, except for registry settings relating to the computer's configuration. Select this option if you want to install the Windows settings on a computer with a different hardware configuration.
- **Backup the Start Menu and Desktop**: Stores the shortcuts and folders on the desktop and Start menu. Use this option to back up the Windows preferences or to keep a list of shortcuts that you can drag and drop to and from the Picture.
- **Backup the Registry**: Stores the entire registry. You can use the Picture for disaster recovery, troubleshooting, or for figuring out where Windows and applications store their information.
- **Store Disk Information**: Stores a list of the files on the hard drive, which can be useful for determining which files have been added, changed, or deleted. This option does **not** store the contents of the files, which means less disk space is used but you cannot use this Picture to restore the files if they are lost.

  If you have more than one drive and select this option, the Picture Drive Letters dialog box opens.

- **Store Disk Contents**: Stores the contents of all the files on the hard drive. This option is good for backing up files, but requires a large amount of disk space. It generally requires about 40 percent of the total disk space in use.

  If you have more than one drive and select this option, the Picture Drive Letters dialog box opens.
- **Take a Customized Picture**: Use this option to specify exactly what you want included in the Picture. The Customized Picture Options dialog opens.

Customized Picture Options dialog



Custom Picture Options

Common Options

□ Windows Desktop
☑ INI-Files
□ Recent Documents
□ Start Menu
☑ Start Menu Programs
□ System File Information

Registry

☑ All Users
□ Current User
□ Machine
□ Root

Drives

☑ C: File Information
□ D: File Information
□ E: File Information

Additional Options

Windows*.* + All Sub-Folder:

The **Custom Picture Options** dialog box lets you select more specific files and settings to include in a Picture.

The Custom Picture Options dialog box opens when you select **Customized Picture** and click next on the Take a Picture dialog box.

Common Settings

- **Windows Desktop**: Stores the shortcuts and folders on the Windows desktop. Use this option when you are adding or removing items from the desktop.
- **INI Files**: When you use this Picture to find changes to the environment, Prism Deploy goes through INI Files line-by-line to find changes and additions. It also finds new INI files. Select this option when you are installing applications.
- **Recent Documents**: Records the contents of the Documents folder in the Start menu
- **Start Menu**: Records the folders and shortcuts found in the Start menu.
- **Start Menu Programs**: Captures the items found in the Programs folder in the Start menu. When you use this Picture to find changes, Prism Deploy finds all new entries in the Program folder and in any new subfolders.
- **System File Information**: Records information about the files in the WINDOWS\SYSTEM or SYSTEM32 directory. This directory is where Windows typically stores drivers and .DLL files. Use this option when you are installing applications or adding drivers.

Registry

- **All Users** (referred to as HKEY_USERS): This registry is where Windows stores information about:
  - how the Windows desktop should look (the default desktop)
  - the customizations that each user has made to the Windows environment.

    Information about individual users cannot be updated in this section of the registry. Only include this option in a Picture to update the default setup.
- **Current User** (referred to as HKEY_CURRENT_USER): When a user starts Windows, the information about that user's desktop is copied from its storage location to Registry (Current User).

Registry information for the current user is often updated during software installations, so include this option when you install software.

- **Machine** (referred to as HKEY_LOCAL_MACHINE): This registry is where Windows stores software- and hardware-specific information. Select this option when you are installing software, changing hardware, adding drivers, and so on.
- **Root** (referred to as HKEY_CLASSES_ROOT): This registry is where Windows keeps software information, including information such as OLE (object linking and embedding) and file associations that support drag and drop operations and shortcuts.

  This information is shared by all users of a computer. A copy of this registry key is located in Registry (Machine), so it is not necessary to include this option if Registry (Machine) is also selected.

Drives

If you want to store file information or contents from one or more drives, click on the letter of the drive whose files you want to store and select an option from the drop down list.

- **File Information**: Store a list of files on the selected drive. This option does **not** store the contents of the files, which means less disk space is used but you cannot use this Picture to restore the files if they are lost.
- **File Contents**: Store the contents of files on the selected drive. This option is good for backing up files, but requires a large amount of disk space. It generally requires about 40 percent of the total disk space in use.

If you want to store file information or contents from a drive not shown, click the **More** button to view additional options. Next to the Additional Options list box, click the **New** button.

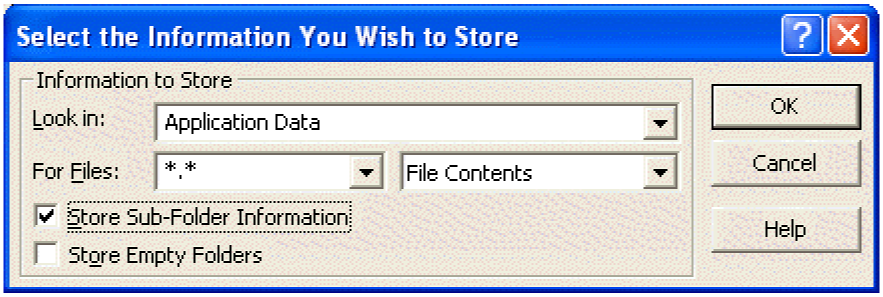
Load

Load a saved Picture definition. See Loading a Picture Definition.

Save

Saves the options defined in this dialog box. See Saving Picture Definitions.

Select the Information You Wish to Store dialog



Select the Information You Wish to Store

Information to Store

Look in: Application Data

For Files: *.* File Contents

☑ Store Sub-Folder Information

□ Store Empty Folders

The **Select the Information You Wish to Store** dialog box lets you include specific file types in a Picture. To open this dialog box, click the **New** button in the Custom Picture Options dialog box.

- **Look in:** Select the part of the computer environment that you want to take a Picture of or type a directory name. You can also add a directory name. For example, you can have Prism Deploy take a Picture of WINDOWS\FONTS or I:\PROGRAMS\DATABASE.
- From the **For Files:** drop down list, select the type of files you want to look for:

  .: all types of files

  config.sys, autoexec.bat: boot text files

***.grp, progman.ini**: Program Manager files

***.ini**: INI files

***.ink,*.pif**: shortcuts

You can also type file names or extensions in this field. For example, type ***.dll** to take a Picture of DLL files. Prism Deploy remembers the file name or extension you enter and includes it in the drop down list so you can find it easily again.

- Select whether to include **File Information** or **File Contents**.
- Select the **Store Sub Folder Information** check box to include subfolders.
- Select the **Store Empty Folders** check box to include folders that do not contain any files or subfolders.

The options you selected are listed in the **Additional Options** list box in the **Custom Picture Options** dialog box.

Saving Picture Definitions

If you often take the same Picture, you can save time with a Picture definition. That is, save the options you selected in the Custom Picture Options dialog box. The next time you customize a Picture, you can load the saved Picture definition. You can save as many Picture definitions as you want, for use later.

Note: Saving a Picture definition saves only the options you selected in the **Custom Picture Options** dialog box. It does not save the contents of the Picture itself.

To save a Picture definition:

1. Select the options you want to use to take the Picture.
2. In the Custom Picture Options dialog box, click the **Save** button.

   The **Save Custom Picture Options** dialog box opens.
3. From the **Save in:** drop down list, select the drive and directory where you want to save the Picture definition.
4. In the **File name:** text box, enter a name for the Picture definition. Click **Save**.

   Prism Deploy saves the definition as a Portable Windows Information (PWI) file.

Loading a Picture Definition

To use a saved Picture definition to take a Picture:

1. In the Custom Picture Options dialog box, click the **Load** button.

   The Load Custom Picture Options dialog box opens.
2. Click on the name of the PWI file you want to use. Click **Open**. The Picture definition is displayed in the Custom Picture Options dialog box.
3. You can modify the options or click **Next** to take the Picture.

Finding and Installing Changes

Finding Changes

Once you have taken Pictures of the computer, you can use the Pictures to find the changes made to the computer by installing software, adding hardware or drivers, or any other alterations. When you use Prism Deploy to find changes, Prism Deploy compares a Prism file with the current environment and displays the changes it finds. You can also compare two Prism files.

Prism Deploy compares each part of the before environment with the after environment. Any files or folders that are new, have been changed, have been deleted, or have had their permissions changed since the before environment are then stored in a Package. Prism Deploy also goes through the registry, INI files, and certain text files (such as AUTOEXEC.BAT and CONFIG.SYS) line by line and puts any changed, new, or deleted lines from the after environment into the Package. Permission changes on registry keys are also captured in the Package.

Creating a Package with the Expert versus the Toolbar

When you create a Package with the Expert, Prism Deploy automatically uses the Baseline Picture it took as the basis for finding changes.

When you click in the toolbar, you can select another Prism file to compare with the current environment or compare two Prism files. This feature gives you greater flexibility in finding precisely the changes you are interested in.

- Create a Package with the Expert:

  Click in the toolbar. Remember, Prism Deploy will take a Baseline Picture first.
- Take a Picture to use for comparison and creating a Package:

  Click in the toolbar to take a picture. Then click to select a Prism file to use for the comparison.
- Build a Package manually:

  See Creating a New Prism File Manually in the next section.

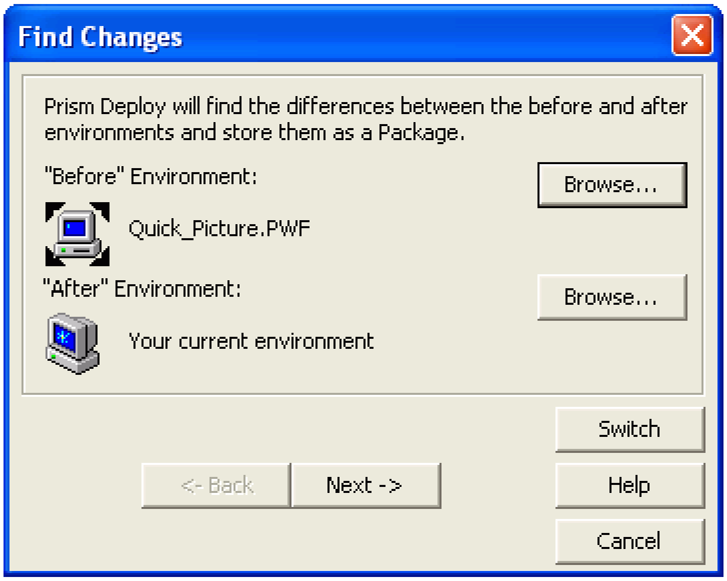
Pick the File You Want to Compare with to Find Changes dialog

The **Pick the file you want to compare with to find changes** dialog box opens when you find changes.

1. From the **Look in:** drop down list, select the directory that contains the Prism file that you want to use to find changes.
2. By default, the dialog box lists Pictures. If you want to compare specific type of Prism file, such as a Package, select **Packages** from the **Files of type:** drop down list.
3. Click on a file name in the list box. Click **Open**.

Prism Deploy uses the selected file to find changes to the computer.

Find Changes dialog



Find Changes

Prism Deploy will find the differences between the before and after environments and store them as a Package.

"Before" Environment:

Quick_Picture.PWF

"After" Environment:

Your current environment

The **Find Changes** dialog box opens when you select a Prism file to use to find changes to the computer.

The **Find Changes** dialog box displays the before and after environments that Prism Deploy compares to find changes.

- **Browse**: To select a different Prism file for the Before or After environment, click the **Browse** button.

  Your computer's current environment is the default for the After environment, which is usually the correct option.
- **Switch**: To switch the before and after environments, click the **Switch** button.
- **Next**: When you are ready to find changes, click **Next**.

  The **Find Changes** dialog box opens, where you can choose the types of changes to find.

Types of Changes to Find dialog

This dialog box opens after you select the before and after environments for finding changes. Here you define whether to find new and add information as well as items that have been deleted, and to set overwrite defaults.

1. Select the **Find things that have added or changed** check box and/or the **Find things that been deleted** check box. You can select one or both of these options, but you must choose one.
2. If you select **Make sure these changes will never replace new files with older ones**, when you install this Prism file, Prism Deploy only replaces existing files if the file version in the Prism file is newer than that on the existing files.

   If this check box is not selected, newer files may be replaced with older ones. (You can change the replace options later if necessary.)
3. Click **Next**. The Pick a file for storing the changes dialog box opens.

Changes in the SharedDLLs

Since many applications share common DLLs and drivers, a special area of the registry is used to keep track of which files are shared and how many applications depend on each file. The registry key HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\SharedDLLs contains a registry value for each registered driver or DLL. Each file keeps a counter (a number) for the applications that depend on this file. When an application installs a new DLL or driver, it often registers the file in this registry branch. If an application installs a driver or DLL that already exists and is registered, the application increments the counter in the registry. Likewise, if an application is uninstalled, it decrements this counter.

When you find changes from the Editor, Prism Deploy monitors the SharedDLLs area of the registry. If it finds that an application has incremented a number for a registered file, Prism Deploy pulls that file into the Prism file, even if it existed prior to the application install. Prism Deploy also increments or decrements the counter as appropriate on each computer when the Prism file is installed or uninstalled.

Installing Changes

Once you have used Prism Deploy to find changes made to the computer, you can reinstall the changes on the computer or distribute the Prism files to another computer. For example, you can:

- Apply Windows settings across a network so all computers have the same settings.
- Quickly install a Web download on other computers without the wait.
- Share applications with others.
- Reinstall applications if something goes wrong.
- Transfer files from your computer at work to the one at home.

When you install changes on a computer, Prism Deploy creates a unique Uninstall file for each computer to which the file was installed. (See Understanding Uninstall Files.)

Options for Installing Prism Files

When you use the Prism Deploy interface to install a file, (that is you click in the toolbar), the file is installed **only** on your computer. To install Prism files on other computers, you need to distribute the Prism file with another method, such as using the Prism Console, attaching it to an e-mail, including a link to the file on the corporate intranet, placing it on a shared network drive, and so on. (See Distributing Prism Files.)

To install a Prism file on your own computer, click in the toolbar to select the file you want to install.

Note: Prism files may contain items that have been deleted. These items are listed under **Deleted Items** in the Editor window. When you install a Prism file on another computer, Prism Deploy deletes these items from the computer.

Creating and Editing Prism Files

Opening a Prism File

To open an existing Prism file for editing:

1. Click 📂 in the toolbar or select **File | Open**.

   The **Pick the file you want to edit** dialog box opens.

2. From the **Look in:** drop down list, select the directory where the file is located.
3. Click on a file name. Click **Open**.
4. When you open a package that has not been upgraded to the latest file format, the Upgrade Package Format dialog opens. This dialog gives you the option of converting the package to the latest format, which is compatible with Windows Vista.

Note: When upgrading a 32-bit package, it must be performed on a 32-bit machine.

5. Prism Deploy opens the selected file and displays it in the Editor window.

Creating a New Prism File

Creating a New Prism File Manually

You can create your own Package or Prism Executable Package and then add folders, files, shortcuts, and variables to the new Prism file.

1. Click 🗋 or select **File | New**.

   The **Pick a name for the new file** dialog box opens.

2. From the **Save in**: drop down list, select the directory where you want to save the file.
3. Type a name for the file in the **File name:** text box.
4. From the **Save as type:** drop down list, select the type of Prism file you want to save.
5. Click **Save**.
6. The **OS Requirements** dialog box opens. Select whether the target OS is 32-bit or 64-bit.
7. Prism Deploy creates a blank Prism file and displays it in the Editor window.

You can:

- Create a directory structure
- Add folders
- Add files
- Add shortcuts
- Add variable folders
- Define the file properties

Creating a Prism File from the Windows Desktop

Prism Deploy is completely integrated with the Windows interface, which means you can create a new Prism file directly from the Windows desktop.

1. Right-click anywhere in the Windows desktop.
2. Select the type of Prism file you want to build.
3. An icon for the new file appears on the desktop.
4. Right-click on the New Prism File icon. New Package.pwc
5. Select **Edit** from the drop-down menu to open the new file in the Editor.

Creating a Directory Structure

When you create a new Prism file from scratch, you need to recreate a directory structure in the Prism file. You can add drive letters or special folders consisting of common Windows directories such as Windows, System, Desktop, and Start menu. Once you have these high level items in place, you can add unique folders for an application's files, for example. You may also want to add drives and folders to existing Prism files.

Creating a Directory Structure

1. In the top left of the Editor window, click the icon next to the path name to the Prism file.

   This is the highest branch in the directory structure, and it can contain only these three virtual folders:

   — **Computer**: Contains items specific to the computer. It is a "virtual" folder in that it does not correspond to any specific folder on the computer. Items in this virtual folder are common across users.

   — **Current User**: Contains items specific to the current user. Like **Computer**, this is a virtual folder.

   — **Deleted Items**: This virtual folder defines files, folders, and registry keys to delete. These are partitioned into **Computer** and **Current User** virtual folders.

2. Click the icon next to the **Computer** node. From this node you can add:

   - **Special folders**: Add certain special folders, such as Windows, Program Files, and Users. Each of these can contain other special folders, such as Common Files and Resources.
   - **Drives**: Add a drive letter. A submenu lists drive letters A–Z. The **Computer** node is the only node where you can add a new drive letter.
   - **Registry branches**: Create the HKEY_LOCAL_MACHINE and HKEY_USERS registry branches.
   - **Variable folders** Create a folder that represents a variable that has been defined for the environment or within Prism Deploy. The New Variable Folder dialog box opens.

3. Click the icon next to the **Current User** node. From this node you can add:

   - **User Profile**: This is the single, top-level special folder associated with the **Current User** node. It can contain other special folders such as Desktop and Start Menu.
   - **Registry branch**: Create the HKEY_CURRENT_USER registry branch.

Adding Items

Many items may be added within the basic Prism File structure described above. To add items, right click on a node in the left side of the Editor window and select **New**. This will open a menu of items that can be added beneath that node.

- **Special folders**: Add special folders, such as Windows, Games, Video, and Fonts. These can be added only under the **Computer** node, **User Profile**, and other Special Folders. This can be done by right-clicking or by clicking in the toolbar.
- **Folders**: Add new folders. These can be added to most items in the Prism File. This can be done by right-clicking or by clicking in the toolbar.
- **Variable folders**: Create a folder that represents a variable that has been defined for the environment or within Prism Deploy. This can be done by right-clicking or by clicking in the toolbar.
- **Files**: Add files. These can be added to most items in the Prism File. This can be done by right-clicking or by clicking in the toolbar.
- **Shortcuts**: Add shortcuts. This can be done by right-clicking or by clicking in the toolbar.
- **Registry keys**: Add registry keys. This can be done by right-clicking or by clicking in the toolbar.
- **Registry values**: Add registry values. This can be done by right-clicking or by clicking ab in the toolbar.

Adding Special Folders

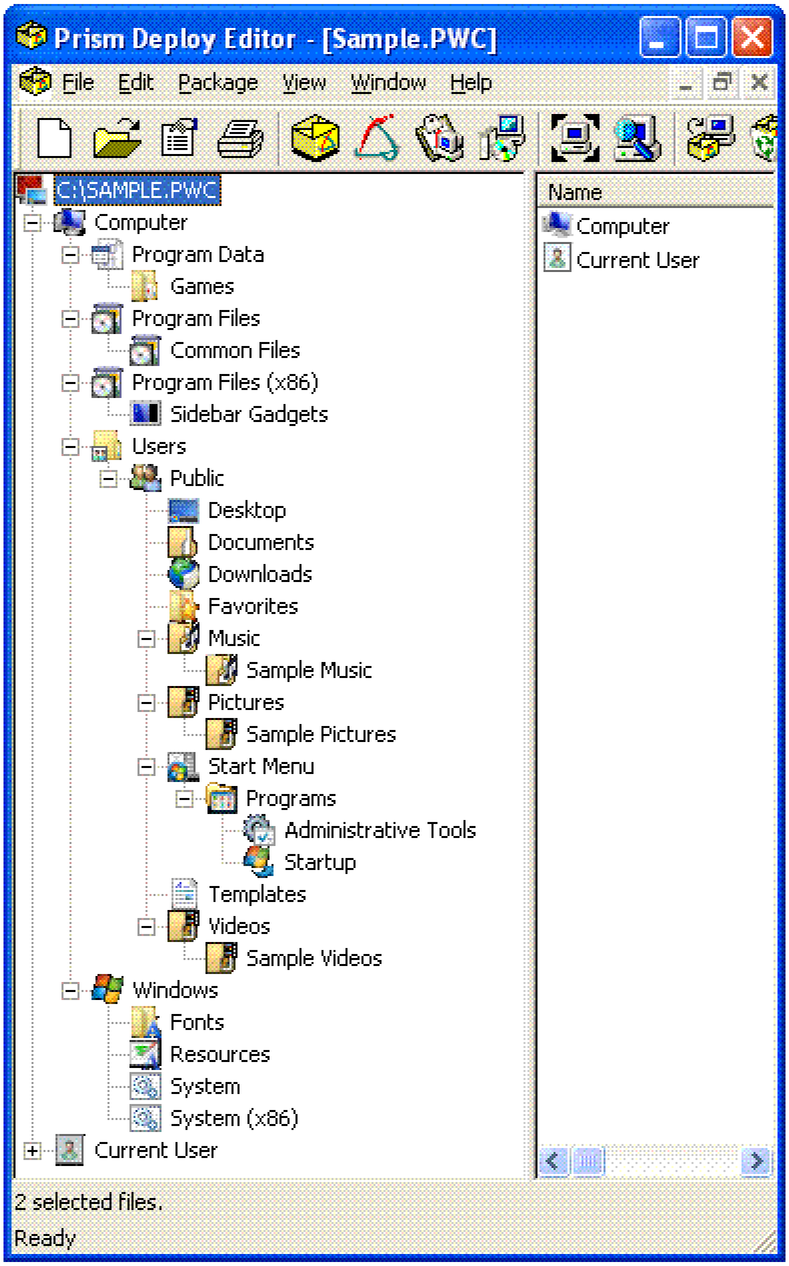
To add a special folder:

1. In the left side of the Editor window, click on the special folder or virtual folder (Computer or Current User) where you want to add a special folder.

   For example, click on **Current User** to add the special folder **User Profile**, or click **Computer** to add the special folder **Windows**.
2. Right-click the item and select **New**, or click in the toolbar
3. Select **Special Folder** from the submenu.
4. Select the desired special folder.

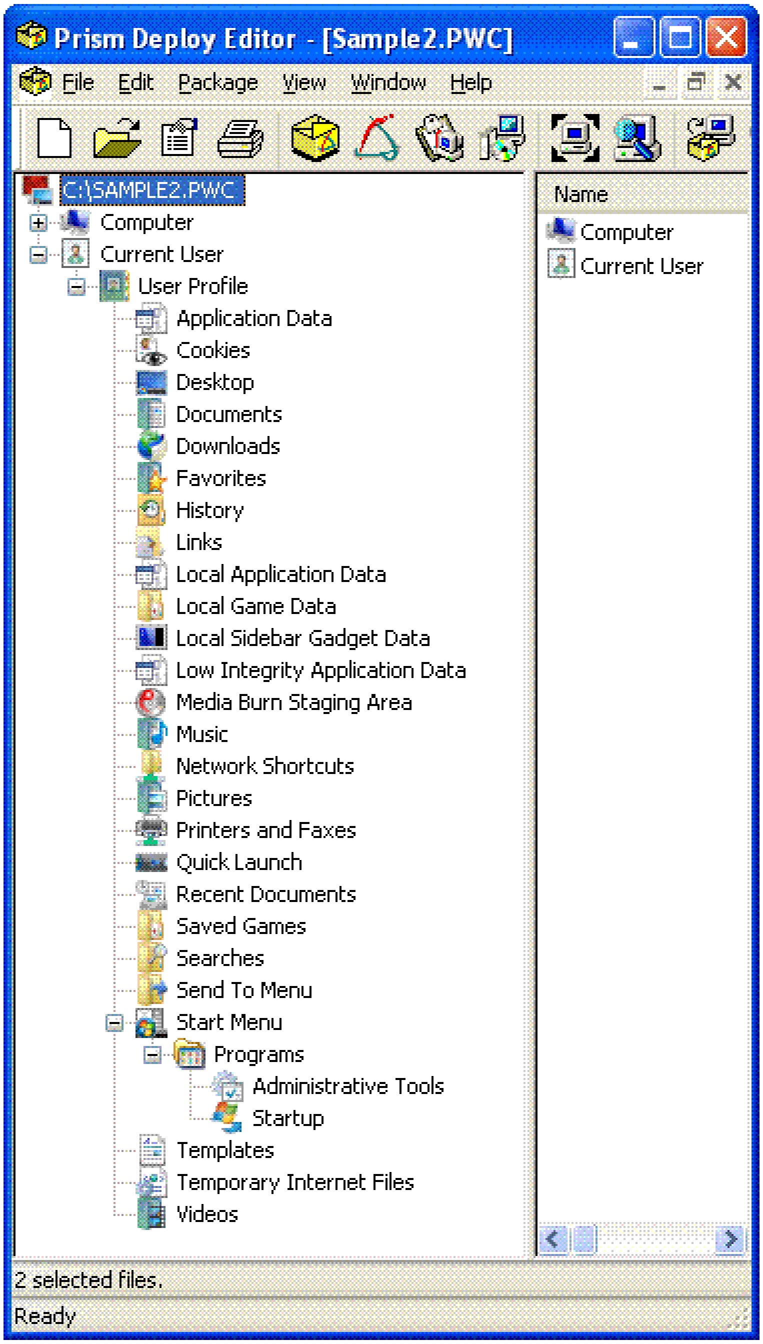
The Editor organizes the special folders hierarchically in the Editor window according to their logical containment relationships. For example, the Programs special folder appears under the Start Menu special folder and the Windows special folder appears under the Computer virtual folder. Therefore, the special folders that can be added depends on the item selected.

The images below show the hierarchical structure of the special folders under the **Computer** and **Current User** virtual folders.



2 selected files.

Ready



2 selected files.

Ready

Adding a New Folder

To add a new folder:

1. In the left side of the Editor window, click on the drive or folder where you want the new folder located.

   For example, click on **C:** to put the folder in the C: drive or click on **Windows** to put the folder in the Windows directory.

2. Click 📁 in the toolbar and select **Folder** from the submenu.

   The **New Folder Name** dialog box opens.

3. Type a name for the folder in the text box. Click **OK**. The new folder is displayed in the Editor window.

If the folder requires NTFS permissions, right-click on the folder and select **Properties**. Click on the Security tab and set the required permissions.

Adding a New Variable Folder

Prism Deploy lets you add a variable folder containing an environment variable to a Prism file.

1. In the left side of the Editor window, click on the drive or folder where you want the new variable folder located.

   For example, click on **C:** to put the folder in the C: drive or click on **Windows** to put the folder in the Windows directory.

2. Click 📁 in the toolbar and select **Variable Folder** from the submenu.

   The **New Variable Folder** dialog box opens.

3. In the text box, enter the name of the environment variable that defines the new folder. Click **OK**. The new variable folder is displayed in the Editor window.

Adding a New File to a Prism File

You can add any type of file on the computer to a Prism file.

1. In the Editor window, select the drive or directory where you want to add the new file.

2. Click 📄.

   The **Add Files** dialog box opens.

3. From the **Look in:** drop down list, select the directory where the file is located.

4. Click on a file name in the list box. Click **Open**.

The file, including its contents, is added to the Prism file.

Adding a Registry Key

Registry keys only exist in registry branches. Therefore you must first have a registry branch before you can add a registry key.

Creating a registry branch

There are three registry branches that can be created:

- **HKEY_USERS**: This registry is where Windows stores information about:
  - How the Windows desktop should look (the default desktop)
  - The customizations that each user has made to the Windows environment.

Information about individual users cannot be updated in this section of the registry. Only include this option in a Picture to update the default setup.

- **HKEY_LOCAL_MACHINE**: This registry is where Windows stores software- and hardware-specific information. Select this option when you are installing software, changing hardware, adding drivers, and so on.

- **HKEY_CURRENT_USER**: When a user starts Windows, the information about that user's desktop is copied from its storage location to Registry (Current User).

  Registry information for the current user is often updated during software installations, so include this option when you install software.

The first two of these registry branches, HKEY_USERS and HKEY_LOCAL_MACHINE, are created under the **Computer** node in the Editor window, while HKEY_CURRENT_USER is created under the **Current User** node. To create one of these branches, do the following:

1. Depending on which of the above branches you want to create:
   - Click next to **Computer**
   - Click next to **Current User**

2. Then do one of the following:
   - Click in the toolbar, and select the desired branch.

   3. Right click on the node, select **New**, and then select the desired branch.
      - For the **Computer** node, the branches are under the **New Registry Key** menu option.

Adding a registry key

Once a registry branch has been created, registry keys are added using the following steps:

1. Highlight the registry branch to which you wish to add a new registry key.
2. Do one of the following:
   - Click in the toolbar.
   - Right click on the registry branch, select **New**, and then select **Registry Key**.
3. The **New Registry Key Name** dialog box opens. Type a name for the registry key in the text box. Click **OK**. The new registry key is displayed in the Editor window.

Adding a Registry Value

To add a registry value:

1. In the Editor window, select a registry directory.

   You must be in a registry directory to create a registry value.

2. Click in the toolbar.
3. Type the name of the registry value in the text box or select the **Default Registry Value** check box to use the default value. (If the selected registry key already has a default value, this check box is not available.)
4. Click **OK**. The Registry Value dialog box opens.

Adding a Shortcut: Define a Target

1. In the left side of the Editor window, click on the drive or folder where you want the new shortcut located (for example, Start Menu Programs).
2. Click .

   The **New Shortcut Expert** opens.

3. Type the path to the application executable (*.EXE) file in the **File Name:** text box or click the **Browse** button to browse to the file.
4. Click **Next** to enter a description for the shortcut.

Importing Registry Files

Prism Deploy lets you import and export registry files. The following are a few examples of when you might want to use this feature:

- If you have been using registry files to make registry changes before purchasing Prism Deploy, you can import the registry files into a Prism file to distribute the changes. Prism Deploy also lets you uninstall the changes.
- You accidentally deleted a key branch of a Prism file and you do not have a backup of the original Prism file. Using REGEDIT on the computer where you created the Prism file, export that branch of the registry to a registry file, then import the registry file into the Prism file.

To import a registry file:

1. Select **Edit | Import Registry File**.

   The **Choose a registry file to import** dialog box opens.
2. Locate the registry file (it has a .REG extension). Click **Open**.
3. Prism Deploy imports the registry key and displays it in the Editor window.

Exporting Registry Files

Prism Deploy lets you import and export registry files. The following are a few examples of when you might want to use this feature:

- If you have been using registry files to make registry changes before purchasing Prism Deploy, you can import the registry files into a Prism file to distribute the changes. Prism Deploy also lets you uninstall the changes.
- You accidentally deleted a key branch of a Prism file and you do not have a backup of the original Prism file. Using REGEDIT on the computer where you created the Prism file, export that branch of the registry to a registry file, then import the registry file into the Prism file.

To export a registry file:

1. In the Editor window, select the registry branch you want to export.
2. Select **Edit | Export Registry File**.

   The **Pick a name for the new registry file** dialog box opens.
3. Select the location where you want to save the registry branch and enter a name for it. Click **Save**.
4. Prism Deploy exports the registry branch to a registry file that Windows can read. If you double click on the registry file, it is merged into the computer’s registry. **BE CAREFUL WITH THIS OPTION!**

Adding Items from the Desktop and Windows Explorer

Prism Deploy lets you add a folders, files, or shortcuts directly from the desktop or from Windows Explorer.

Locate the item you want on the desktop or in Windows Explorer. Then drag and drop it into the Editor window. The item remains on your hard drive and it is added to the Prism file.

Caution: On Windows Vista, if you execute the Editor with full administrative privilege (for example, using the “Run as Administrator” context menu), then you will not be able to add items to a package via drag and drop. The Editor does not normally need to be launched with special privileges.

Customizing Files for a Diverse Environment

Creating a Directory Structure

When you create a new Prism file from scratch, you need to recreate a directory structure in the Prism file. You can add drive letters or special folders consisting of common Windows directories such as Windows, System, Desktop, and Start menu. Once you have these high level items in place, you can add unique folders for an application’s files, for example. You may also want to add drives and folders to existing Prism files.

Creating a Directory Structure

1. In the top left of the Editor window, click the icon next to the path name to the Prism file.

   This is the highest branch in the directory structure, and it can contain only these three virtual folders:

   — **Computer**: Contains items specific to the computer. It is a "virtual" folder in that it does not correspond to any specific folder on the computer. Items in this virtual folder are common across users.

   — **Current User**: Contains items specific to the current user. Like **Computer**, this is a virtual folder.

   — **Deleted Items**: This virtual folder defines files, folders, and registry keys to delete. These are partitioned into **Computer** and **Current User** virtual folders.

2. Click the icon next to the **Computer** node. From this node you can add:

   - **Special folders**: Add certain special folders, such as Windows, Program Files, and Users. Each of these can contain other special folders, such as Common Files and Resources.
   - **Drives**: Add a drive letter. A submenu lists drive letters A–Z. The **Computer** node is the only node where you can add a new drive letter.
   - **Registry branches**: Create the HKEY_LOCAL_MACHINE and HKEY_USERS registry branches.
   - **Variable folders** Create a folder that represents a variable that has been defined for the environment or within Prism Deploy. The New Variable Folder dialog box opens.

3. Click the icon next to the **Current User** node. From this node you can add:

   - **User Profile**: This is the single, top-level special folder associated with the **Current User** node. It can contain other special folders such as Desktop and Start Menu.
   - **Registry branch**: Create the HKEY_CURRENT_USER registry branch.

Adding Items

Many items may be added within the basic Prism File structure described above. To add items, right click on a node in the left side of the Editor window and select **New**. This will open a menu of items that can be added beneath that node.

- **Special folders**: Add special folders, such as Windows, Games, Video, and Fonts. These can be added only under the **Computer** node, **User Profile**, and other Special Folders. This can be done by right-clicking or by clicking in the toolbar.
- **Folders**: Add new folders. These can be added to most items in the Prism File. This can be done by right-clicking or by clicking in the toolbar.
- **Variable folders**: Create a folder that represents a variable that has been defined for the environment or within Prism Deploy. This can be done by right-clicking or by clicking in the toolbar.
- **Files**: Add files. These can be added to most items in the Prism File. This can be done by right-clicking or by clicking in the toolbar.
- **Shortcuts**: Add shortcuts. This can be done by right-clicking or by clicking in the toolbar.
- **Registry keys**: Add registry keys. This can be done by right-clicking or by clicking in the toolbar.
- **Registry values**: Add registry values. This can be done by right-clicking or by clicking ab in the toolbar.

Changing File Path Names

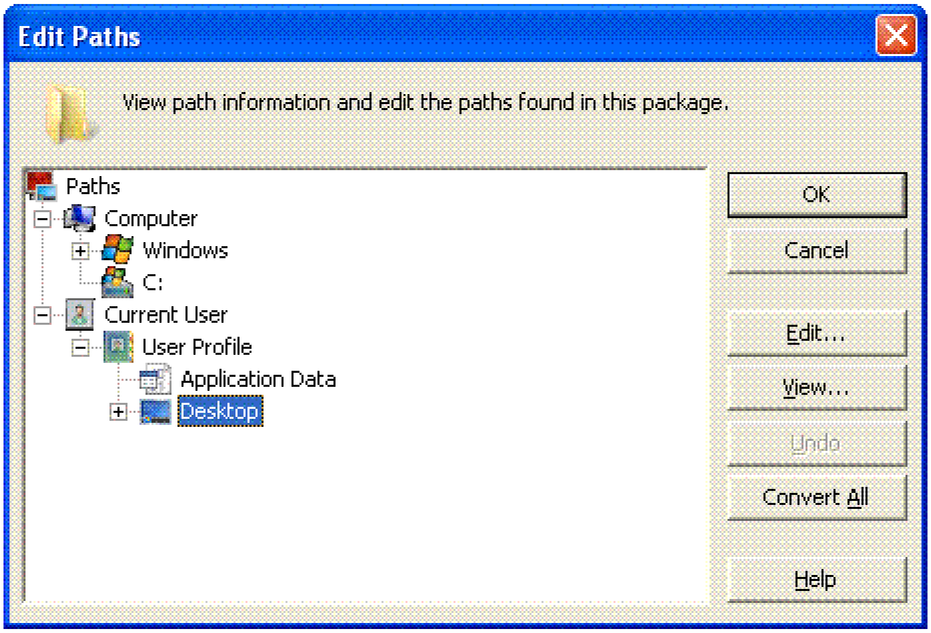
If you are installing files on computers that are on a network or to your computer at home, chances are they are not configured identically. Prism Deploy’s paths utility tracks and changes path information so you can quickly customize the installed files for their new location. In the **Edit Paths** dialog box, you can view path information and edit the paths.

Viewing Path Information

To view the path information for a file:

1. In the Editor window, click **C:** in the toolbar or select **Edit** | **Paths**.

   Prism Deploy takes a moment to scan through the file for path references. When it finishes, the **Edit Paths** dialog box opens showing you the path references.



2. To see where a path is referenced, highlight a directory. Click the **View** button.

   The Path Information dialog box opens.

3. Click the **Convert All** button to convert all paths to the latest (Vista-compatible) format. For more information see Windows Vista.

Changing Path Information

To change the path information:

1. In the **Edit Paths** dialog box, click on the directory that you want to change.

   Prism Deploy updates the path for the directory that you select, as well as any files or subfolders that are below it in the directory structure.

2. Click the **Edit** button.

   The Edit Path dialog box opens. When you have changed the path name, (**changed)** is listed next to statements that have been affected by the change.

> **Note**: Folders that contain Special Folders cannot have their paths edited. In these cases the **Edit** button will be disabled.

3. To undo changes, select **Undo** in the **Edit Paths** dialog box.

4. Click **OK** when all changes have been made. Prism Deploy updates the affected files to reflect the changes.

Working with Variables

Prism Deploy lets you use environment and Prism Deploy variables inside registry settings, shortcuts, path names, and INI files. This feature lets you install Prism files even when the people receiving the file need personalized entries distributed, such as e-mail addresses. Variables also let you install one Prism file in a heterogeneous environment.

Prism Deploy recognizes some common folders and registry keys and automatically substitutes a variable for the path names or keys when it takes a Picture or finds changes. If Prism Deploy does not substitute a variable for a specific instance of a folder or key, you can include variables in the file properties or add a variable folder to the Prism file. Click the options listed below for an overview.

Variable Properties dialog

Use the **Variable Properties** dialog box to define the name, default value, source, and requirements (including user prompts) of a variable. To open the **Variable Properties** dialog box, click the **Add** button in the Variables tab of the File Properties dialog box.

Tabs on this dialog box include:

- Name tab
- Requirements Tab
- Source Tab
- User Prompt Tab

Variable Resolution Order

There are four ways a variable can receive a value. Prism Deploy uses the steps defined below, in the order given, to obtain the value for a variable when you install the Prism file containing the variable. The variable receives a value from:

1. The source you have defined (if any) in the Variable Source tab of the Variable Properties dialog box.
2. If the source you have defined is **This Variable**, Prism Deploy tries to get its value from the currently executing environment. Note that Prism Deploy inherits this environment from the process that launches it. Checking the value of the variable in another, parallel process (such as a command prompt window) may not show the same value that Prism Deploy will inherit.
3. If the variable is not defined after the previous steps, Prism Deploy uses the default value, if any, you defined in the Variable Name tab of the Variable Properties dialog box.
4. If appropriate, based on the settings you have defined in Variable User Prompt tab of the Variable Properties dialog box, Prism Deploy displays a window asking for the variable’s value. The window also contains a **Cancel** button, letting the user cancel the operation at that point.
5. Finally, Prism Deploy will decide to continue installing the file based on whether the length of the variable’s value (0 if it does not have one) meets the minimum and maximum length properties you have defined in the Variable Requirements tab of the Variable Properties dialog box.

Customizing the Prism File for Multiple Users

For target computers that are configured for multiple users, you can further refine your Prism file to allow for unattended installation or to customize the file for an individual user. The primary settings for these scenarios are in the File Properties | Requirements tab and the Variable Properties | Requirements tab. The settings for all users versus the current user can also be adjusted with the command line options, giving a wide array of combinations.

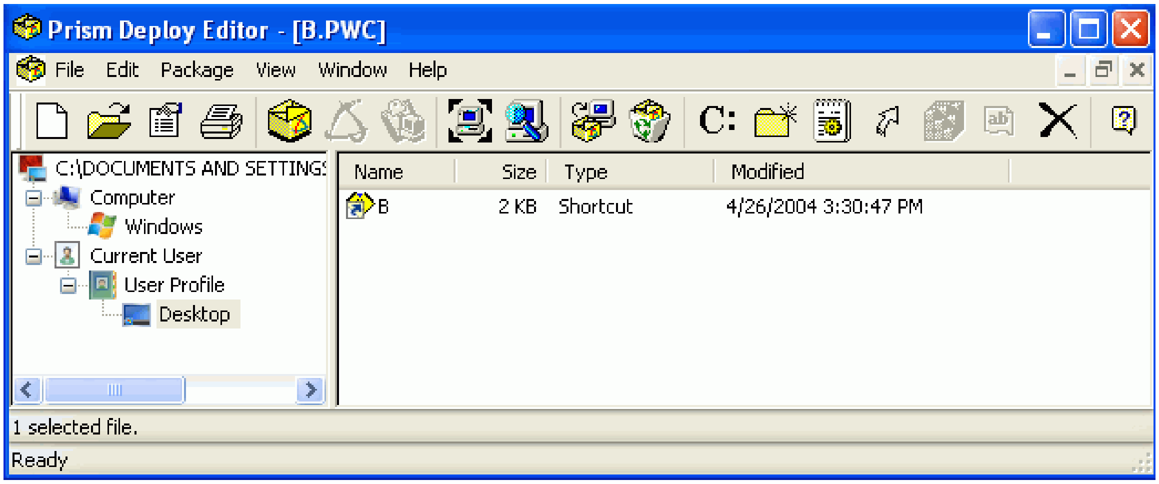
The property settings and command line options let you:

- Perform an unattended installation. Prism Deploy lets you prepare a Prism file for unattended installation after hours or in the background while someone is logged on. The file is installed on the computer immediately. All users will get the user specific changes the next time they log on. Prism Deploy takes the drudgery out of repeated installation for each user set up on the target machine.
- Uninstall the Prism file from any user login. To uninstall a file, you do not have to be logged on as the user who installed it originally.
- Update all users on the computer through a single installation of the Prism file and avoid the problem of reinstalling the same settings repeatedly, under each user account.
- Uninstall a file to remove all changes for all of the users, at one time. Or, use the command line option /CurrentUser to uninstall a Prism file from only one user, without disturbing the settings of the other users. (See "Scripting Prism Deploy Tasks" in the *Deployment Guide* or the Prism Console online Help for information on the command line options.)

Editing Prism Files

Editor Main Window

When a Picture, Package, or Prism Executable Package is open, the Prism Deploy Editor window displays the file in much the same way as Windows Explorer. The contents of a Prism file include many types of items, including files, folders, special folders, registry keys, and drives.



Name	Size	Type	Modified
B	2 KB	Shortcut	4/26/2004 3:30:47 PM

1 selected file.

Ready

Layout

The Editor window consist of a **toolbar** at the top, a **tree view** of the file on the left side, a **detail pane** on the right side, and a **status bar** at the bottom of the page.

- **Toolbar**: The toolbar provides quick access to commonly used functions. If a toolbar function is not applicable to the currently selected item, its icon will be disabled. Click here for a complete list of icons in the toolbar and their meaning.
- **Tree view:** The left side of the Editor window displays a tree view of the file. Prism Deploy organizes the tree into a hierarchical structure. Click the **+** or **-** signs to expand or collapse the items in the tree. Items in the tree may be special folders, registry branches, folders, or other item types described below.
- **Detail pane:** The right side of the Editor window displays the contents of the selected item in the tree view. Click on a column label to sort the items by name, size, type, or date modified.

- **Status bar:** The Status Bar provides status on your current selection and actions.

Contents

There are a variety of item types that constitute a Prism file:

- **Computer**: Contains items specific to the computer. It is a "virtual" folder in that it does not correspond to any specific folder on the computer. Items in this virtual folder are common across users.
- **Current User**: Contains items specific to the current user. Like **Computer**, this is a virtual folder.
- **Deleted Items**: This virtual folder defines files, folders, and registry keys to delete. These are partitioned into **Computer** and **Current User** virtual folders.
- **Special folders:** Special Folders represent common directories on a Windows computer, such as Windows, Desktop, Application Data, Common Programs, and Start Menu. Each special folder is associated with either the Computer (common) or Current User (per user) virtual folder. Special folders may contain folders, variable folders, files, and shortcuts. Some special folders may also contain other special folders. For example, the Windows special folder contains the System special folder.
- **Registry branches:** Registry branches contain registry keys and values. These appear directly under the **Computer** and **Current User** virtual folders. For example, the HKEY_LOCAL_MACHINE branch belongs to the **Computer** virtual folder, and the HKEY_CURRENT_USER branch belongs to the **Current User** virtual folder.
- **Drives:** A computer's drives appear directly under the **Computer** virtual folder. This is the only place they may appear.
- **Folders:** Folders denote folders on the computer, and may appear within drives, special folders, and other folders.
- **Variable folders:** Variable folders represent a variable that has been defined for the environment.
- **Files:** Files on the computer. Files appear only in the detail pane. Icons associated with files have special interpretations in the Editor:

  In Picture files, grayed items indicate that only file information is stored, not the file contents.

  Packages and Prism Executable Packages always contain file contents unless you created them by comparing two Pictures that contained only file information or a file was busy when it was created.

  If the item is not grayed, the file contents and file information are stored in the Prism file.

  A red X on an item in a Prism file means that Prism Deploy discovered that the item has been deleted from the system. The file is removed from the target computer when the Prism file is installed.

- **Shortcuts:** Shortcuts to files on the computer. These appear only in the detail pane.
- **Registry keys:** Registry keys defined in a registry branch or within another registry key.
- **Registry values:** The registry value associated with a registry key. These appear only in the detail pane.

Opening an Item within a Prism File

Prism Deploy displays the items contained in a Prism file the same way Windows Explorer does.

- If the Prism file stored a file's contents (the file's icon in the Editor window is not grayed), you can open the file directly from Prism Deploy.
- If it is a text file, Prism Deploy opens the file in the default editor (usually Notepad).
- If the file has a recognizable extension, such as .DOC, Prism Deploy launches the file in the application in which it was created (for example, Microsoft Word).
- If it is an executable (*.EXE) file, Prism Deploy tries to launch it.

To open a file:

- Right-click on its name and select **Open** from the drop down menu. Prism Deploy opens the file.
- If you want to edit an executable file such as AUTOEXEC.BAT, right click on the file and select **Edit**.
- If the file does not have an association with an editor or an application, the **Open With** option is available.

Windows opens a dialog box letting you select an application to associate with the file.

Finding Text in Prism File Items

Prism Deploy lets you find text in any item in the Prism file that is currently open in the Editor window. For example, when you install software, the setup program usually asks you to enter your name. You may also have to enter your company name, an e mail user name, and who knows what else. Prism Deploy can find all instances of your name in a Prism file. This feature is also useful if you just want to see where Windows or applications store certain information.

To find text in the items in a Prism file:

1. Select **Edit | Find**.
2. Type the text you want to look for in the **Find What:** text box.
3. Select whether to **Match whole word only** or **Match case**. You can select one or both of these options.
4. Select the appropriate check boxes to indicate where you want Prism Deploy to look.

   You can select as many options as you would like. If you are not sure where the information is located, you select all the options. However, selecting all the options increases the time it takes to look.
5. Click **Find Next**.

   As Prism Deploy searches for the text, an animated magnifying glass is displayed in the **Find** dialog box.

   If Prism Deploy finds the text in the Editor window, it highlights the item. For example, if you searched for your name, Prism Deploy would highlight the registry value "User" with your name in the Data column of the Editor window. If Prism Deploy finds the text within a file, such as an INI file, it shows the text at the bottom of the **Find** dialog box.
6. Click **Find Next** to find the next instance of the text or click **Cancel** to close the **Find** dialog box.

Replacing Text in Prism File Items

Prism Deploy lets you find and replace text in any item in the Prism file that is currently open in the Editor window. For example, when you install software, the setup program usually asks you to enter your name. You may also have to add a network user name, an e mail user name, and who knows what else. You have recently changed your name and want to update all those user and "registered to" names. Prism Deploy can find all instances of your name in a Prism file and replace them with your new name.

To find and replace text:

1. Select **Edit | Replace**.
2. Type the text you want to look for in the **Find What:** text box.
3. Enter the text you want to use to replace the existing text in the **Replace With:** text box.
4. Select whether to **Match whole word only** or **Match case**. You can select one or both of these options.
5. Select the appropriate check boxes to indicate where you want to Prism Deploy to look.

   You can select as many options as you would like. If you are not sure where the information is located, select all the options. However, selecting all the options increases the time it takes to look.
6. Click **Find Next** to find each instance, then click **Replace** to replace the selected text. Or, click **Replace All** to replace all instances of the text.

Paths

Check the Paths Embedded in the Package

With the Package open in the Editor, click **C:** in the toolbar. The Editor will show you every path that is referenced anywhere in the Package (including targets of shortcuts, registry data, INI file entries, and so on).

Sometimes you will discover there is a path that needs to be changed. For example, there may be a pointer to C:\DOCUMENTS AND SETTINGS\ADMINISTRATOR\FAVORITES. You probably want to change the "Administrator" portion of the path to a variable so that the logged-in user’s name is inserted into the path when the Package is installed. The Edit Paths dialog box is the place to make that change.

Changing File Path Names

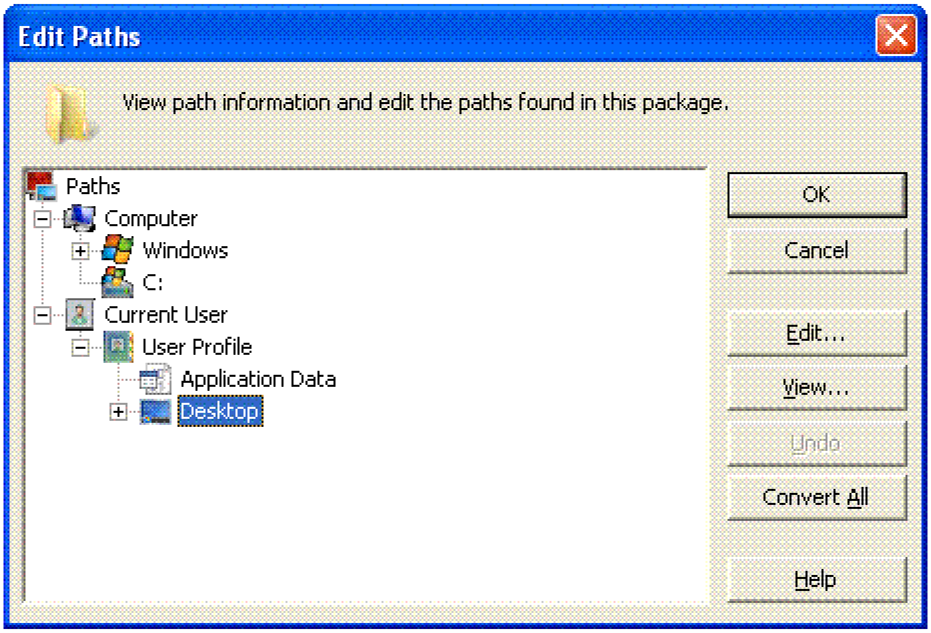
If you are installing files on computers that are on a network or to your computer at home, chances are they are not configured identically. Prism Deploy’s paths utility tracks and changes path information so you can quickly customize the installed files for their new location. In the **Edit Paths** dialog box, you can view path information and edit the paths.

Viewing Path Information

To view the path information for a file:

1. In the Editor window, click C: in the toolbar or select **Edit** | **Paths**.

   Prism Deploy takes a moment to scan through the file for path references. When it finishes, the **Edit Paths** dialog box opens showing you the path references.



View path information and edit the paths found in this package.

2. To see where a path is referenced, highlight a directory. Click the **View** button.

   The Path Information dialog box opens.

3. Click the **Convert All** button to convert all paths to the latest (Vista-compatible) format. For more information see Windows Vista.

Changing Path Information

To change the path information:

1. In the **Edit Paths** dialog box, click on the directory that you want to change.

   Prism Deploy updates the path for the directory that you select, as well as any files or subfolders that are below it in the directory structure.

2. Click the **Edit** button.

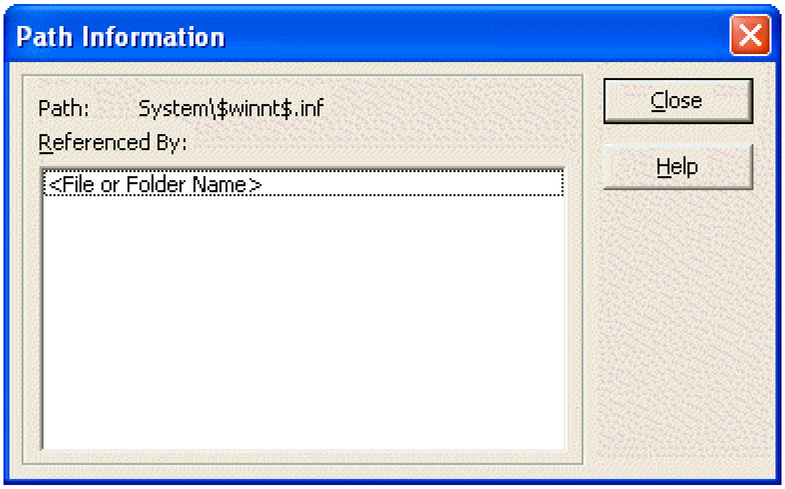
   The Edit Path dialog box opens. When you have changed the path name, (**changed)** is listed next to statements that have been affected by the change.

Note: Folders that contain Special Folders cannot have their paths edited. In these cases the **Edit** button will be disabled.

3. To undo changes, select **Undo** in the **Edit Paths** dialog box.

4. Click **OK** when all changes have been made. Prism Deploy updates the affected files to reflect the changes.

Path Information dialog



Path: System\$winnt$.inf

Referenced By:

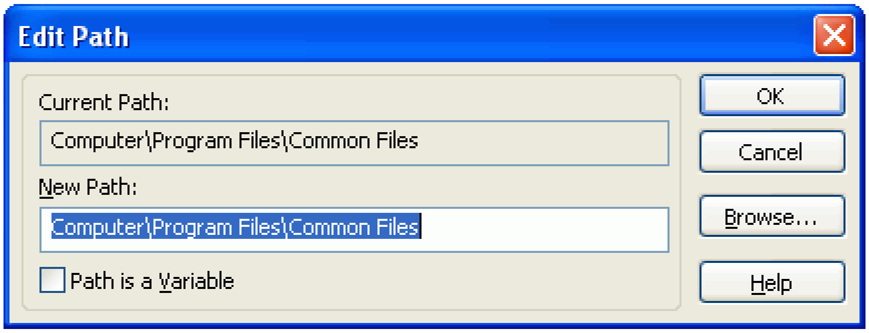
<File or Folder Name>

The **Path Information** dialog box displays information such as the section and key of a registry entry referenced in the selected path.

You may also see variables in some paths. Prism Deploy automatically substitutes the variable **windir** for a Windows directory and **winsysdir** for a System directory, among other variables. This substituted variable lets you more easily install the file for users in a heterogeneous environment.

Select **Close** to exit the **Path Information** dialog box and return to the Edit Paths dialog box.

Edit Path dialog



Current Path:

Computer\Program Files\Common Files

New Path:

Computer\Program Files\Common Files

□ Path is a Variable

The **Edit Path** dialog box lets you change a path name. Prism Deploy updates the path that you select, as well as any paths that are below it in the directory structure.

The Edit Path dialog box opens when you click the **Edit** button on the Edit Paths dialog box.

- **New Path**: Type the path to which you want to change the current statement in the text box or click the **Browse** button to select a new path.

Note: Always enter the full path when specifying the new path.

- **Path Is a Variable:** Select the check box if the new path is to be a variable, for example WINDIR. Click **OK**.

  The Variable Properties dialog box opens.

Renaming and Deleting

Renaming an Item in a Prism File

You can rename any of the items in a Prism file.

1. In the right side of the Editor window, right click on the item that you want to rename. Select Rename from the drop down menu.
2. The item’s name is highlighted. Type a new name for the item and press <Enter> to replace the old name.

Deleting an Item from a Prism File

You can delete any of the items in a Prism file. If you delete a folder, the contents of the folder are also deleted.

1. In the Editor window, click on the item that you want to delete.
2. Select **Edit | Delete** or click ✕ in the toolbar.
3. A verification window opens to confirm the deletion. Click **OK**.
4. The item is deleted from the Prism file.

Printing a Prism File

Printing the Contents of a Prism File

When you print a Prism file, Prism Deploy prints a list of the file contents, including the file name, size, time, and date.

1. Click 🖨 or select **File | Print**.

   The **Print Contents** dialog box opens.

2. Select the **Print the Entire File** or the **Print the Selected Branch** option. The branch may be a drive letter, folder, or a registry branch.

   Depending on the option you selected, different **File Content Options** will be displayed. You may select none, one, or both of the **File Content Options**. However, we suggest selecting at least one of the two options. Otherwise, the report will contain only a header with no text.

   - **Print the Entire File**: Select one of these options:
     - **Print File and Folder Information** to print the drives, folders, and file names within the Prism file.
     - **Print Registry Information** to print the registry branches within the Prism file.
   - **Print the Selected Branch**:

     If you select this option and the branch is a drive letter or folder, you can:

     - **Print all Files and Subfolders**: Print a list of all the files and subfolders in the selected drive or folder.
     - **Print only the Files in this Folder**: Print a list of the files in the selected folder.

     If you select Print the Selected Branch and the branch is a registry key, you can:

- **Print all Values and Subkeys**: Print a list of all the registry values and subkeys in the selected registry branch.
- **Print Only the Values in this Key**: Print just the values and data for the selected registry key.

3. Click **OK** to print the selected information. It prints on the default printer.

Tip: If you wish to print the contents to a text file, install a Generic/Text Only printer and direct the print job to a file. A dialog box will ask where to save the file when printing.

Defining Prism File and Item Properties

Prism File Properties

Defining Prism File Properties

Prism Deploy lets you enter a detailed description for a Picture, Package, or Prism Executable Package, specify installation requirements, and define options for installing the file on other computers. For example, you can specify that Prism Deploy should install the file only when certain disk space requirements are met.

To define properties for an open file, click 📋 in the toolbar or select **File | Properties**.

> **Note**: The **File Properties** dialog box defines default properties for the Prism file currently displayed in the Editor window. You can also define file properties for each individual item within a Prism file. (See Item Properties.)

File Properties for Packages and Pictures

When you select 📋 with a Prism Deploy Package or Picture open in the Editor window, the following tabs are available.

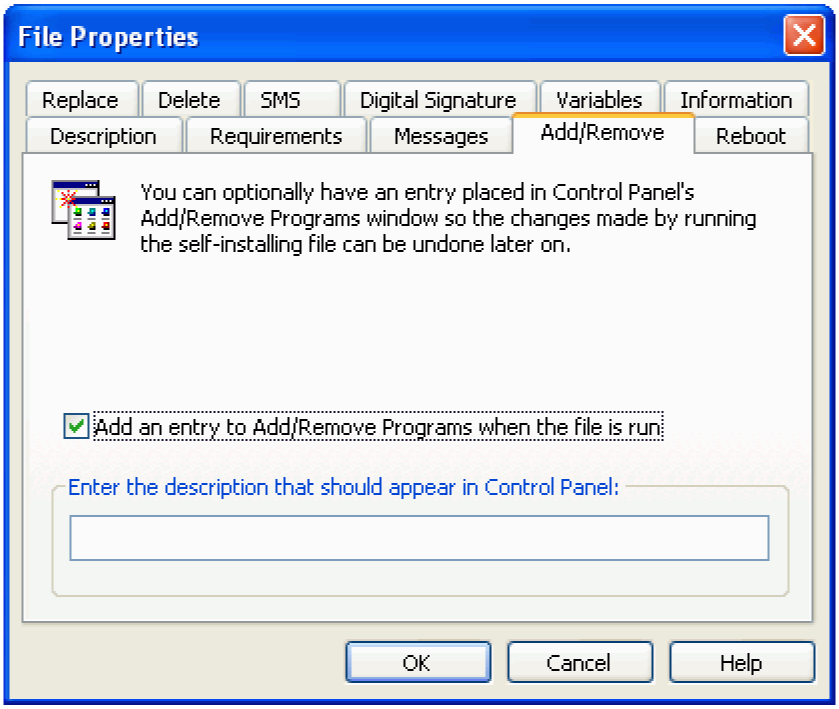
- Delete tab
- Description tab
- Digital Signature tab
- Information tab
- Messages tab
- Reboot tab
- Replace tab
- Requirements tab
- SMS tab
- Variables tab

File Properties for Prism Executable Packages

When you select 📋 with a Prism Executable Package open in the Editor window, the File Properties include all of the tabs listed above, plus:

- Add/Remove tab

Add/Remove Programs Tab



You can optionally have an entry placed in Control Panel's Add/Remove Programs window so the changes made by running the self-installing file can be undone later on.

☑ Add an entry to Add/Remove Programs when the file is run

Enter the description that should appear in Control Panel:

The **Add/Remove** tab lets you define whether the user can undo the changes made by the Prism Executable Package. If you select this option, the name that you specify is added to the Windows Control Panel’s **Add/Remove Programs** window. If the user highlights an entry in the Control Panel that was added by Prism Deploy and selects **Add/Remove**, the changes are uninstalled.

Caution: If you install the Prism Executable Package through the Prism Client or from within the Prism Deploy program itself (by selecting **Package | Install** or clicking), it is treated as a Package.

The contents can be uninstalled via the Uninstall file. However, an entry is not added to the Add/Remove list in the Windows Control Panel.

- **Add an entry to Add/Remove Programs when the file is run**: Select this option to tell Prism Deploy to add an entry in the computer's Control Panel Add/Remove Programs list during the Prism file installation.

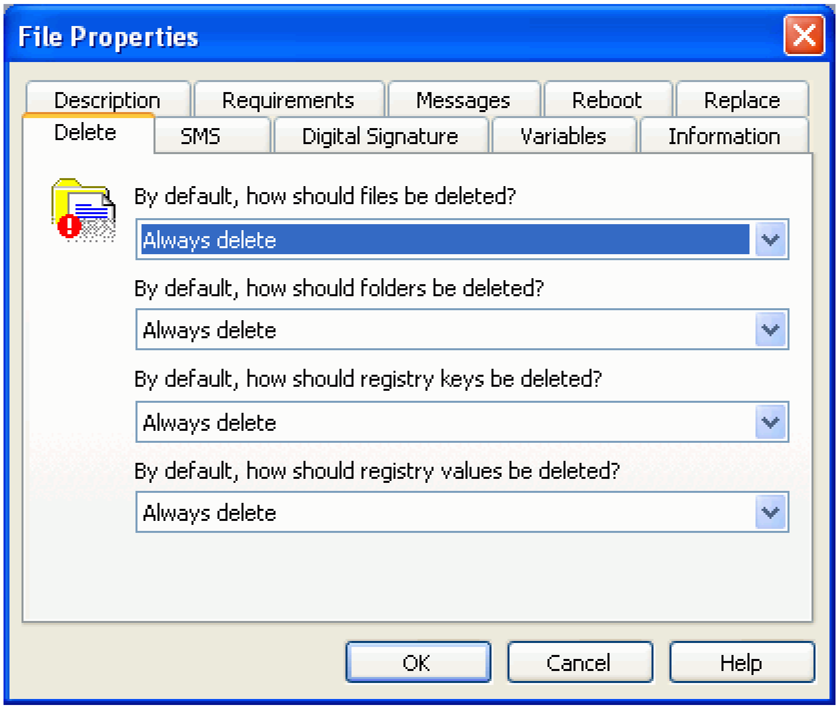
  If you select this option and you are distributing an application that adds its own entry into the Control Panel’s **Add/Remove Programs**, Prism Deploy adds a second entry. For this reason, we recommend making the description unique so that the user can tell if the program or Prism Deploy made the entries.

  Note: If you are delivering configuration changes, new files or folders, or some other type of update that does not automatically create an uninstaller, you must select the check box for **Add an entry to Add/Remove Programs when the file is run**.

  You must also enter the description to be entered in the Add/Remove Programs area of the Control Panel, if you want to be able to uninstall the software.

- **Description**: Enter a description for display in the Control Panel's Add/Remove Programs list on each target computer.

Delete Tab



By default, how should files be deleted?
Always delete

By default, how should folders be deleted?
Always delete

By default, how should registry keys be deleted?
Always delete

By default, how should registry values be deleted?
Always delete

Prism files may contain items that were deleted from the computer. When you install the Prism file, Prism Deploy deletes items with the same name from the target system. This tab lets you define separate options for when files, folders, registry keys, and registry values should be deleted.

Deleting Files

- **Delete if file is identical**: Delete an existing file only if it is identical to one in the Prism file. Prism Deploy compares the name, date/time, and version of the files.
- **Delete if file is different**: Delete a file only if it is different from an existing file in any way.
- **Always delete**: Always delete files. This option is the default.
- **Never delete**: Never delete files.

Deleting Folders

- **Delete only if empty**: Delete folders only if they are empty.
- **Always delete**: Always delete folders, even if they contain files or subfolders. This option is the default.
- **Never delete**: Never delete folders.

Deleting Registry Keys

- **Delete only if empty**: Delete the registry key only if it is empty. It does not contain any other registry keys or values.
- **Always delete**: Always delete registry keys, even if it contains other registry keys or values. This option is the default.
- **Never delete**: Never delete registry keys.

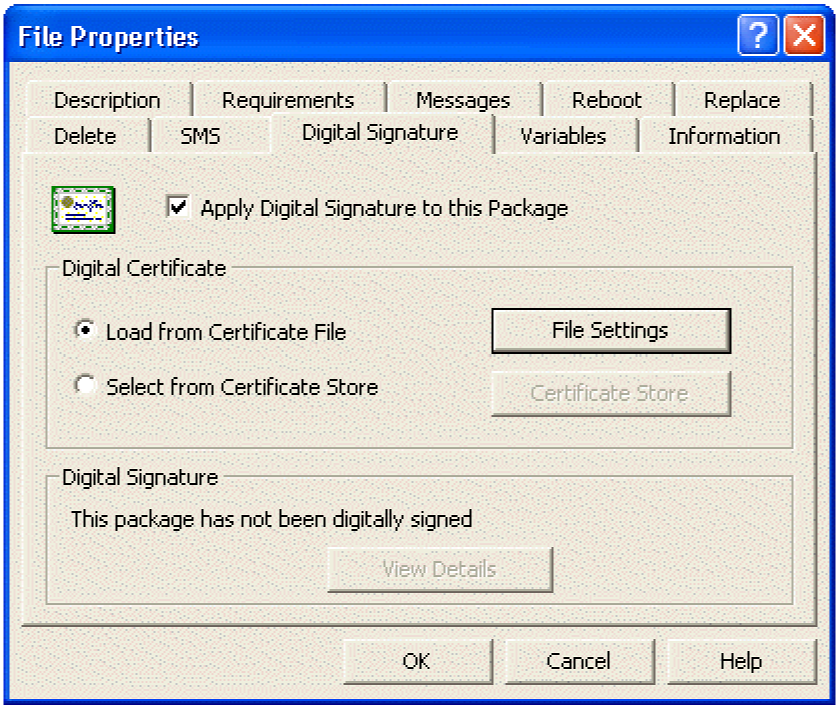
Deleting Registry Values

- **Delete if value is identical**: Delete a registry value only if it is identical to an existing value.
- **Delete if value is different**: Delete a registry value only if it is different from an existing value in any way.
- **Always delete**: Always delete registry values. This option is the default.
- **Never delete**: Never delete registry values.

File Description tab

Use the **Description** tab to enter a detailed description of the Prism file. This description could be helpful for differentiating between two similar files. The description is for informational purposes only and is not displayed to the end user.

Digital Signature tab



☑ Apply Digital Signature to this Package

Digital Certificate

● Load from Certificate File

○ Select from Certificate Store

Digital Signature

This package has not been digitally signed

The File Properties | Digital Signature tab lets you sign a Prism file with an X.509 certificate. When the Prism file is installed, the Prism Client uses the Package Authentication Policy to grant permission for installing the file.

Note: Before using this tab, you should be familiar with file authentication and understand the use of certificates. Prism Deploy can insure the authorship of a Prism file only if the certificate is secure and has been set up correctly before assigning it on this tab.

Note: The Prism file is not signed until it has been saved by closing the file or exiting from the Editor.

Options include:

☑ **Apply Digital Signature to this Package**: With this option selected, the Prism file currently open in the Editor is digitally signed with the certificate that you select below. The Client computers must be configured to accept the certificate used to sign this Prism file.

Digital Certificate

- **Load from Certificate File**: Use a certificate file to sign the Prism file. The certificate and private key must be combined in a single PFX file.

  Click the **File Settings** button to select a certificate file. The Certificate File Selection dialog box opens.
- **Select from Certificate Store**: Choose a certificate from a Certificate Store to sign the Prism file.

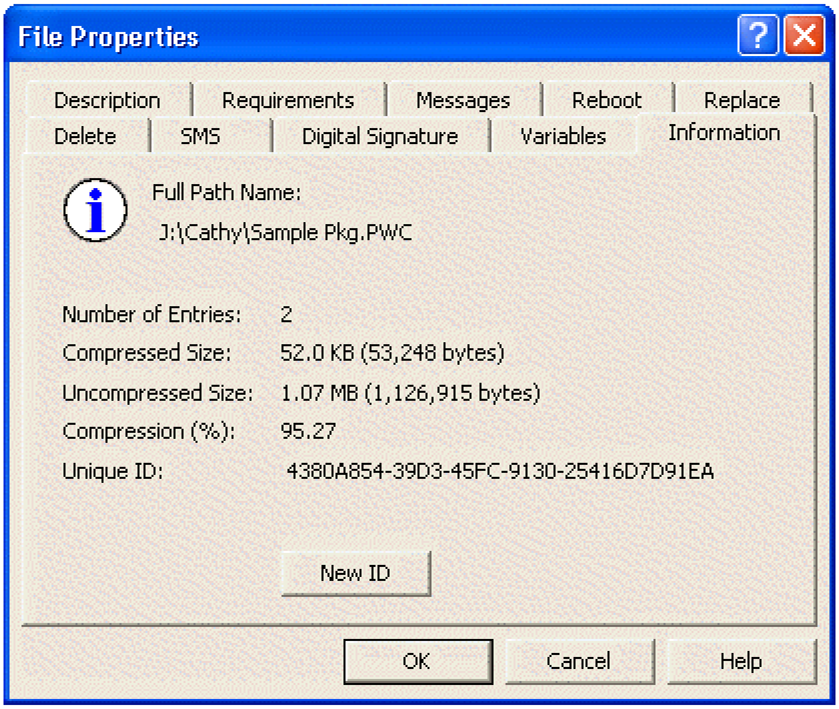
  Click the **Certificate Store** button to select a certificate from the Certificate Store. The Select Certificate from Store dialog box opens.

Digital Signature

- **View Details**: View the details of the certificate used to sign the Prism file, including its Certification Authority (CA), level of trust, and certificate thumbprint. When you click this button, a Windows dialog box opens displaying detailed information about the certificate.

  If you changed the certificate file on this dialog box, the details window shows the certificate file in use the last time the Prism file was signed. Details for the new certificate are not displayed until you close and re-open the Prism file.

Information Tab



Full Path Name:
J:\Cathy\Sample Pkg.PWC

Number of Entries: 2
Compressed Size: 52.0 KB (53,248 bytes)
Uncompressed Size: 1.07 MB (1,126,915 bytes)
Compression (%): 95.27
Unique ID: 4380A854-39D3-45FC-9130-25416D7D91EA

The **Information** tab displays the following information:

- The full path and file name of the Prism file
- The number of items contained in the file, both with and without contents
- The compressed size of the file
- The uncompressed size of the file (that is, the total size of all items if they were uncompressed)
- The ratio of compression (the compressed size divided by the uncompressed size)

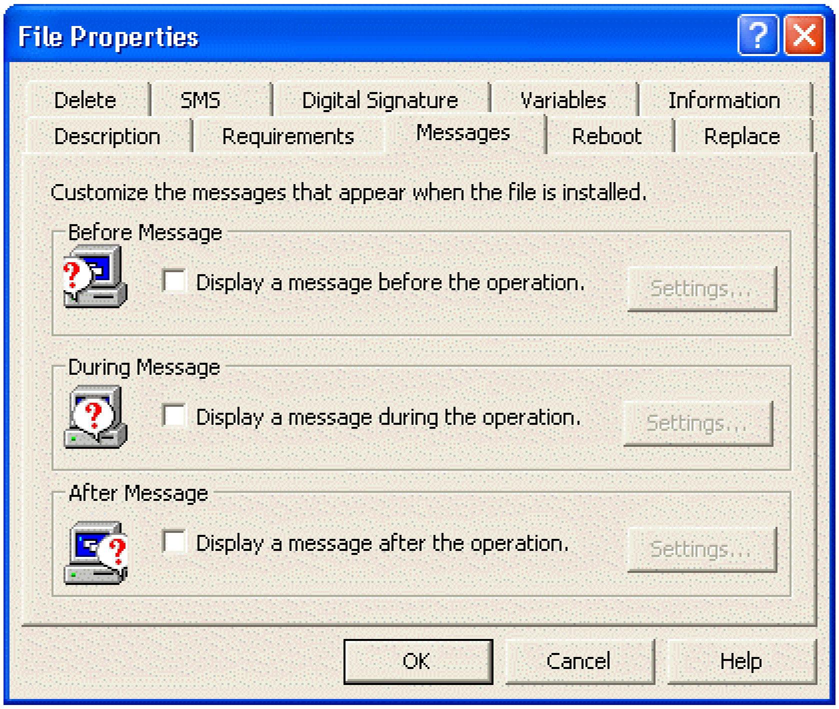
- The unique ID assigned by Prism Deploy to this Prism file. This ID is used to track uninstall information in the Uninstall file.

  The **New ID** button lets you assign a new ID to the file. Use this button along with **File | Save As** to create two unique versions of the same Prism file. (See Saving or Duplicating a Prism File from the Editor Window.)

> **Warning**: When you give an existing file a new ID, Prism Deploy can no longer determine whether this file has already been installed on a certain computer or where the installation information may be kept.

With the exception of **New ID**, you cannot edit any of this information. If changes are made to the Prism file, the information displayed here is updated automatically.

Messages Tab



Customize the messages that appear when the file is installed.

□ Display a message before the operation.

□ Display a message during the operation.

□ Display a message after the operation.

In the File Properties | Messages tab, you can choose whether to display a message before, during, or after installing the Prism file. Use the messages to offer users a chance to cancel the operation before it begins, alert them that the operation is still proceeding or describe follow up actions. You can display a message at one stage of the operation, two stages, or at all stages.

You can customize each message and display a bitmap graphic in the message box. In the **Messages** tab, select the check box for Before, During, or After Message. A dialog box is displayed that lets you enter a message and select a bitmap.

Displaying a Message Before the Operation

Click the **Display a message before the operation** check box in the File Properties | Messages tab to display a message and/or a graphic before the Prism file is installed to a target computer. Use the Before Message dialog box to customize the message.

If you do not set a Before Message through the file properties, a message is still displayed if:

- You specify a prompt with the command line (See "Scripting Prism Deploy Tasks" in the *Deployment Guide* or the Prism Console online Help.)
- You install the file from the Prism Deploy Editor window

- The user installs the file by double clicking the file in Windows Explorer
- The user launches the file from an e mail message
- The user launches the file from a Web page

Enter a message to display before the file is installed

Select the **Display Text** check box in the Before Message dialog box and type the message in the text box. If you select the check box but do not enter a message, the default message is displayed. The default message depends on these options:

- **Ask the user if the file should be installed**. The default message asks the users to confirm that they want to install the Prism file on the computer. The user can choose between **Yes** and **No** buttons.
- **Warn the user that the file will be installed.** The default message warns the recipient that the Prism file is about to be installed. The user clicks **OK** to acknowledge the message and install the Prism file. With this option, the user does not have the choice of canceling or not installing the file.

Choose a graphic to display before the file is installed

☑**Display Bitmap**: Select the Display Bitmap check box in the Before Message dialog box and enter a path and filename for the bitmap. If you select the Display Bitmap check box but do not enter a file name, the default graphic is displayed. The default graphic shows the New Boundary Technologies logo.

Select the Ask or Warn option

◉Select the **Ask the user if the file should be installed** option to display **Yes** and **No** buttons on dialog box. The user can decide whether or not to install the file.

◉Select the **Warn the user that the file will be installed** option to display an **OK** button in the message box. The user must click **OK** to install the file.

Test your message

After typing your message text and selecting a graphic, click **Test**. The message box is displayed as it will appear to the user. Click a button in the test window to close it.

Displaying a Message During the Operation

Click the **Display a message during the operation** check box in the File Properties | Messages tab to display a message and/or a graphic while the Prism file is being installed on a target computer. Use the **During Message** dialog box to customize the message.

Enter a message to display while the file is installed

☑**Display Text**: Select the Display Text check box in the During Message dialog box and type your custom message in the text box. If you select the Display Text check box but do not enter a message, the default message is displayed. The default message gives the name of the Prism file and tells the user that it is being installed.

Choose a graphic to display while the file is installed

☑**Message Bitmap**: Select the Message Bitmap check box in the During Message dialog box and enter a path and filename for the bitmap. If you select the Message Bitmap check box but do not enter a file name, the default graphic is displayed. The default graphic shows the New Boundary Technologies logo.

Display animation while installing the file

☑**Display animation while installing the file**: With this option selected, an animated image is displayed that indicates the operation is continuing. It does not reflect the progress of the operation.

Allow users to cancel

☑**Allow the user to cancel while installing the file**: With this check box selected, a **Cancel** button is displayed in the message window. This button lets the user cancel the operation while it is in progress. If the user cancels the operation, Prism Deploy returns the computer to its previous state.

Test your message

After typing your message text and selecting a graphic, click **Test**. The message box is displayed as it will appear to the user. If you did not add a **Cancel** button, the test window disappears after a short time. Otherwise, click a button in the test window to close it.

Displaying a Message After the Operation

Click the **Display a message after the operation** check box in the File Properties | Messages tab to display a message and/or a graphic after the Prism file is installed on a target computer. Use the After Message dialog box to customize the message.

Enter a message to display after the file is installed

☑**Display Text**: Select the Display Text check box in the After Message dialog box and type your custom message in the text box. If you select the Display Text check box but do not enter a message, the default message is displayed. The default message tells the user that the Prism file was successfully installed.

Choose a graphic to display after the file is installed

☑**Message Bitmap**: Select the Message Bitmap check box in the After Message dialog box and enter a path and filename for the bitmap. If you select the Message Bitmap check box but do not enter a file name, the default graphic is displayed. The default graphic shows the New Boundary Technologies logo.

Test your message

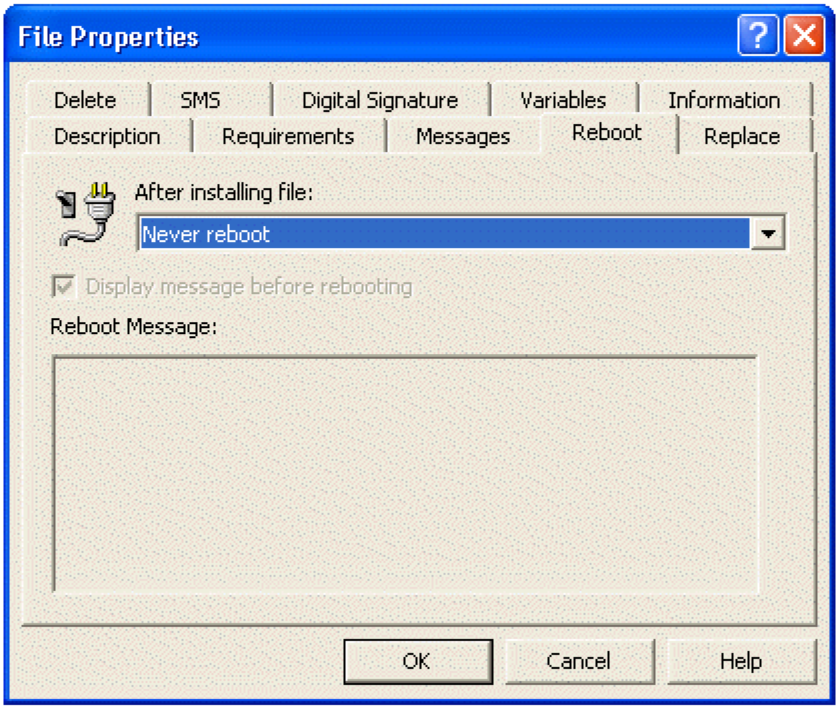
After typing your message text and selecting a graphic, click **Test**. The message box is displayed, as it will appear to the user. Click **OK** in the test window to close it.

Editing the Before, During, and After Messages

Once a message is set to display before, during, or after the file operation, you can edit or modify it at any time.

In the File Properties | Messages tab, click the **Settings** button for the Before Message, During Message, or After Message. The dialog box you used to set up the message originally is displayed. Make your changes. Click **OK**.

Reboot Tab



After installing file: Never reboot

☑ Display message before rebooting

Reboot Message:

The **Reboot** tab lets you specify whether Prism Deploy should reboot the computer after installing the file. Some applications require a reboot after installation for the new settings to take effect. In this case, you should have Prism Deploy reboot the computer after it installs the Prism file.

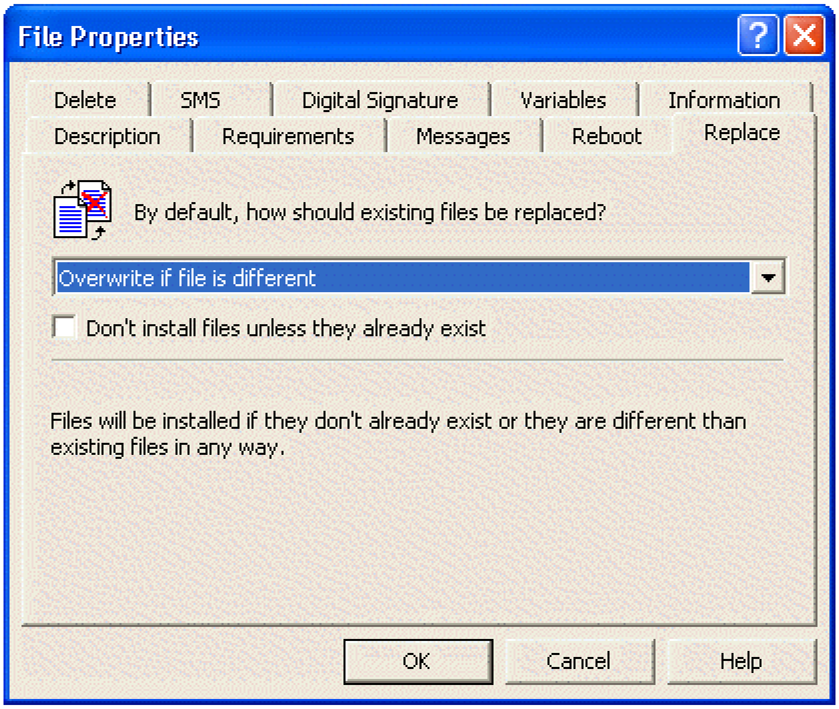
From the **After installing file:** drop down list, select one of these options for rebooting the target computer:

- **Never reboot**: Do not reboot after installing the file. This option is the default.
- **Attempt to reboot**: If the person receiving the file is working when the program is distributed, Prism Deploy gives each open application a chance to save its work. Each application has the ability to stop the reboot if it cannot save its work. The message displayed to the user includes an **OK** and **Cancel** button, giving the user the option of rebooting at a later time.
- **Always reboot**: Prism Deploy gives each application a chance to save its work. Then Prism Deploy reboots the computer no matter what response it gets from the open applications. The person receiving the file may lose unsaved documents. The message displays only an **OK** button.

Note: If you selected **Attempt to reboot** or **Always reboot**, you can include a message that will display before Prism Deploy reboots the computer. Select the **Display message before rebooting** check box. Type the message in the **Reboot Message:** text box.

If you select the Display message before rebooting check box and **do not** specify text with this option, a default message is displayed.

Replace Tab



By default, how should existing files be replaced?

Overwrite if file is different

□ Don't install files unless they already exist

Files will be installed if they don't already exist or they are different than existing files in any way.

The **Replace** tab lets you identify the circumstances under which existing files are replaced by the Prism file installation.

Select an option from the drop down list:

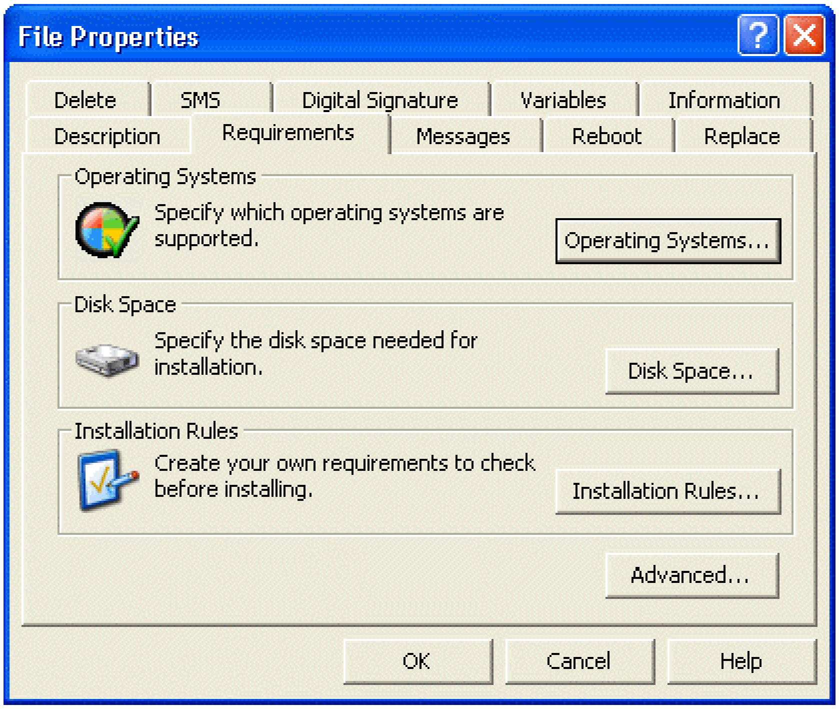
- **Overwrite if file is different**: Files are installed if they do not already exist or if they differ from existing files in any way. This option is the default.
- **Overwrite if date/time is later**: Files are installed if they do not already exist or if their date/time stamp is later than existing files.
- **Overwrite if date/time is earlier**: Files are installed if they do not already exist or if their date/time stamp is earlier than existing files.
- **Never overwrite existing file**: Files are installed only if they do not already exist.
- **Overwrite if version is different**: Files are installed if they do not already exist or their version numbers differ from existing files.
- **Overwrite if version is later**: Files are installed if they do not already exist or if they have a later version number than existing files.
- **Overwrite if version is earlier**: Files are installed if they do not already exist or if they have an earlier version number than existing files.
- **Overwrite if version is identical**: Files are installed if they do not already exist or if their version numbers are identical to existing files.
- **Overwrite if file is identical**: Files are installed if they do not already exist or if they are identical to existing files.
- **Always overwrite existing file**: Files are always installed.

Note: If version information is not available for a file and the replacement option that you select compares the versions, the corresponding date/time option is used instead. For example, if Prism Deploy is installing a .LOG file (which does not contain a version number) with the **Overwrite if version is later** option selected, the **Overwrite if date/time is later** option is used.

- ☑**Don't install files unless they already exist**: Select this check box to install the Prism file on the target computer only if the files already exist. This option prevents the addition of new files to the target computer.

  For example, click this check box and select **Overwrite if date/time is earlier** from the drop down list to replace accidentally overwritten files without adding anything new.

File Requirements Tab



Operating Systems: Specify which operating systems are supported.

Disk Space: Specify the disk space needed for installation.

Installation Rules: Create your own requirements to check before installing.

The **Requirements** tab lets you define the operating system required for installing the file, set the amount of free disk space that must be available, or create your own requirements to check before installing the Prism file.

Operating System: Specify the Supported Operating Systems

Use this option to prevent installation on unsupported operating systems. Click the **Operating System** button to specify the operating systems or roles that are compatible with the Prism file.

When you click the **Operating System** button, the Operating System Requirements dialog box opens.

Disk Space: Check the Amount of Free Disk Space

Use the Disk Space option to check the amount of available disk space before installing the Prism file. You can check the available disk space and set a minimum requirement for any drive, special folder, or a network volume. With the default options, Prism Deploy does not check disk space before installing a file.

When you click the **Disk Space** button, the Disk Space Requirements dialog box opens.

Installation Rules: Create Your Own Requirements for Installation

Installation rules let you customize the installation requirements for the target computer. If the target computer does not match the requirements defined with a rule, the Prism file is not installed on that computer. These rules let you fine tune the installation requirements with specific information about the operating system, the computer, or the user.

Generally, these installation requirements are used if you are distributing a Prism file with a command or other alternate method, rather than through the Console. However, installation requirements set through the File Properties can provide an extra level of protection, on top of the requirements you set with a predefined or user-defined group in the Console. For example, if installing the changes on the wrong brand of CPU would be harmful, create a rule to specify the CPU maker on this tab and set up a predefined group based on CPU maker in the Console.

When you click the **Installation Rules** button, the Installation Rules dialog box opens.

Advanced: Customizing the Prism File for Multiple Users

For target computers where multiple users log in, you can further refine your Prism file to allow for unattended installation or customize the file for an individual user. The primary settings for these scenarios are in this File Properties | Requirements tab and the Variable Properties | Requirements tab.

The settings for all users versus the current user can also be adjusted with the command line options, giving a wide array of combinations. See "Scripting Prism Deploy Tasks" in the *Deployment Guide* or the Prism Deploy Console online Help for information about the command line options and syntax.

When you click the **Advanced** button, the Advanced Requirements dialog box opens.

Note: The Advanced Requirements apply only to Prism files that you distribute from the command line or with an alternative method of installation. If you install the file from the Prism Deploy Console, the file is always installed for all users.

Operating System Requirements dialog

The Prism file is installed only on target computers that match the selection(s) on this dialog box. Select a check box to install the Prism file under a specific operating system. Or, clear a box to ensure that it is never installed on a target computer with that operating system.

For example, to distribute a patch that applies only to Windows Vista, select the Windows Vista check box and clear all of the other check boxes. When you distribute the Prism file, it is only installed on target computers running Windows Vista.

Note: Prism Deploy’s default setting is to select all operating systems. If you have an operating system specific Prism file (such as a driver installation), it is important to define the appropriate operating system.

The Operating System Requirements dialog box opens when you click the **Operating Systems** button on the File Properties | Requirements tab.

Select from these options:

- **Windows Family**: Identifies either the family or the specific version of Windows running on the target computer.

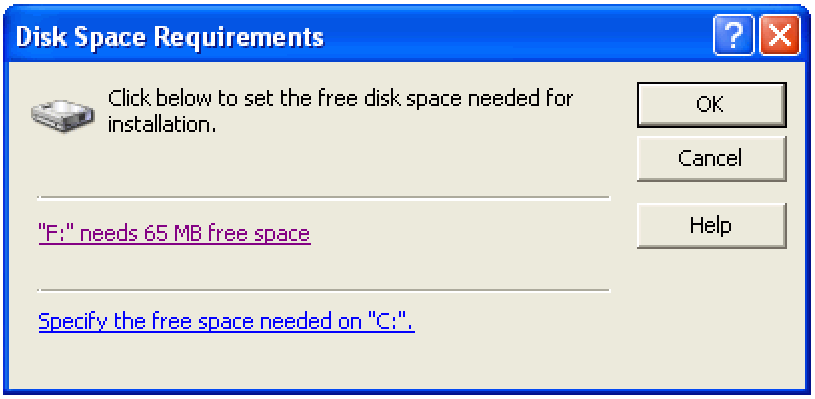
Note: The older OS's are shown for backwards compatibility with older packages..

- **Computer Role**: Identifies the target computer as a server, workstation, or domain controller.

  The check boxes under **Computer Roles** behave like the other boxes; you clear them to prevent unwanted installation. For example, if you wish to prevent installation on servers, select only the Workstation check box.

Note: Use the **Future Versions** check box to ensure that the Prism file can be installed on an operating system released after those listed. For example, if you want the Prism file to install on the next version of Windows, select the **Future Versions** check box within the **Windows NT Family** list.

Disk Space Requirements dialog



Click below to set the free disk space needed for installation.

"F:" needs 65 MB free space

Specify the free space needed on "C:".

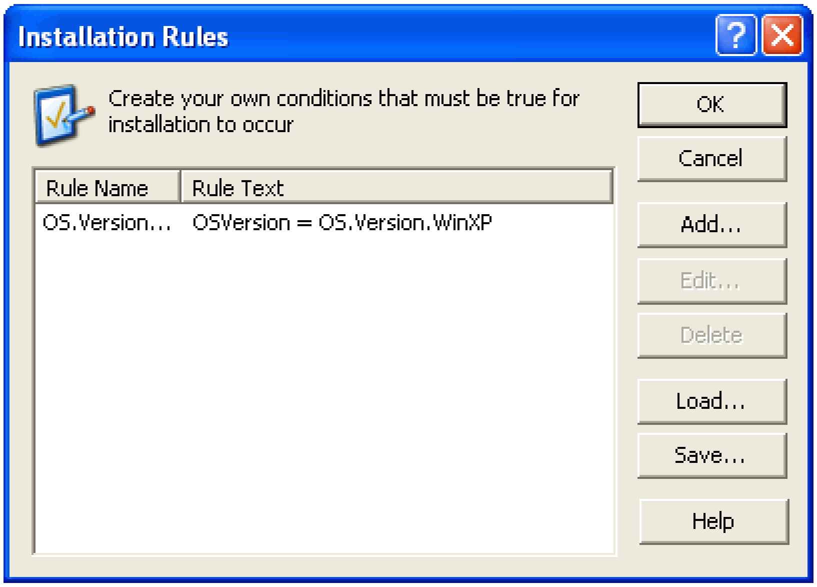
On this dialog box, click the link for the drive or network volume to be checked before installing the Prism file. You can check the available disk space and set a minimum requirement for any drive, special folder, or network volume.

The Disk Space Requirements dialog box opens when you click the **Disk Space** button on the File Requirements Tab.

To set a disk space requirement, click the link for that folder, drive, or network volume. The Item Properties | Disk Space Requirements tab for the item opens.

This dialog box includes a link for the C: drive and any other items that had disk space requirements set directly through the Item Properties | Disk Space Requirements tab.

Installation Rules dialog



Create your own conditions that must be true for installation to occur

Rule Name	Rule Text
OS.Version...	OSVersion = OS.Version.WinXP

The Installation Rules dialog box lists all of the rules used to define installation requirements for the Prism file. If you set up more than one rule, all of the requirements are evaluated before installing the Prism file.

This dialog box opens when you click the **Installation Requirements** button on the File Properties | Requirements tab.

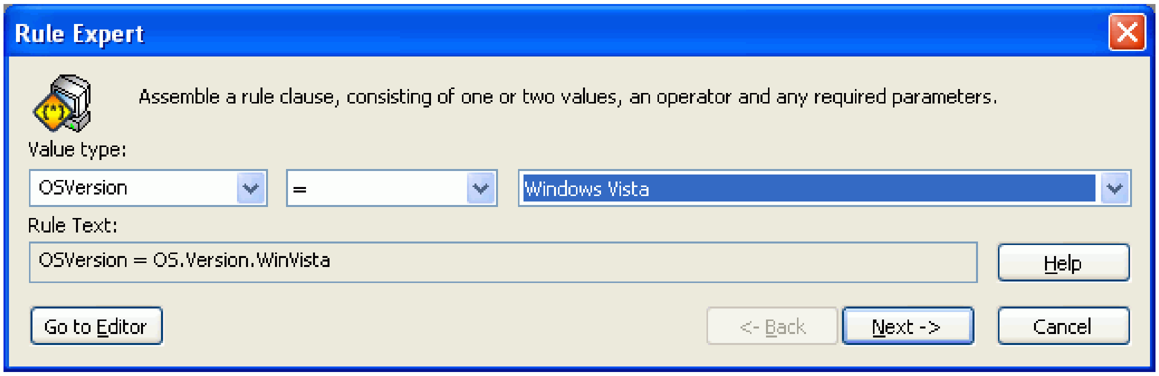
Options on this dialog box include:

- **Add**: Set up a new rule for setting installation requirements. All of the rules listed here are evaluated before installing the Prism file. The Rule Expert leads you through the process of setting up a rule and selecting variables.
- **Edit**: Edit or update the rule that is currently selected in the list of rules.
- **Delete**: Remove the rule that is currently selected in the list of rules.
- **Load**: Add a rule to the current Prism file by loading or importing a Prism Ruleset file (.RULESET) that contains the rule.

Caution: When you load a saved Ruleset file, it replaces the Rule of the same name.

- **Save**: Save the rule as a Prism Ruleset file (.RULESET). Saving and importing rules can offer a convenient way to transfer rules to another Prism file or between Prism files and configuration groups in the Prism Console.

Rule Expert: Assemble a Rule Clause



Assemble a rule clause, consisting of one or two values, an operator and any required parameters.

Value type: OSVersion = Windows Vista

Rule Text: OSVersion = OS.Version.WinVista

Use the fields on this dialog box to define a rule for installation requirements set up through the File Properties. Installation rules let you customize the installation requirements for the target computer. If the target computer does not match the requirements defined with a rule, the Prism file is not installed on that computer. For example, if installing the changes on the wrong brand of CPU would be harmful, create a rule to specify the CPU

This dialog box opens when you click the **Add** button on the Installation Rules dialog box.

Options on this dialog box include:

- **Value Types**: Sets the basis for the rule. This type of value on target computers is compared to the rule before installing the Prism file.
- **Value type identifier**: If the **Value Type** requires an identifier (those value types enclosed in angle-brackets, for example <File>), identify the item here. For example, if you choose <File> for the **Value Type**, enter the path of the file—"C:\Program Files\Acme Software\Acme.exe"—in the **Value** text box. The Prism file compares the version of this file on the target computer with the value in the last field on this dialog box.

  All strings must be enclosed in quotes (" "), which the Rule Expert adds for you automatically. If you type a rule directly in the Rule Editor, remember to add the quotes. (See Notes on Rule Syntax.)
- **Comparison Operator**: The comparison operator determines how the value on the target computer is compared to the rule.

For a complete list and descriptions of the Comparison Operators available in the drop-down list, see Comparison Operators.

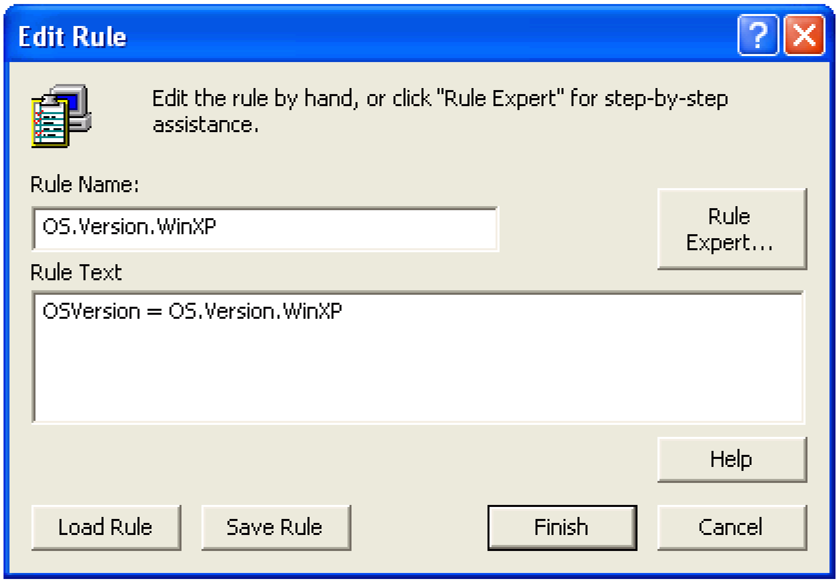
- **Value**: Type the variable value or select an item from the drop-down list. The items available in the drop-down list depend on the selection you made in the Value Types field. For example, if you choose first letter in the Value Types field, the Value field lists the drive letters A, B, C, D, E, and so on.

  If you type a value directly in this field or directly in the Rule Editor, it must match the values created by New Boundary Technologies for the **Value Type**.

  For a complete list of the values, see:

  - Predefined Variables
  - User-defined Variables

Edit Rule dialog



Edit Rule

Edit the rule by hand, or click "Rule Expert" for step-by-step assistance.

Rule Name: OS.Version.WinXP

Rule Text: OSVersion = OS.Version.WinXP

This dialog box lists the rule name and the text of the rule itself. Review the information displayed here and edit these entries directly if you want to make changes.

This dialog box opens when you:

- Click the Next button on the Rule Expert: Rule Name dialog box.
- Click the **Go to Editor** button on the Rule Expert: Assemble a Rule Clause dialog box.

Options on this dialog box include:

- **Rule Name**: Descriptive name of the rule.
- **Rule Text**: The text of the rule, including the variable and values that define the installation requirements. You can edit the rule text or type it directly in this text box. (See Rule Variables and Operators for Installation Requirements.)
- **Rule Expert**: Add additional criteria by joining another rule to the existing rule. The Rule Expert leads you through the process of identifying a conjunction (AND or NOT) and creating a rule.
- **Save Rule**: Save the rule as a Prism ruleset file (.RULESET) file. This file can be loaded into another Prism file.

- **Load Rule**: Click this button to load a Prism ruleset file (.RULESET) into the Prism file.

Caution: If the ruleset file has the same name as an existing rule, the existing rule is overwritten

Choose a Joining Method for the Clause dialog

Choose a conjunction for joining two or more rules that define the installation requirements. The following example shows two rules joined with the conjunction OR:

(CPU = "PENTIUM 5" AND OSVERSION = W2K) OR (CPU = "PENTIUM 4" AND OSVERSION = W95)

This dialog box opens when you click the **Add Rule Expert** button on the Edit Rule.

Choose from these conjunctions:

- **AND**: The Prism file is only installed on the computers that meet both or all sets of criteria.
- **OR**: The Prism file is installed on computers that meet either or any of the criteria.

Advanced Requirements dialog

For target computers where multiple users log on, you can further refine your Prism file to allow for unattended installation or to customize the file for an individual user.

By using the property settings and command line options you can:

- Update all users on the computer through a single installation of the Prism file and avoid the problem of reinstalling the same settings repeatedly under each user account.
- Uninstall the Prism file when any user is logged in. To uninstall a file, the user performing the uninstall does not need to be the same user who installed the file originally.
- Uninstall a file to remove all changes for all of the users, at one time. Or use the command line option.

Note: These Advanced Requirements apply only to Prism files that you distribute from the command line or with an alternative method of installation. If you install the file from the Prism Console, the file is always installed for all users.

The Advance Requirements dialog box opens when you click the **Advanced** button on the File Requirements Tab.

Options on this dialog box include:

- **Unattended\All Users**: Choose the **Unattended/All Users** option button when you plan to perform an unattended installation or for changes you want to apply to all users on the target computer.

  With this option, Prism Deploy installs the changes in the Prism file to:

  - all current users on the target computer when they log on
  - any new users when they are created

- **Current User Only**: Choosing the **Current User Only** option button results in the Prism file being installed only to the person logged on at the time the Prism file is installed.

  Use this option when only one person on the target computer should receive the changes or additions in the Prism file.

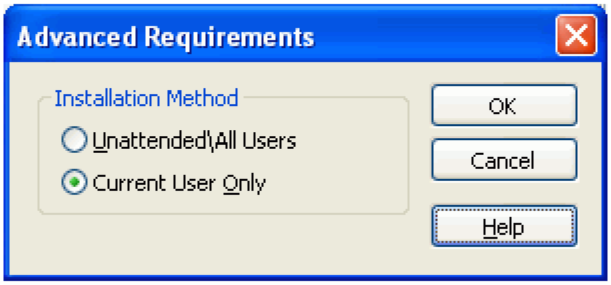
Note: The current user is either the person logged on to the computer at the time or, if no one is logged on, the default user.

Customizing Changes for One User

If you have both an update for a specific user on a target machine and changes for all users of that machine, you must prepare two separate Prism files. In the first Prism file, set up the changes to be installed for All Users of the computer. This Prism file is installed first. Then, set up a second Prism file with the changes destined for a specific user.

For example, if you have a patch for the company word processing application, create a Prism file that will go to all users on the target computer, as described under "Preparing changes to install for all users on the target computer". If you also have a new document template for the department budget and only one of the users needs access to it, create a second Prism file and set its properties as follows:

1. Click 📋 or select **File** | **Properties** to open the File Properties dialog box.
2. Click the Requirements tab.
3. Click the **Advanced** button. The Advanced Requirements dialog opens.
4. Check to be sure that the **Current User Only** is selected. (This option is the default value.)



Advanced Requirements

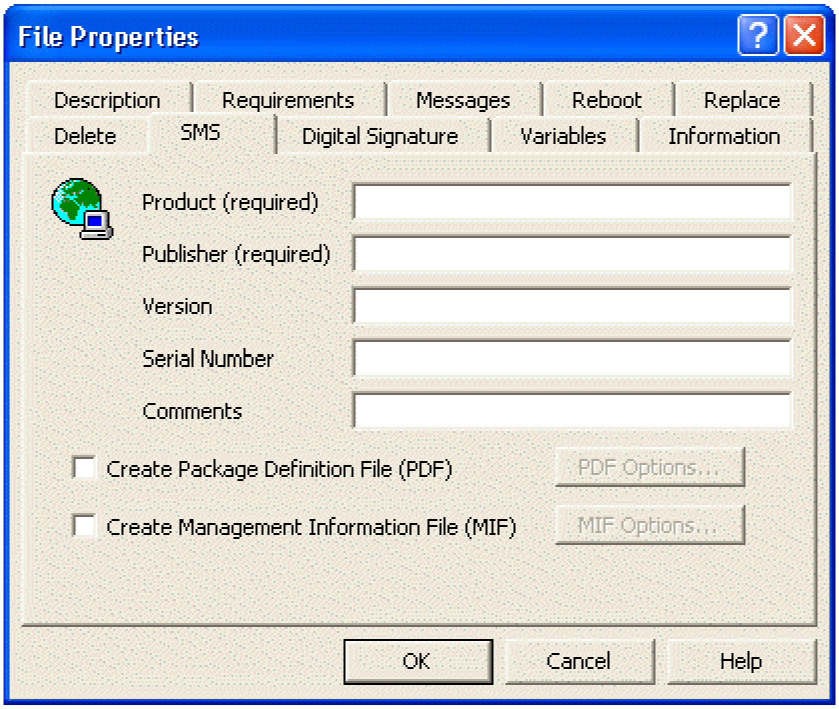
Installation Method

○ Unattended\All Users

● Current User Only

5. Provide a copy of the Prism file to the target user. Ask this person to install the Prism file while logged on.

Setting Up Files for SMS Integration



File Properties

Product (required)	
Publisher (required)	
Version	
Serial Number	
Comments	

□ Create Package Definition File (PDF)

□ Create Management Information File (MIF)

The Prism File Properties | SMS tab helps you integrate Prism files into SMS's package building process, reporting functions, and information gathering functions. On this tab, you can enable the creation of PDFs (Package Definition File) or MIFs (Management Information File) and enter information about the Prism file that is required by SMS.

The PDF records information you specify on this tab, along with information about the Prism file properties for use when building the SMS Package. When it is installed, the Prism file creates the MIF in the %WINDIR% directory on the target computer. SMS retrieves this file to gather information about the success or failure of the Prism file installation.

The Prism file created with information entered on this tab does not require SMS for installation. Prism Deploy creates the additional file that you specify here, but the operation and characteristics of the Prism file do not change.

Product Information

The fields in the upper section of the tab record information about the software product being installed by the Prism file. If the Prism file is being used to make other types of changes, rather than to install a product, enter a descriptive title or identifying information in the **Product** and **Publisher** fields.

- Product
- Publisher
- Version
- Serial Number
- Comments

Create PDF and MIF Files

Select one or both check boxes in this section to create an SMS PDF or MIF file. When you close the Prism file in the Editor, the PDF is created in the same location as the Prism file. When the Prism file is installed or uninstalled, the MIF is created in the %WINDIR% folder.

- **Create Package Definition File (PDF)**: Select this check box to create a PDF when you close the Prism file in the Prism Deploy Editor.

  A PDF is a text file used by SMS to describe the origin and content of a Prism file. It contains instructions on how to install the Prism file. The PDF is given the same name as the Prism file with either .SMS or .PDF as the file extension. If you later modify the Prism file's properties, the PDF is automatically updated when the Prism file is closed.

  PDF Options: Click the PDF Options button to specify the type of PDF file to create. Choose from these options on the Package Definition File Options dialog box.

  SMS Version 1.x format

  SMS Version 2.0 format

- **Create Management Information File (MIF)**: Select this check box to create an MIF in the %WINDIR% directory on the target computer when the Prism file is installed or uninstalled.

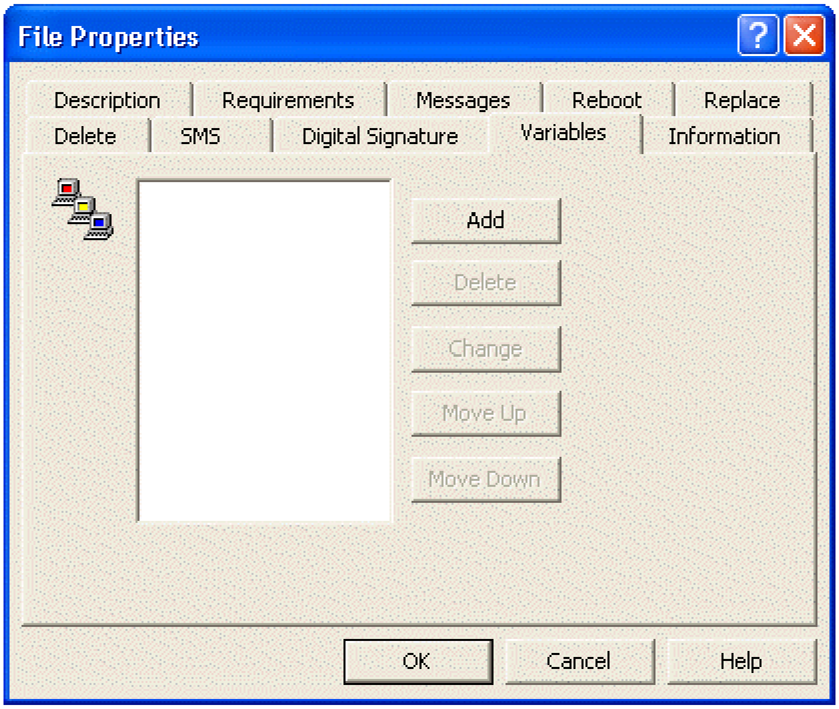
  The MIF is a text file that describes the software installed by the Prism file and information about whether the installation was successful. If the installation fails, the file also records a diagnostic error message.

  MIF Options: Click the **MIF Options** button to name the install and uninstall MIF files to be created. Choose from these options on the Management Information File Options dialog box.

Install MIF Name

Uninstall MIF Name

Variables Tab



The **Variables** tab lets you enter variables that are resolved when Prism Deploy installs this file. These variables can be used to indicate differing target paths, registry keys, and so on. The variable's source can be a file, another variable, an INI file, or a registry key, or you can prompt the user to enter the variable's value.

- **Add a variable**: Click the **Add** button to add a new variable. The Variable Properties dialog box opens.

  As you add variables, they are displayed in the list box.
- **Delete a variable**: To delete a variable, highlight it and click the **Delete** button.
- **Change a variable**: To modify a variable, highlight it and click the **Change** button. The Variable Properties dialog box opens.
- **Change the resolution order**: The variables are resolved in the order they are listed. If a variable should be resolved before others in the list, highlight it and click the **Move Up** button to move it up in the list. The variable is resolved before the variables listed below it.

  To move a variable down in the list and have it resolved later, highlight a variable and click the **Move Down** button.

Variable Properties

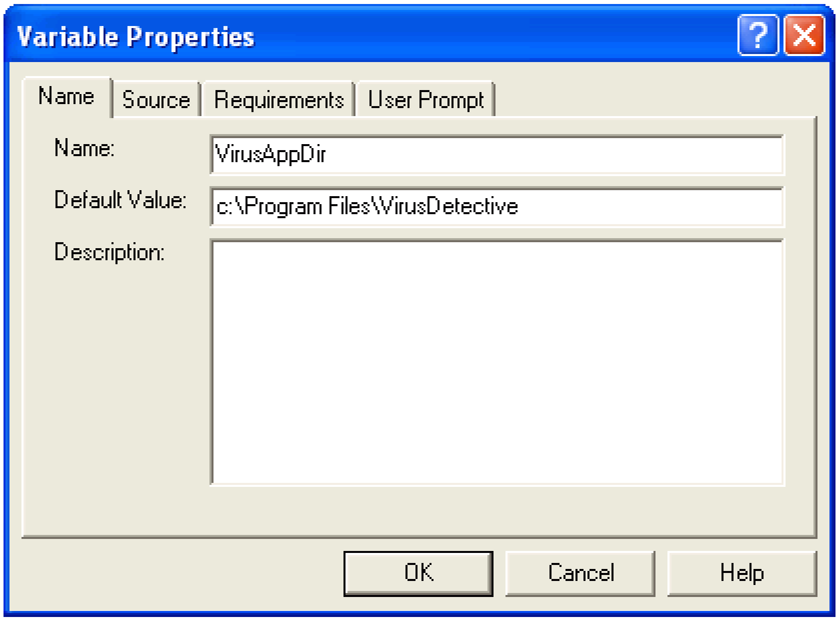
Variable Properties dialog

Use the **Variable Properties** dialog box to define the name, default value, source, and requirements (including user prompts) of a variable. To open the **Variable Properties** dialog box, click the **Add** button in the Variables tab of the File Properties dialog box.

Tabs on this dialog box include:

- Name tab
- Requirements Tab
- Source Tab
- User Prompt Tab

Name Tab



Name: VirusAppDir

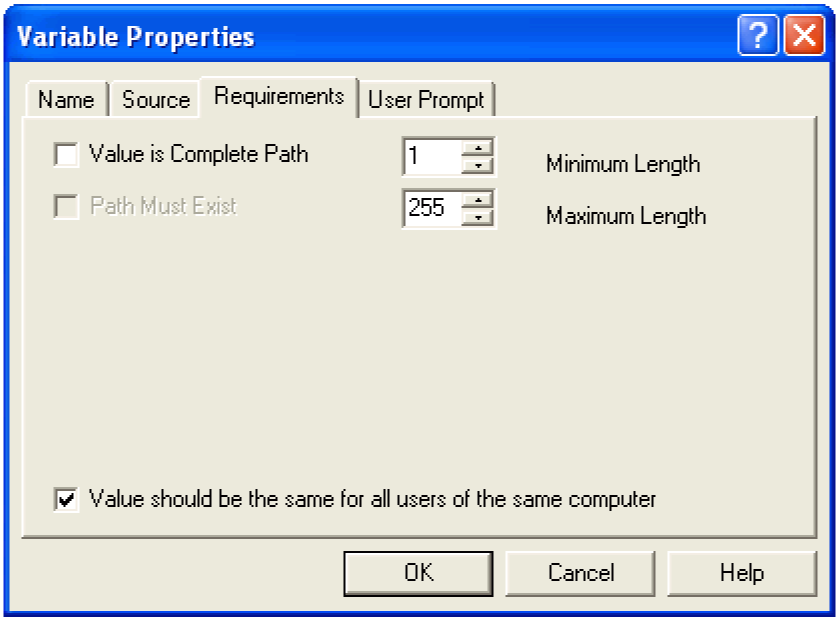
Default Value: c:\Program Files\VirusDetective

Description:

In the Variable Properties | Name tab, name the variable you are setting up and if required, specify a default value for the variable. This tab opens when you click the **Add** or **Change** buttons on the File Properties | Variables tab

- In the **Name:** text box, enter a descriptive name for the variable. Do not choose a name that is a currently defined environment variable, unless you intend the variable to retrieve a value from the environment.
- As an option, enter the default value for the variable in the **Default Value:** text box. This value is used if no other resolution is possible. (See Variable Resolution Order .)
- As an option, enter a description of the variable in the **Description:** text box. This description is only for you, it is not displayed to users who receive the Prism file. If you enter a description, it is displayed in the File Properties | Variables tab.

Variable Requirements Tab



☐ Value is Complete Path

☐ Path Must Exist

1 Minimum Length

255 Maximum Length

☑ Value should be the same for all users of the same computer

Use the **Requirements** tab to define requirements for the new variable's value.

- **Value is Complete Path**: Select if the variable's value must be a fully qualified path. For example, a fully qualified path may begin with a drive letter (D:\FOLDER) or a UNC (Uniform Naming Convention) path (\\Server\folder).
- **Path Must Exist**: Select if the specified path must exist on the target computer before the variable can be resolved. If the path cannot be resolved, the Prism file that contains the variable is not installed.
- **Minimum Length** and **Maximum Length:** Set the length of the variable's value.

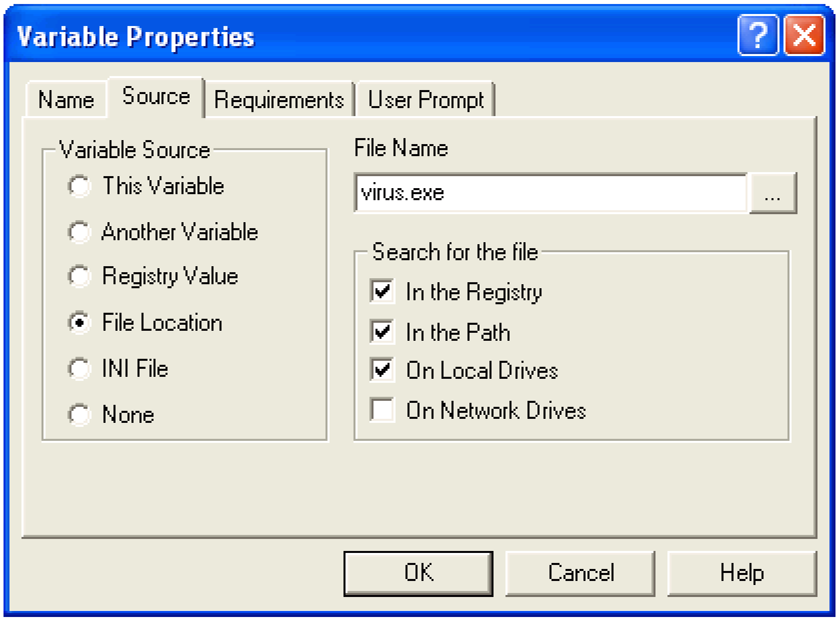
All Users

- ☑**Value should be the same for all users of the same computer**: With this option selected, if the target computer has more than one user set up and this option is set, the variable will have the same value for all users of the computer. When the file is installed, the variable value resolves for the current user. The value is used for all users of the computer. In most cases, leave this box checked.

  For example, check this box if the variable is the path for a new application to be added to a target computer and all users have the same application, stored in the same location.
- ☐**Value should be the same for all users of the same computer**: With this option turned off, the variable resolves for each unique user when they log on. The variable, such as the name of a Start menu item, can be different for each user.

  For example, turn this option off if the variable is a folder that will be placed on the desktop.

Source Tab



Variable Source

○ This Variable
○ Another Variable
○ Registry Value
● File Location
○ INI File
○ None

File Name: virus.exe

Search for the file

☑ In the Registry
☑ In the Path
☑ On Local Drives
☐ On Network Drives

Select the appropriate option button for the variable's source:

- **This Variable:** Select this option to retrieve the variable's new value from an existing environment variable of the same name or from this variable's default value.

  If this option is selected, Prism Deploy searches the target computer's environment first to find a value for the variable. If a corresponding variable is not located in the environment, Prism Deploy uses the default value, which is specified on the Variable Properties | Name tab. This option is the default.
- **Another Variable**: Select this option to retrieve the new variable's value from another variable. The source variable should be defined before the Prism file is installed. This other variable is typically one that is set at the computer or through a login script.

  Use this option if an environment variable is defined but you want the option of overriding that variable when the Prism file is installed. To have Prism Deploy prompt you for the override value, define a prompt for the variable. (See Variable Properties | User Prompt Tab.)

  Enter the name of the source variable in the **Source Variable Name** text box.
- **Registry Value**: Select this option to retrieve the variable's value from the specified registry value under the specified key. Leave this blank to use the registry key's default value. Only REG_SZ and REG_EXPAND_SZ values are supported.

  In the **Registry Key** text box, enter the full registry key path under which Prism Deploy can find the source registry value. The root key can be spelled out (for example, HKEY_CURRENT_USER) or abbreviated (for example, HKCU).

  In the **Registry Value** text box, enter the registry value from which to retrieve the variable's value. Leave the text box blank to use the registry key's default value. Only REG_SZ and REG_EXPAND_SZ value types are supported.

  Hint: If you do not know where Windows stores a registry value, let Prism Deploy help you find it. Take a Picture of the registry. Make a change that affects the value you want to locate. Find the changes with Prism Deploy. You see the affected registry keys immediately.

- **File Location**: Select this option to resolve the variable's value to the location (full path) on the target system of the file that you enter in the text box. The first file found with the same name is used.

  By default, the App Paths key in the registry is searched first. Followed by the directory on the search path, then all of the local drives. If the file is not found in App Paths or in the directory on the search path, this process can take a long time.

  For example, this option may be useful when upgrading software that is or may be installed in a different location on each user's computer, such as a virus scanning program.

  1. Set up a variable to represent the virus program's folder and set the source to the location of the SCAN.EXE program.
  2. In the **File Name** text box, type the name of the file whose location will be used to resolve this variable. Enter the file name only, not the complete path. Or click the ... button to choose the source file. Only the file name is retrieved.
  3. Under **Search for the file,** select the check boxes to indicate the area on the target computer Prism Deploy should search to locate this file. Prism Deploy searches in the order that the check boxes are listed (that is, it searches in the registry first, followed by the path, and so on). These options let you speed up searches by skipping a complete search of the hard disk. Choose from the following options:

     In the Registry
     In the Path
     On Local Drives
     On Network Drives

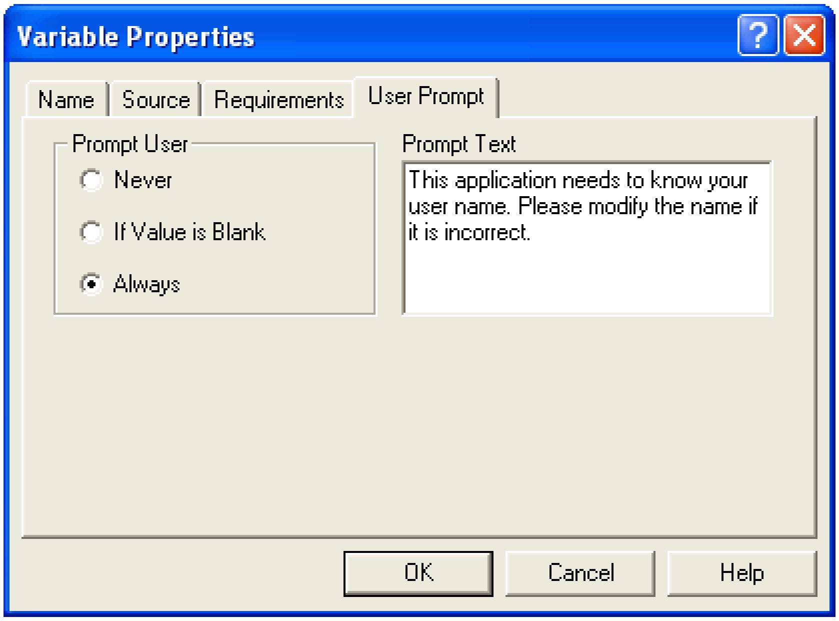
- **INI File**: Select this option to retrieve the variable's value from a key in a section of an INI file. For example, if the computer's IP address is stored in an INI file, you can configure a Prism Deploy variable to read the INI entry to resolve the variable.

  1. Type the INI file's path and name in the **INI File Name** text box, or click the ... button to choose the source file. If you omit the path, the INI file is assumed to be in the Windows directory.
  2. In the **INI File Section** text box, enter the section of the INI file from which to retrieve this variable's value. You do not need to include the brackets around the section name.
  3. In the **INI File Key** text box, enter the key from which to retrieve the variable's value.

- **None**: The variable's value is retrieved only from its default value, which is specified in the Variable Properties | Name tab.

Use this option to specify a file name, path, or other variable that is not dependent on the existing environment. You can also use this option to prompt the user for a value.

User Prompt Tab



Prompt User

○ Never

○ If Value is Blank

● Always

Prompt Text

This application needs to know your user name. Please modify the name if it is incorrect.

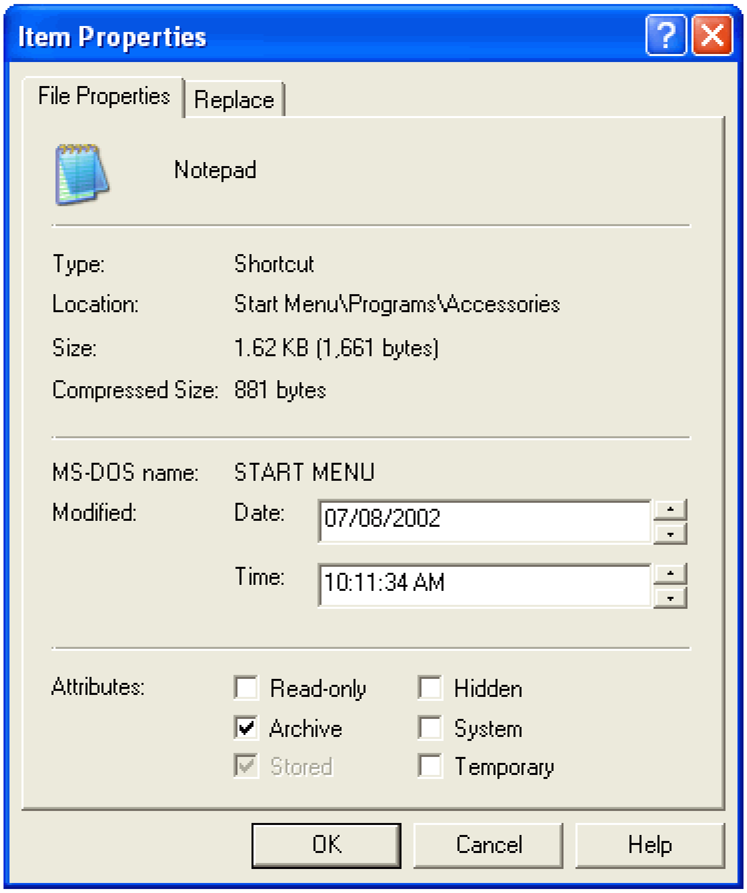
To prompt the user for the variable’s value when the file is installed, select an option under **Prompt User**:

- **Never**: The user is never prompted for the value.
- **If Value is Blank**: The user is prompted for the variable’s value only if it fails to retrieve the value from another source.
- **Always**: The user is always prompted for the variable’s value.

If you selected **Always** or **If Value is Blank**, in the **Prompt Text** box, enter a text message to display to the user, along with the prompt for the variable.

Item Properties

Item Properties dialog



Notepad

Type: Shortcut

Location: Start Menu\Programs\Accessories

Size: 1.62 KB (1,661 bytes)

Compressed Size: 881 bytes

MS-DOS name: START MENU

Modified: Date: 07/08/2002

Time: 10:11:34 AM

Attributes: □ Read-only □ Hidden

☑ Archive □ System

☑ Stored □ Temporary

On the **Item Properties** dialog box you can view properties for each item within a Prism file and define when to delete or replace the item when Prism Deploy installs the file.

To open the **Item Properties** dialog box, right click on a file, folder, registry key, or value in the Editor window and select **Properties** from the drop down menu.

Note: You can view file properties for folders, registry keys, and other high level items. However, you can only modify the properties and set replace options for the low level items such as documents, files, shortcuts, and registry values.

The Item Properties | File Properties tab displays the item's file name, type, location, size, compressed size, and MS DOS name.

- You can change the **Modified Date** and **Time**. Changing these values may be useful if you selected the **Overwrite if date/time is earlier** or **later** option in the Replace tab of the File Properties dialog box.
- You can also modify the file's **Attributes** by selecting the necessary check box(es):

Read only: Makes the file read only. It can be viewed but not modified.

Archive: Indicates that this file should be archived. Some programs use this option to control which files are backed up.

Stored: This box is checked when the file's contents are stored in the Prism file.

Hidden: Indicates this file is a hidden file.

System: Indicates this file is a system file.

Temporary: Indicates this file is a temporary file.

- If the item you selected was listed under **Deleted Items** in the Editor window, you can specify the conditions under which the item is deleted from the target computer. Set these options on the Delete tab.

  The delete options vary depending on whether you select a file, folder, registry key, or registry value.
- If the item you selected was not a deleted item, click on the Replace tab to select options for when Prism Deploy should replace an existing item of the same name.

  The replace options vary depending on whether you select a file or folder or a registry value.

Note: If you selected a registry value in the Editor window, see Defining Registry Value Properties.

Setting Disk Space Requirements

The Item Properties for a drive, special folder, or network volume let you specify how much free disk space the volume must have before installation will occur. Prism Deploy does not check disk space before installing a file, unless you configure it to do so. However, if it runs out of disk space before completing the installation, the Prism file uninstalls any changes it made.

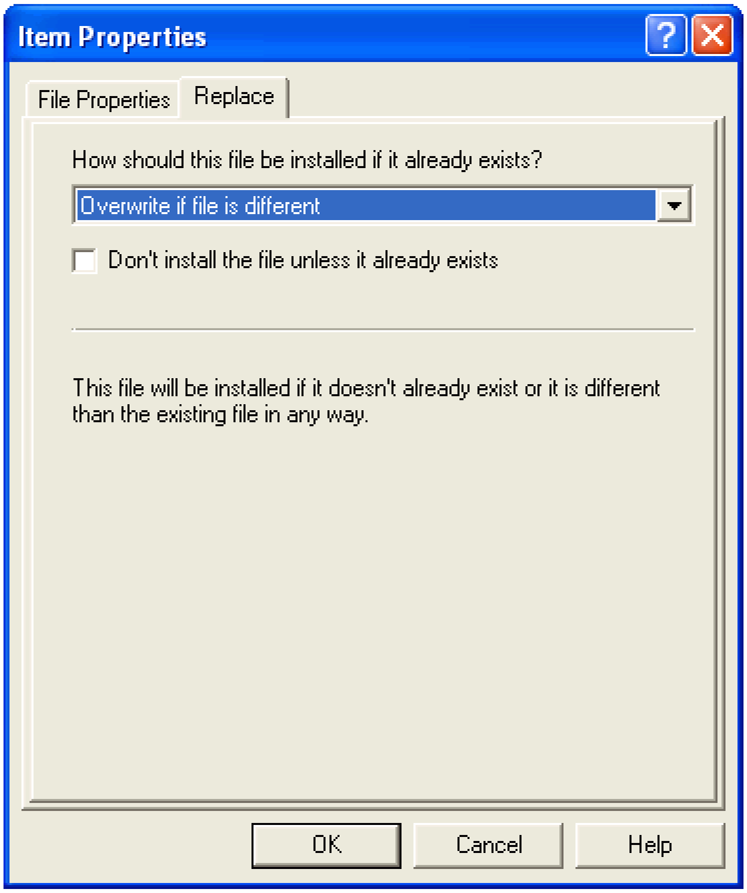
The disk space requirements can be updated through the Item Properties | Disk Space tab or through the File Properties | Requirements tab.

This tab is available on the **Item Properties** dialog box when you right-click on a drive, special folder, variable folder, or volume in the Editor window.

Options on this tab include:

- Check disk space before installing
- Minimum free disk space

Defining Replace Options



How should this file be installed if it already exists?

Overwrite if file is different

□ Don't install the file unless it already exists

This file will be installed if it doesn't already exist or it is different than the existing file in any way.

The options on the **Replace** tab of the Item Properties dialog box tell Prism Deploy how to install an item if it already exists on the user's system. The replace option set here supercedes any of the defaults set in the File Properties | Replace tab.

- Select an option from the drop down list:
  - **Use default settings**: Prism Deploy uses the default setting defined in the File Properties | Replace tab. This option is the default.
  - **Overwrite if file is different**: The file is installed if it does not already exist or if it differs from the existing file in any way.
  - **Overwrite if date/time is later**: The file is installed if it does not already exist or if its date/time stamp is later than the existing file.
  - **Overwrite if date/time is earlier**: The file is installed if it does not already exist or if its date/time stamp is earlier than the existing file.
  - **Never overwrite existing file**: The file is installed only if it does not already exist.
  - **Overwrite if version is different**: The file is installed if it does not already exist or if its version number differs from the existing file.
  - **Overwrite if version is later**: The file is installed if it does not already exist or if it has a later version number than the existing file.
  - **Overwrite if version is earlier**: The file is installed if it does not already exist or if it has an earlier version number than the existing file.
  - **Overwrite if version is identical**: The file is installed if it does not already exist or if its version number is identical to the existing file.
  - **Overwrite if file is identical**: The file is installed if it does not already exist or if it is identical to the existing file.

— **Always overwrite existing file**: The file is always installed.

- ☑**Don’t install files unless they already exist**. Select this check box to prevent a new file from being added.

Defining Delete Options for Folders

The options in the **Delete** tab of the Item Properties dialog box tell Prism Deploy when to delete a folder with the same name from a user’s computer. The delete option you select here supersedes any defaults defined in the File Properties | Delete tab. Select an option from the drop down list:

- **Use default settings**: Prism Deploy uses the default setting defined in the File Properties | Delete tab. This option is the default.
- **Delete only if empty**: Delete the folder only if it is empty.
- **Always delete**: Always delete the folder, even if it contains files or subfolders.
- **Never delete**: Never delete the folder.

Defining Delete Options for Registry Keys

The options in the **Delete** tab of the Item Properties dialog box tell Prism Deploy when to delete a registry key with the same name from a user’s computer. The delete option you select here supersedes any defaults defined in the File Properties | Delete tab. Select an option from the drop down list:

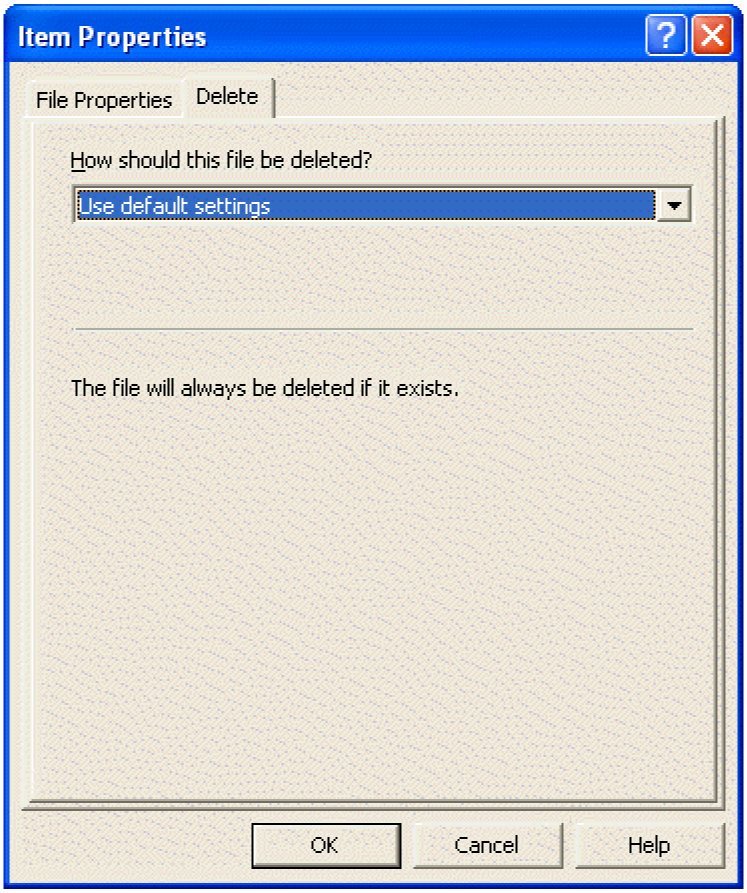
- **Use default settings**: Prism Deploy uses the default setting defined in the File Properties | Delete tab. This option is the default.
- **Delete only if empty**: Delete the registry key only if it is empty (it does not contain any other registry keys or values).
- **Always delete**: Always delete registry keys, even if it contains other registry keys or values.
- **Never delete**: Never delete registry keys.

Defining Delete Options for Registry Values

The options in the **Delete** tab of the Item Properties dialog box tell Prism Deploy when to delete a registry value with the same name from a user’s computer. The delete option you select here supersedes any defaults defined in the File Properties | Delete tab. Select an option from the drop down list:

- **Use default settings**: Prism Deploy uses the default setting defined in the File Properties | Delete tab. This option is the default.
- **Delete if value is identical**: Delete a registry value only if it is identical to an existing value.
- **Delete if value is different**: Delete a registry value only if it is different from an existing value in any way.
- **Always delete**: Always delete registry values.
- **Never delete**: Never delete registry values.
- **Subtract from existing value**: Delete the parts of a registry value that are contained in the existing value. This option should be used only when you want to delete an item from a registry value representing a list of items. This option is also only valid on registry values that are strings (that is REG_SZ, REG_MULTI_SZ, and REG_EXPAND_SZ).

Defining Delete Options for Files in a Prism File



How should this file be deleted?

Use default settings

The file will always be deleted if it exists.

The options in the **Delete** tab of the Item Properties dialog box tell Prism Deploy when to delete a file with the same name from a user's computer. The delete option you select here supercedes any defaults defined in the File Properties | Delete tab.

Select an option from the drop down list:

- **Use default settings**: Prism Deploy uses the default setting defined in the File Properties | Delete tab. This option is the default.
- **Delete if file is identical**: Delete the file only if it is identical to an existing file.
- **Delete if file is different**: Delete the file only if it is different from an existing file in any way.
- **Always delete**: Always delete the file.
- **Never delete**: Never delete the file.

Defining Registry Value Properties

To view and modify registry value properties, right click on a registry value on the right side of the Editor window and select **Properties** from the drop down menu.

The Item Properties | Registry Value Properties tab displays the name of the registry value and its location in the registry.

- Select an option from the **Data Type:** drop down list to change the registry value's data type:

  REG_SZ

REG_EXPAND_SZ
REG_MULTI_SZ
REG_DWORD
REG_BINARY
REG_COUNTER

- In the **Data:** text box, modify the value's data.

Merge Changes into List-Type or String-Type Registry Values

You can merge changes into list-type registry values (REG_MULTI_SZ) and string-type registry values (REG_SZ and REG_EXPAND_SZ), rather than overwriting the existing values. By default, Prism Deploy updates the registry values that hold the PATH component using this method. Prism Deploy's RULES.INI file can be configured to handle other string registry values in this way. (See the *Deployment Guide* or Console online help for more information about the RULES.INI file.)

Permissions

Overview of Permissions for Items

Permissions are used to control the level of access a user or group has to a file, folder, or registry key in Windows. When you define permissions for an item, you specify which users or groups have access to it and you define what they can do with it (for example, read, write, execute, and so on). Prism Deploy captures changes to permissions when finding changes to a system, and the Prism Deploy Editor lets you edit these permissions and define your own permissions. When the Prism file is installed, the permissions are added to or subtracted from the permissions for the items on the target computer. Prism Deploy handles permissions in much the same way as it handles registry values. It adds permissions to and subtracts them from a system's existing permissions, just as it adds and subtracts registry values without replacing the entire registry.

Caution: Before adding permissions to an item in the Prism file, be familiar with NTFS and registry permissions and how they are implemented in Microsoft Windows. Also, plan the permissions before adding them to a Prism file. Setting up permissions without a thorough understanding of their effect may cause problems that are difficult to reverse.

Example

You are using Prism Deploy to install a new software application—AcmePro. For security, you limit access to C:\PROGRAM FILES, but for this application to work, users need permission to write to the folder C:\PROGRAM FILES\ACMEPRO\USER DATA. You can add the correct permissions to the target computers in either of these ways:

- After installing AcmePro but *before* capturing it in a Prism file, right-click on the User Data folder in Windows. Choose **Properties** from the pop-up list. In the **Security** tab, give users the necessary access to the User Data folder.

  Now capture the AcmePro installation in a Prism file—the necessary permissions are automatically captured along with the files and folders. When the Prism file is installed, the permissions for this folder are also installed.
- If you prefer, set the permissions through the Editor. Capture the AcmePro installation with the Prism Deploy Editor. With the captured installation open in the Editor, go to the Item Properties | Security tab for the User Data folder. Set permission for everyone to write to this folder. When the Prism file is installed, the permissions for this folder are also installed.

Permissions Displayed in the Editor

Before capturing or updating permissions, it is important to understand how the Editor displays permissions. Prism Deploy only displays the explicit permissions for an item; it does not display the effective (inherited *plus* explicit) permissions for an item. So, when viewing the permissions through the Editor, the permissions you see depend on where they were added originally before you captured them in a Prism file. You see only the permissions for the folder or file that are to change. The permissions dialog boxes are blank for items that inherit these permissions.

For example, if you take a Picture of a folder that has explicit permissions, the **Item Properties | Security** tab for the folder reflects the permissions. However, the **Item Properties | Security** tab is blank for a file within the folder. In contrast, the Window's Security property sheet for each item in the folder displays the permissions for the parent folder (if inheritance has been set up).

Inheritance of Permissions

Files and folders often inherit the permissions of their parent folder. This means you do not usually have to define permissions for large sets of files or folders. Often adding permissions to or subtracting permissions from a single parent folder can accomplish what you want.

You cannot use Prism Deploy to remove inherited permissions. However, you can reverse the inherited permissions by using the **Item Properties | Security** tab to deny permissions to a file or subfolder. For example, Joe can read the folder C:\DATA\FINANCIALS because he has inherited this permission from his read access to C:\. Using Prism Deploy to remove his access to C:\DATA\FINANCIALS *does not* change his read access to this folder. To remove Joe's read access to C:\DATA\FINANCIALS, you must remove his access to C:\. Of course, this solution may prevent Joe from doing *anything* on the C:\ drive.

In the example above, it is better to add a **Deny** permission for Joe to C:\DATA\FINANCIALS. Using Prism Deploy to install a **Deny** permission prevents Joe from reading this folder and any files or folders within it. But, the deny permission is only for the C:\DATA\FINANCIALS folder. Joe and the software application he runs still have access to the rest of the hard drive.

Capturing Permissions

Permissions are captured along with other file information when you take a Prism Deploy Picture. However, the permissions are only captured and displayed in the Picture for the parent folder where the permissions were originally assigned. If you are taking a custom Picture and the parent item is not part of the Picture, the Prism file does not contain any permissions.

When you use the Editor to find changes, Prism Deploy **only** captures permissions if there was a change in the permissions between the before and after environments, Prism Deploy captures the permissions explicitly assigned to newly installed items. The permissions dialog boxes are blank for items where the permissions did not change.

Manually Setting Permissions

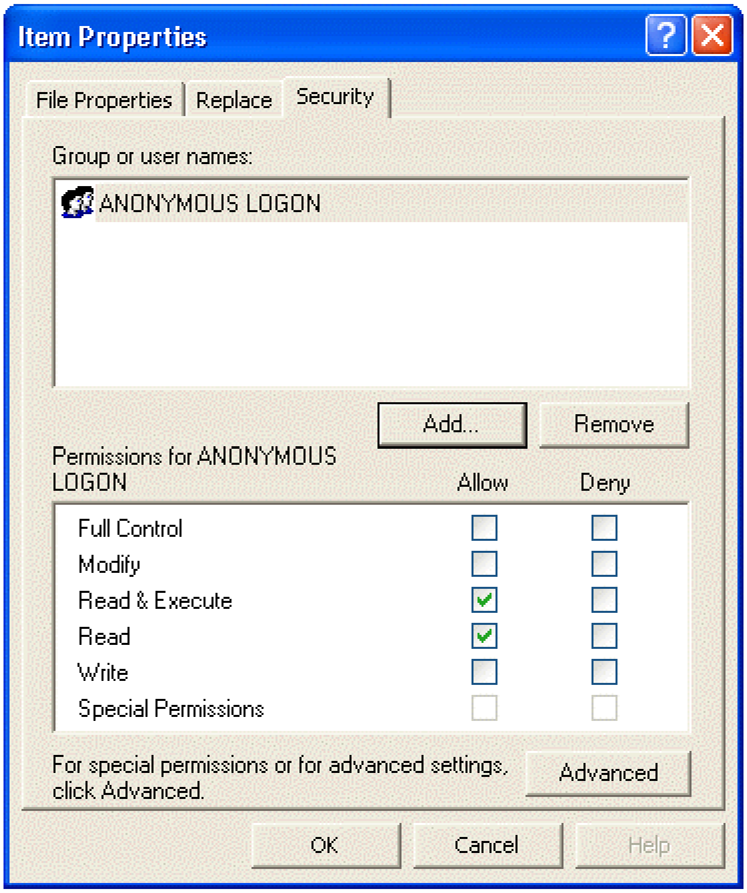
Permissions for an item in the Prism file can be set through the **Item Properties | Security** tab.

Installing Permissions

The permissions defined on the **Item Properties | Security** tab are added to or subtracted from items when the Prism file is installed on a target computer. The permissions of the current user do not generally matter during installation. Prism Deploy can add and subtract permissions regardless of the current user's ability to do so.

A Prism file containing permissions can only be installed if the target computer supports permissions. If the target computer does not support permissions, installation of the Prism file will fail. For example, a Prism file that adds permissions to a folder on the C: drive will fail on a system with a FAT hard disk—a format that does not support permissions.

Security tab



Group or user names:

ANONYMOUS LOGON

Permissions for ANONYMOUS LOGON	Allow	Deny
Full Control	□	□
Modify	□	□
Read & Execute	☑	□
Read	☑	□
Write	□	□
Special Permissions	□	□

For special permissions or for advanced settings, click Advanced.

The **Item Properties | Security** tab lists the objects (users, computers, domains, groups, and so on) being granted or denied permissions. It also lists the permissions that control the type of access they have to the item. If you are viewing the properties for an item in the **Deleted Items** portion of the Editor, then this tab lists the permissions that are to be taken away. Through this tab, you can add names and set specific permissions for each name.

The **Item Properties | Security** tab opens when you right-click on an item in the Editor main window, then click the Security tab.

Caution: Before using this tab, read Overview of Permissions for Items. Using this option without a thorough understanding of permissions could cause problems that are difficult to reverse.

Note: This dialog box is designed to follow the format of the Security property sheet in Microsoft Windows. Like the Windows interface, the appearance of this tab varies between the different versions of Windows. The fields or buttons on your computer may differ slightly from the descriptions provided here.

Users, Computers, and Groups

The box in the upper part of the tab lists the users and groups for whom permissions will be added to the item. To add more users or groups, click **Add** to add the names. Then define new permissions just as you would when changing the permissions in Windows. If you are editing a **Deleted** item, the permissions you define here are taken away from the existing permissions during installation.

- **Add**: Click the **Add** button to add a user or group that should have permissions allowed or denied for the item. The Select User or group dialog box opens.
- **Remove**: Click the **Remove** button to remove the highlighted name. The designated permissions will no longer be added to or subtracted from the user or group during installation.

Note: You cannot use Prism Deploy to remove inherited permissions. See Overview of Permissions for Items for more information.

Permissions

In the lower part of the tab, set the level and type of permissions for the name highlighted in the upper part of the tab. To allow or deny access to the item, select the appropriate check box. In most cases, the standard levels of permissions listed here are sufficient.

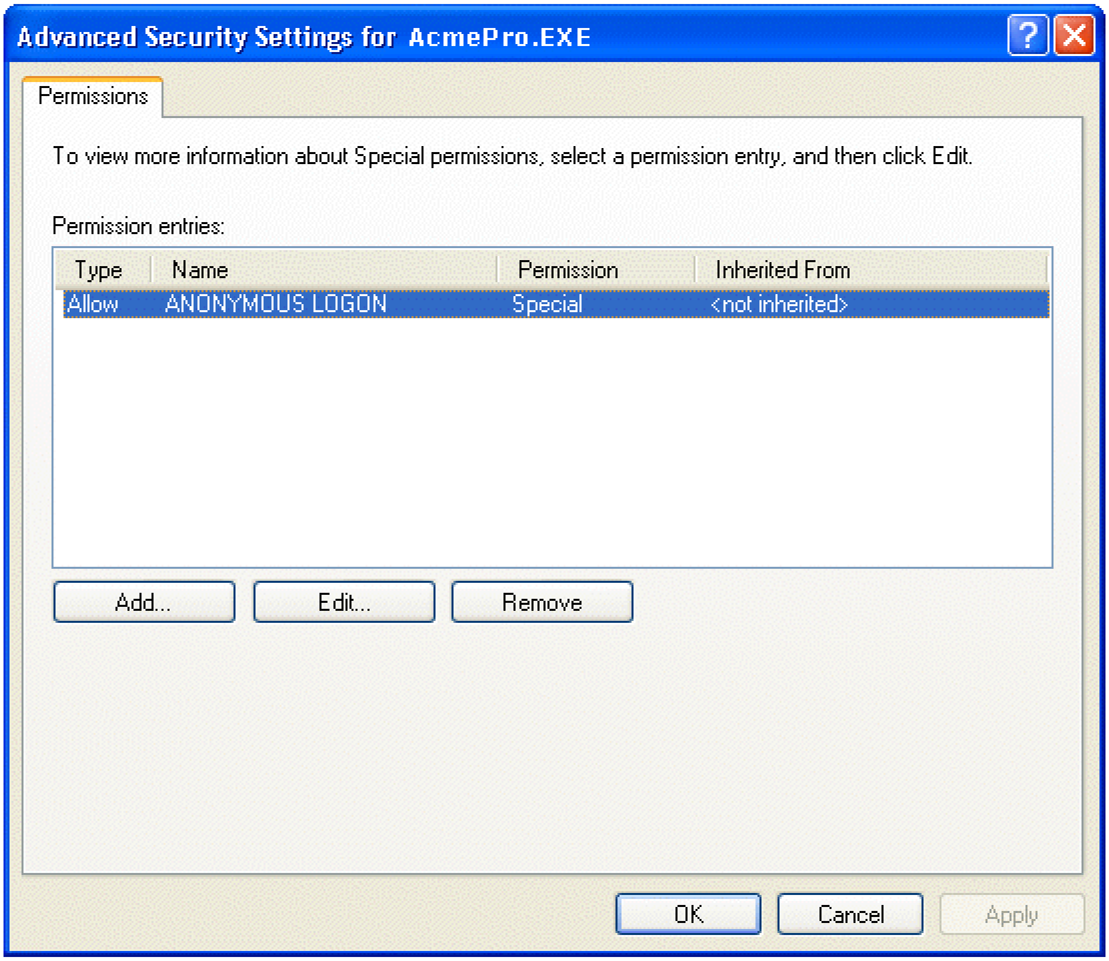
- ☑**Allow.** With this option selected, the user or group is granted the permission. Selecting **Full Control** automatically grants all permissions.
- ☑**Deny**: With this option selected, the user or group is denied this type of access to the item.

Caution: Be careful when using Deny. Deny takes precedence over Allow.

Special Access Permissions

To customize the permissions or set special access permissions, click the **Advanced** button. The Access Control Settings dialog box opens.

Access Control Settings dialog



To view more information about Special permissions, select a permission entry, and then click Edit.

Permission entries:

Type	Name	Permission	Inherited From
Allow	ANONYMOUS LOGON	Special	<not inherited>

The Access Control Settings dialog box gives a more detailed view of the information presented in the Item Properties | Security tab. Use this dialog box to customize the permissions to any level of detail.

The Access Control Settings window opens when you click the **Advanced** button on the **Item Properties | Security** tab.

Note: This dialog box is designed to follow the format of the Security property sheet in Microsoft Windows. Like the Windows interface, the appearance of this tab varies between the different versions of Windows. The fields or buttons on your computer may differ slightly from the descriptions and graphic provided here.

Add

To add a user or group, click the **Add** button. A new dialog box opens where you can add names to the permissions list.

After adding names, Prism Deploy opens the Permission Entry where you can set special access permissions or customize the permissions.

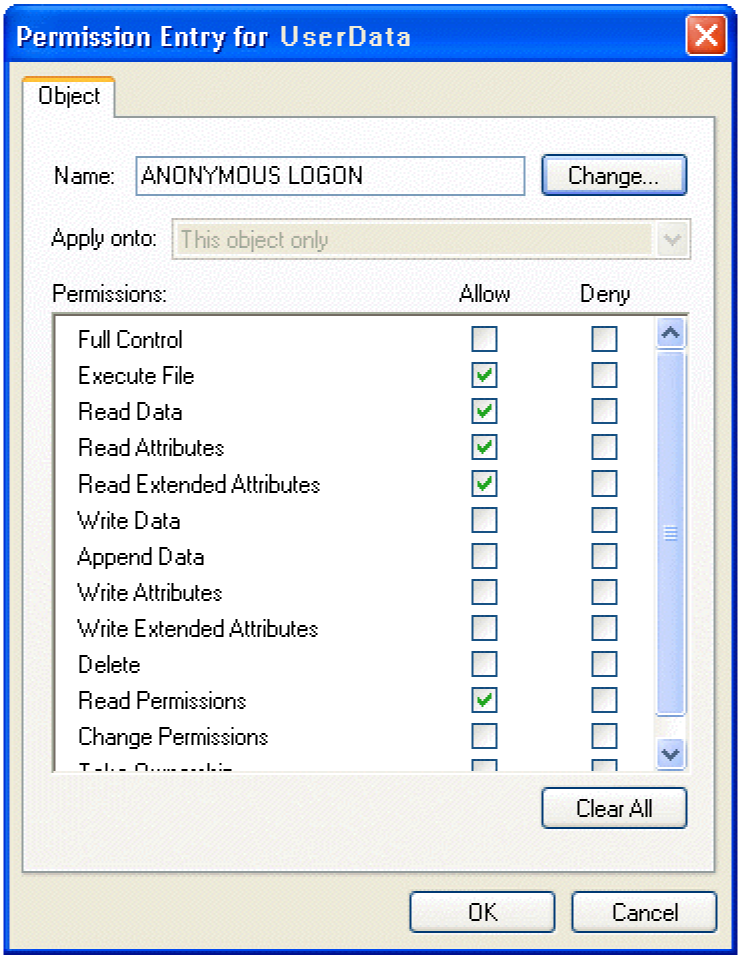
Remove

To remove the permissions set for a user or group, highlight the name and click the **Remove** button. The name is removed from the list and the permissions are removed from the item for this user or group.

View/Edit

Click the **View/Edit** button to set special access permissions or customize the permissions for the current item. When you click this button, the Permission Entry dialog box opens.

Permission Entry dialog



Name: ANONYMOUS LOGON

Apply onto: This object only

Permissions:	Allow	Deny
Full Control	□	□
Execute File	☑	□
Read Data	☑	□
Read Attributes	☑	□
Read Extended Attributes	☑	□
Write Data	□	□
Append Data	□	□
Write Attributes	□	□
Write Extended Attributes	□	□
Delete	□	□
Read Permissions	☑	□
Change Permissions	□	□
Take Ownership	□	□

The Permission Entry dialog box lets you set the permissions at the individual level, giving you finer or more granular control over the specific permissions granted for an item.

The Permission Entry dialog box opens when you click the **View/Edit** button on the Access Control Settings dialog box or the **OK** button on the **Select User or Group** dialog box.

> **Note**: This dialog box is designed to follow the format of the Security property sheet in Microsoft Windows. Like the Windows interface, the appearance of this tab varies between the different versions of Windows. The fields or buttons on your computer may differ from the descriptions and graphic provided here.

Options include:

- **Change**: Click the **Change** button to change the name of the user or group. The permissions are applied to the account listed in this text box.
- **Apply onto**: This option controls how the permissions are applied to this item and its child items. The default varies, depending on the type of item. For example, for folders, the default option is to apply the permission to the folder itself, along with any folders or files beneath it.

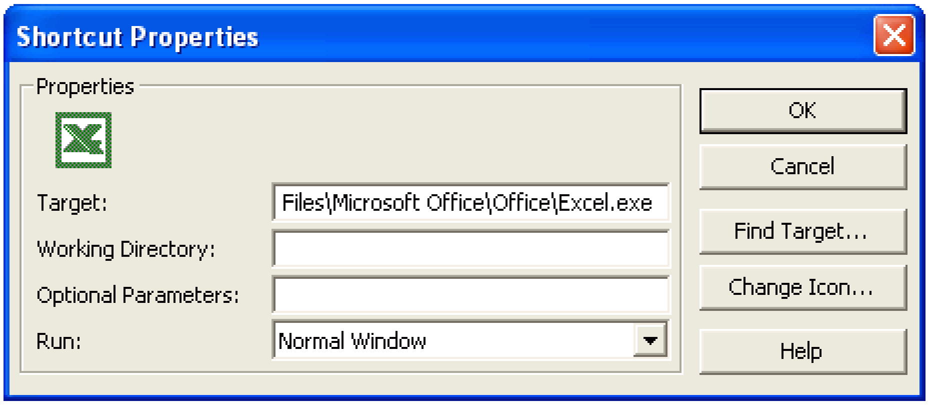
- **Allow/Deny**: Select the check box to allow or deny the specific permission.

  ☑**Allow**: With this option selected, the user or group is granted the permission. Selecting **Full Control** automatically grants all permissions.

  ☑**Deny**: With this option selected, the user or group is denied this type of access to the file or folder.

Shortcut Properties

Shortcut Properties dialog



Target: Files\Microsoft Office\Office\Excel.exe

Working Directory:

Optional Parameters:

Run: Normal Window

For the shortcuts in a Prism file, you can change the path, define a working directory and optional parameters, and change the icon that is displayed in the shortcut.

The **Shortcut Properties** dialog box opens when you double-click or right click on a shortcut in the Editor window and select **Modify** from the pop-up menu.

- **Target:** Shows where the executable for the shortcut is located. If necessary, enter a new path.

  Click the **Find Target** button to open the folder containing the executable file.
- **Working Directory**: If necessary, type the path to the working directory.

  Some applications need access to files in a particular directory in order to execute and must be launched from the folder where these files reside. This directory is referred to as the working directory. If your shortcut points to an executable and it needs a working directory specified, enter the path to the working directory.
- **Optional Parameters**: If your shortcut is a program and it needs command line arguments to run properly, enter them in the text box.

  For example, if you usually type **graph.exe /t** in the command line when starting a program called **Graph**, type **/t** in this text box.
- **Run**: Select an option from the drop down list to specify how the application will run when the user opens it.

  The options are **Normal Window**, **Minimized**, and **Maximized**. These are the same options available when defining a program’s properties with the Start menu’s **Settings | Taskbar** option.
- **Change Icon** button: Change the icon for the shortcut. The Change Program Icon dialog box opens.

Working with Variables

Prism Deploy lets you use environment and Prism Deploy variables inside registry settings, shortcuts, path names, and INI files. This feature lets you install Prism files even when the people receiving the file need personalized entries distributed, such as e-mail addresses. Variables also let you install one Prism file in a heterogeneous environment.

Prism Deploy recognizes some common folders and registry keys and automatically substitutes a variable for the path names or keys when it takes a Picture or finds changes. If Prism Deploy does not substitute a variable for a specific instance of a folder or key, you can include variables in the file properties or add a variable folder to the Prism file. Click the options listed below for an overview.

Variable Resolution Order

There are four ways a variable can receive a value. Prism Deploy uses the steps defined below, in the order given, to obtain the value for a variable when you install the Prism file containing the variable. The variable receives a value from:

1. The source you have defined (if any) in the Variable Source tab of the Variable Properties dialog box.
2. If the source you have defined is **This Variable**, Prism Deploy tries to get its value from the currently executing environment. Note that Prism Deploy inherits this environment from the process that launches it. Checking the value of the variable in another, parallel process (such as a command prompt window) may not show the same value that Prism Deploy will inherit.
3. If the variable is not defined after the previous steps, Prism Deploy uses the default value, if any, you defined in the Variable Name tab of the Variable Properties dialog box.
4. If appropriate, based on the settings you have defined in Variable User Prompt tab of the Variable Properties dialog box, Prism Deploy displays a window asking for the variable's value. The window also contains a **Cancel** button, letting the user cancel the operation at that point.
5. Finally, Prism Deploy will decide to continue installing the file based on whether the length of the variable's value (0 if it does not have one) meets the minimum and maximum length properties you have defined in the Variable Requirements tab of the Variable Properties dialog box.

Prism Deploy Substituted Variables

When Prism Deploy finds the following directories or key names, it substitutes a variable.

- For the Windows and System directories it substitutes WINDIR and WINSYSDIR, respectively.
- For keys in HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\currentversion\explorer\shell folders, it substitutes the key corresponding to the same name with:
-

APPDATA

CACHE

COMMON DESKTOP

HISTORY

NETHOOD

PERSONAL

COMMON PROGRAM

COMMON START MENU

COMMON STARTUP

COOKIES

DESKTOP

FAVORITES

FONTS

PRINTHOOD

PROGRAMS

RECENT

SEND TO

START MENU

STARTUP

TEMPLATES

Example of a Prism File Substituted Variable

For example, when Prism Deploy finds a reference to a Windows directory, it automatically converts the full path name into the variable WINDIR. This substitution lets you install the file on another computer without regard for where Windows files are located.

Let us say you want to apply the wallpaper setting "Sample."

1. Take a Picture of the computer.
2. Change the wallpaper setting to Sample.
3. Let Prism Deploy find the changes.
4. No matter where the Windows files are located, the resulting PWC file includes the following lines:

   In Registry (Current User) | Control Panel | Desktop:

   ab Wallpaper %WINDIR%\Sample.bmp

Prism Deploy automatically substituted the variable %WINDIR% for the path to the Windows directory. When you install the changes on another computer, Prism Deploy expands the variable by putting the correct path to the Windows directory in **Registry (Current User) | Control Panel | Desktop**.

Where Environment Variables Are Defined

There are many ways to set and use variables in stand-alone and network environments. Here are some common ways of defining variables.

Environment variables pertain to the entire Windows environment. Windows and all applications that support variables can use the environment variables. Common places for setting environment variables are:

- **autoexec.bat**. The temp variable, which tells Windows and other applications where to store temporary files, is often set in AUTOEXEC.BAT on Windows computers. For example, set temp = C:\Windows\temp.
- **Environment tab**. The Environment tab in Control Panel | System lets you define system or user variables. This tab is where the temp variable is defined.
- **Login scripts on a network**. The types of variables supported and how you set them depends on the type of network operating system.

Windows also sets its own environment variables. These variables represent important folders for the operating system. For example, a variable is defined for the windows folder (WINDIR). The purpose of this variable is to represent the path to the folder, no matter what the actual path may be. WINDIR may be C:\WINDOWS on a Windows 2000/XP or higher computer. These variables are defined when booting the computer and are stored in the registry.

To include a variable in a Prism file when an environment variable has not been defined, Prism Deploy lets you define an internal variable. These variables are only available for the current Prism file and only Prism Deploy can resolve the variable. (See the Variable Properties dialog box.)

Add Variables to Prism File Properties

In the File Properties | Variables dialog box, you can enter variables that will be resolved when Prism Deploy installs this file. These variables can indicate differing target paths, registry keys, and so on. The variable's source can be a file, another variable, an INI file, a registry key, or you can prompt the user to enter the variable's value.

Manually Entering a Variable

Use environment variables or internal Prism Deploy variables within INI files, shortcuts, or registry data. If you are using an internal variable, define the variable before beginning the following steps. (See Item Properties.)

To insert the variable:

1. Locate the file or registry data where you want to insert the variable (for example C:\WINDOWS\MAIL.INI) and open it for editing.
2. Insert the variable. The variable name must be enclosed in quotes and the percent (%) sign.

   For example, inside C:\WINDOWS\MAIL.INI, find the [email address] section and replace your user name with "%USERNAME%".
3. Save the file or registry data and close the Editor.

Prism Deploy resolves the environment or Prism Deploy variable, in this case %username%, when the Prism file is installed. If Prism Deploy is unable to resolve an environment variable, it distributes the variable "as is," percent (%) sign and all. You can prevent this from happening by explicitly adding a "username" variable to Prism Deploy's variable list.

Where Are Percent Signs Needed?

When adding variables to Prism files, you need to enclose them in percent signs in the following areas:

FILE CONTENTS (such as INI files)

REGISTRY VALUE DATA

In all other areas such as folder and file names, registry key names, and registry value names, use the Prism Deploy Editor special menu options for creating a variable. You do not need to use the percent sign when using variable folders, variable registry keys, and so on.

Adding a Variable Registry Entry

Prism Deploy lets you add a registry key or registry value containing a variable to a Prism file. For example, if you want to distribute a Prism file that creates a unique registry value on each user's computer, set up a Prism Deploy variable to create the unique value each time the Prism file is installed.

You can use either an environment variable that is set outside of Prism Deploy or an internal variable that is set within Prism Deploy. If you choose an internal variable, you must first define the variable. (See Where Environment Variables Are Defined and the Variable Properties dialog box.)

1. On the left side of the Editor window, click on the registry key where you want to create a new variable registry key or value.
2. Select **Edit | New**.
3. Choose either **Variable Registry Key** or **Variable Registry Value** from the drop-down menu.
4. In the text box on the **New Variable Registry** dialog box, type the name of the environment variable or Prism Deploy variable that defines the new entry.
5. Click **OK**. The new variable is displayed in the Editor window.

Variable Folders

Add Variable Folders to Prism Files

Prism Deploy lets you add a variable folder, containing an environment variable, to a Prism file. For example, use a variable folder if you want to distribute a Prism file that updates a DLL in an application directory, but the directory could be different on multiple computers. In this case, set up a Prism Deploy variable to search for the application executable file and install the DLL in the same directory.

Adding a New Variable Folder

Prism Deploy lets you add a variable folder containing an environment variable to a Prism file.

1. In the left side of the Editor window, click on the drive or folder where you want the new variable folder located.

   For example, click on **C:** to put the folder in the C: drive or click on **Windows** to put the folder in the Windows directory.

2. Click 📁 in the toolbar and select **Variable Folder** from the submenu.

   The **New Variable Folder** dialog box opens.

3. In the text box, enter the name of the environment variable that defines the new folder. Click **OK**. The new variable folder is displayed in the Editor window.

Changing Existing Folders to Variables in Prism Files

If you already have a folder that you want to turn into a variable, the simplest thing to do is to edit the paths for the file. (See Changing File Path Names for more information.)

This method is very powerful because Prism Deploy automatically changes all references to a folder (in INI files, the registry, and so on) to the variable you request.

Saving Files

Saving a Prism File

A Package is saved automatically by Prism Deploy as a .PWC (Portable Windows Change) file when you close it. You also have the option of saving a copy of the Package under a different name, or saving it as a Prism Executable Package (.EXE).

Saving or Duplicating a Prism File from the Editor Window

When you close a Prism file, Prism Deploy automatically saves it. However, if you want to duplicate the Prism file you have open in the Editor window or change it to a different type of Prism file, use Save As to create a copy under a different name.

1. Select **File | Save As**.

   Prism Deploy displays the **Pick a file to save to** dialog box.
2. Select the drive or location in the **Save in:** field.
3. Type a name for the file in the **File name:** field.
4. Select the type of file you want to save from the **Save as type** drop down list.
5. Click **Save**.

Prism Deploy saves the file under the new name and displays the contents of the Prism file in the Editor window. You can now edit this file.

Creating a Duplicate Prism File with a New ID

When you select **File | Save As** and save the Prism file under a new name, Prism Deploy gives the new file the same identification number (ID) as the original file. The Prism Deploy Rollback file, along with utilities for scheduling and uninstallation information on the computer, considers these two files to be the same. To create two unique versions of the same file, each with unique uninstallation data, save the file as follows:

1. Save a new version of the Prism file with **File | Save As**, as described above.
2. With the new Prism file open, click 📋 or select **File | Properties.**
3. Select the File Properties | Information tab.
4. Click the **New ID** button. A new ID number is generated.
5. If you are duplicating a Prism Executable Package, enter a new description in the File Properties | Add/Remove tab for the Control Panel's Add/Remove Programs.

Alternate Methods for Distributing Prism Files

Prism Deploy gives you a number of options for distributing Prism files. The most direct and efficient is to use the Prism Console. The Console lets you set up your Prism files for installation, maintain a permanent record of the computers where they are installed, and initiate installation over the network or to remote users with just a few clicks. If you prefer to use a command line or other distribution tool, Prism Deploy also supports a wide variety of alternate methods for distribution.

In each case, the Prism Client, which resides on the target computer, lets you install the Prism file.

How the Prism Client Works

Prism files are easily distributed to computers using the Prism Deploy Client. Before you decide how to install Prism Deploy, you need to understand how the Prism Client accomplishes its tasks: either through the Console, Prism commands, and/or through Prism Deploy shell extensions.

The Client, which is installed on each target computer, communicates with the Prism Console and provides the vehicle for installing software and making other changes on a target computer. The Prism Deploy Client also executes commands that are invoked via login scripts, batch files, third-party management suites, and Prism Deploy's own shell extensions. These features let you use a variety of methods for distributing Prism files to target computers.

Note: You cannot edit Prism files using the Prism Client, because the Client does not have a user interface. You must use the Editor to edit Prism files.

Distribution via E-mail

Just write and address the e-mail message, then attach the Prism file. Prism Deploy’s shell extensions install the contents of the Prism file when the recipient double-clicks the attachment.

The only limitation to this method is that the e-mail server or program may limit the size of files that you can attach to a message. If so, use this method for smaller Prism files.

Link the Prism File to a Web Page

Use this method to deliver Prism files to remote laptop users or place the Prism files on the corporate intranet page for general use. Users simply click on the link to install the Prism file.

Note: The Prism Client shell extensions must be installed on the target computers in order for the Packages (.PWC) to launch automatically from a Web page.

Edit the HTML source file to create a link to the Prism file. For example, if the Prism file is named SR3.PWC, it is stored in the same folder as the Web page, and the name of the link that will be displayed on the Web page is "Install MyApp for Service Release 3," add this line to the HTML source file:

```
<A HREF="SR3.PWC">Install MyApp for Service Release 3</A>
```

Distribution Using a Network Login Script

Add a command to the NetWare or Windows logon scripts to have Prism files or scripts run at each logon. Here is a NetWare example to give the "Graphics" group the settings to install and use the new Laser 7000 printer:

```
IF MEMBER OF "GRAPHICS"
@"C:\Prism Deploy\ClientInstallFiles\ptclient.exe /InstallPackage" "F:\Prism
Deploy\Laser_7000.pwc"
END
```

The following is an example to give the "Headquarters" group the latest service release for the corporation's productivity suite. You must first create a batch file that has the following command and save the file to the NETLOGON shared location on the server. Then assign the "Headquarters" group the batch file using the server User Manager program.

```
"C:\Prism Deploy\ClientInstallFiles\ptclient.exe" /InstallPackage "F:\Prism Deploy\OFFXP.pwc"
```

More Information

See "Scripting Prism Deploy Tasks" in the *Deployment Guide* or the Prism Console online Help for information about the command line options and syntax.

Distribution Using a Network Management Suite

Using Prism Deploy commands, Packages are ideal for distribution using network management suites. Network management suites, such as Microsoft's Systems Management Server, Novell's ZENworks, and Tivoli's Management Framework, are used to roll out packaged software to workstations on the network. This roll out is usually accomplished by configuring the suite software to run a command on targeted systems. Prism Deploy commands underlie all of the Client's operations, and can be inserted directly into the management suite's distribution mechanism.

Distribution Using Prism Deploy Commands

Prism Deploy commands automate the Client to perform most Prism Deploy functions in the background. The commands and command switches define exactly how and when the Client executes a command. Use Prism Deploy's commands singly, grouped in Prism Deploy scripts, or in batch files.

Use the command line when you want to automate common tasks such as distributing an application, installing an update, or uninstalling an application that is no longer needed. You can complete Prism Deploy tasks by calling the appropriate commands from a batch file, a login script, the Startup folder, or any shortcut on the computer. You can even group multiple Prism Deploy tasks in a single text file to create a script.

For More Information:

See "Scripting Prism Deploy Tasks" in the *Deployment Guide* or the Prism Console online Help for information about the command line options and syntax.

Distribution Using a Scheduling Utility

Schedule a recurring task to reinstall Prism files on a regular basis or schedule a task to **occur** just once to deliver a new application to remote users. Add the new task to your scheduling utility to tell the Prism Client (PTCLIENT.EXE) to install a Prism file. Use the Task Scheduler in Microsoft Internet Explorer, the Task Scheduler service on Windows, Norton Scheduler or any other utility that lets you run an executable at a specified time.

Distribution Using Windows Explorer

Since the Prism file contains all the files needed to correctly install its contents, all you need to do is to double-click on a Prism file through any Windows browser. The shell extensions do the work! For example:

- If the Prism file installs a large software application that you need to share between work and home, you might decide to store it on a USB drive.
- If you want to be able to repair or reinstall a malfunctioning application while you are on the road with your laptop, you can create a CD-ROM containing the Prism files for all your critical applications. Then, insert the CD-ROM and run the Prism file again to reinstall or "fix" the software.
- If you are on a network, put useful Prism files on a shared network drive.

Distribution Using a Windows Start-up File

To have a Prism file run when you start Windows, place a shortcut to the Prism file in any of the following places, either for all users of a computer or for just a specific user:

- C:\Documents and Settings\All Users\Start Menu\Programs\Startup
- C:\Documents and Settings\<username>\Start Menu\Programs\Startup
- C:\WINDOWS\Start Menu\Programs\StartUp
- C:\WINNT\Profiles\<username>\Start Menu\Programs\StartUp
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce

Shell Extensions

Distribution Using Prism Deploy Shell Extensions

Prism Deploy's shell extensions register several new file types in the local registry. If the shell extensions have been installed on the computers, users can install and uninstall Prism files simply by double-clicking or right-clicking on Prism files they have found in Windows Explorer, received as e-mail attachments, or downloaded from the corporate intranet. Prism Deploy's shell extensions give you great flexibility for distributing Prism files.

Prism Deploy Shell Extensions

Prism Deploy’s shell extensions register several new file types in the local registry. If the shell extensions have been installed on the computers, users can install and uninstall Prism files simply by double clicking or right clicking on Prism files they have found in Windows Explorer, received as e mail attachments, or downloaded from the corporate intranet.

For example, a network administrator can attach a Prism file to an e mail message giving the recipients an update to their antivirus software. Likewise, users can browse to a drive on the network where Prism files are stored and simply double click on a file to install it on their computer. If you plan to distribute Prism files through the corporate intranet, users simply double click to install the Prism files shown as links on the Web site. You can configure Prism files to contain detailed "before installation" messages so users know exactly what will happen when they install the file.

Caution: Prism Deploy’s shell extensions give you great flexibility for distributing Prism files. However, if you have strict requirements about what users may and may not install on their computers or if you want only network administrators to install or uninstall Prism files on users’ computers, then you might decide against installing the shell extensions on the computers.

Context Menus

Prism Deploy registers several shell extensions, but only the following types display context menus: PWC (Package files), PWF (Picture files), PWR (Uninstall files), and EXE (Prism Executable Packages).

PWF (Picture Files) Shell Extensions

Edit opens the Picture in the Prism Deploy Editor for modification. (Only available if the Prism Deploy Editor is installed.)

PWC (Packages) Shell Extensions

Open tells Prism Deploy to install the selected Package. Prism Deploy’s default setup prompts the user with the message, "Are you certain you want to install the file <filename>?" The user can answer **Yes** or **No.**

Print means the contents of the Package will be printed to the user’s default printer.

Edit opens the Package in the Prism Deploy Editor for modification (only available if the Prism Deploy Editor is installed).

Uninstall tells Prism Deploy to uninstall the changes it made by installing the Package.

PWR (Uninstall Files) Shell Extensions

Uninstall tells Prism Deploy to uninstall the changes it made by installing the Prism file associated with this Uninstall file.

EXE (Prism Executable Package) Shell Extensions

Open tells Prism Deploy to install the selected Prism Executable Package. The Prism Executable Package does not display a prompt to the user prior to installation, unless a prompt was added to the Prism Executable Package through the Prism Deploy Editor. (See Messages Tab and Variable User Prompt Tab for information on setting user prompts.)

Edit opens the Prism Executable Package in the Prism Deploy Editor for modification (only available if the Prism Deploy Editor is installed).

PTS (Prism Deploy Script Files)

Run launches the script and plays all commands written to the script. The user will be asked "Are you sure you want to play the script <name>?" The user can answer **Yes** or **No**.

Print means the contents of the script are printed to the default printer.

Edit opens the script in your default text editor for modification. (See "Script Statements" in the *Deployment Guide* or Console online help.)

SMS and Network Management Suites

Integrating Prism with Microsoft SMS and other Network Suites

Prism Deploy compliments SMS (Microsoft Systems Management Server) and other network suites. The installation of Prism files is independent of the method you use to deliver them, so they can easily be distributed through these distribution systems that you already have in place. Also, any Prism Deploy command can be carried out through one of these other tools.

Prism Deploy includes options for integrating directly with SMS. When you build a Prism file in the Editor, you have the option of creating a Package Definition File (PDF) and/or a Management Information File (MIF). These files ease the process of deploying a Prism file from SMS, while ensuring information about the Prism file is recorded in the SMS reporting and inventory.

Naming the PDF or MIF Created by the Prism File

The File Properties | SMS tab helps you integrate Prism files into SMS's (Microsoft Systems Management Server) reporting and information gathering functions. On this tab, you can specify the name of a file to be created when the Prism file is installed on a target computer. These types of file can be created:

- PDF (Package Definition File for use with SMS version 1.x) or SMS (for use with SMS version 2.0)

  The PDF/SMS file records information you specify on this tab, along with information about the Prism file properties for use when building an SMS Package. This information is automatically pulled in when you create a new Package from definitions in the SMS Administrator Console
- MIF (Management Information File)

  When it is installed, the Prism file creates the MIF in the %WINDIR% directory on the target computer. SMS retrieves this file to gather information about the success or failure of the Prism file installation. This information is reported in the advertisement status area of the SMS Administrator Console

Note: The Prism file created does not require SMS for installation. The Prism file creates the additional file that you specify here, but its operation and characteristics do not change.

To name the PDF or MIF created by the Prism file:

1. Open the Prism file in the Prism Deploy Editor.
2. Click 📋 in the toolbar or select **File | Properties**.
3. Click on the File Properties | SMS tab.
4. Enter the Product, Publisher, and other information.
5. Select a check box for creating a PDF, MIF, or both types of file. Click **OK**.
   - When you close the Prism file in the Editor, the PDF is created in the same location as the Prism file.
   - When the Prism file is installed or uninstalled, the MIF is created in the %WINDIR% directory.

Creating an SMS Package Referencing a Prism Deploy Command

Prism Deploy can be used in conjunction with SMS's (Microsoft Systems Management Server) distribution method to invoke Prism Deploy commands and install Prism files on SMS client computers. Any Prism Deploy command can be carried out in this way, including /InstallPackage, /UninstallPackage, /TakePicture, /FindChanges, and so on. (See "Scripting Prism Deploy Tasks" in the *Deployment Guide* or the Prism Console online Help for information on the commands.)

It is beyond the scope of this online help to describe all the procedures needed to create SMS packages and schedule jobs. However, the following covers the main steps and the specific SMS integration features in Prism Deploy. (See the SMS documentation for details.)

Before you Begin

- SMS site and logon servers must be properly configured.
- SMS client software must be running on the target computers.
- Prism Deploy service must be installed.

Note: See the SMS documentation for detailed information on the SMS settings and options.

Distributing Changes with SMS

To use Prism Deploy and SMS together to distribute changes:

Step 1. Create a Prism file or script that contains the software, files, updates, or actions that you want to distribute.

- Enter information on the File Properties | SMS tab to specify the information recorded when the Prism file is installed.
- The setting for **Installation Method** in the File Properties | Requirements tab must be left with the default setting—Current User Only. You will use SMS to control whether the Prism file goes only to the currently logged-in user or to all users of the computer.
- Copy the Prism file and its PDF to a SMS distribution folder, typically a shared location on a server. If the Prism file is not a Prism Executable Package, also copy the Client executable (PTCLIENT.EXE) to the same location.
- After creating the Prism file, customizing it as needed, and testing the file, you are ready to distribute it with SMS.

Step 2. Create an SMS Package.

Please refer to your SMS documentation for more detailed information.

Examples of Using Prism Files and Commands with SMS

The following examples illustrate ways to use Prism files with SMS:

- Create a Prism file for applications and application upgrades. Then create an SMS Package and schedule an Advertisement to distribute the Prism file to users.
- The SMS Package can include a Prism Deploy command to install a single Package, or it can call a Prism Deploy script that carries out a number of tasks and install multiple Prism files.
- Use an SMS Package containing a Prism command to take customized back-up Pictures of users' computers by using the /TakePicture command with a custom Picture definition (.PWI) file.
- Distribute the settings users need to connect to a new network printer. Just create the Prism file, build an SMS Package and Program, then Advertise it.
- You have recently picked up a number of new tips for adjusting the registry through a Microsoft certification class you attended, and you would like to make those changes on the computers you manage. Create a Prism file with the changes to the registry, then create an SMS Package and Program with a corresponding Advertisement to let SMS distribute the changes.

Using ZENworks and other Network Management Suites with Prism Deploy

Many network management suites have the capability to roll out their own "packages" to computers on the network, including Novell's ZENworks®, Tivoli® Enterprise Console, and so on. Since the Prism Deploy Client is a command-driven program, Prism Deploy Packages are ideal for distribution using these tools. Prism Executable Packages can also be distributed easily through network management suites. (See the following sections for more detailed information on some of these tools.)

Ideas for Use

- Create Prism files for common applications, then use the network management tool to deliver them to users.

- Use the network management tool to redistribute a shortcut to an important application that a user has accidentally deleted. The Prism file you distribute can contain either the entire application, or just the missing shortcut.
- Distribute a ZENworks package to call Prism Deploy's /UninstallPackage command to uninstall software that you previously installed with Prism Deploy.
- Write a Prism Deploy script that takes a backup Picture of the registry and other important files on users' computers. Use ZENworks to schedule the script to run once a week.

Creating a NAL Application Object Referencing a Prism Deploy Command

Refer to Novell's documentation for information about installing and using ZENworks. The following steps provide a brief overview:

1. Before you begin, in order to use NAL with Prism Deploy, you must:
   - Install Prism Deploy on a server where users have access.
   - Install ZENworks on your server
   - Install ZENworks snapins so you can create Application Objects within NWAdmin
   - Be sure that users have access to the ZENworks program to receive the applications that are assigned to them.
2. Run NWAdmin and create a new Application Object.
3. Give the application object a description and fill in the path to the Prism Executable Package or Prism Client executable PTCLIENT.EXE.
4. In the NAL field for command line parameters, specify an appropriate Prism Deploy command line switch, followed by the appropriate parameters, if using PTCLIENT.EXE to install a Package. (See "Scripting Prism Deploy Tasks" in the *Deployment Guide* or the Prism Console online Help for information on the commands.)
5. Fill in properties for the ZENworks Application Object. You do not need to use the settings for registry, INI files, application files, text files, and icon/shortcuts because Prism files already contain all those settings.
6. Assign the Application Object to users or groups.

Uninstalling Prism Files

Understanding Prism Deploy Uninstall Files

Prism Deploy's Uninstall files (.PWR) hold the compressed information that Prism Deploy needs to put a system back the way it was before a Prism file was installed. If you do not want to keep the changes made by Prism Deploy, you can simply uninstall them. Below are some key points that explain how Uninstall files work for Prism Deploy Packages and Prism Executable Packages.

Prism Deploy generates two Uninstall files when a Prism file is applied. Prism Deploy separates the Prism file into user and computer-specific settings during the file installation and creates a specific Uninstall file for each. When uninstalling the Prism file, you can do any of the following:

- Uninstall the entire Prism file
- Uninstall only the user-specific settings or only the computer-specific settings using the command line switches /CommonInfo and /UserInfo. (See "/UninstallPackage" in the *Deployment Guide* or the Console online help for information on the command lines.)

Uninstall files are unique to each target system. For example, Machine A may already have the same version of a DLL that is contained in a Prism file. Therefore, when the Prism file is installed and the Uninstall file is created on that system, the Uninstall file contains information about incrementing the DLL counter. Machine B has an older version of the DLL, so the Prism file updates the DLL, and the older version is stored in the Uninstall file, in case the Prism file needs to be rolled back.

If a Prism file is installed more than once on the same system without rolling back a previous installation, Prism Deploy tracks this information in the registry and stores additional Uninstall files. Each new Uninstall file tracks only the information that was changed by each subsequent installation of the Prism file. If you decide you do not want to keep the changes made by the file, you will have a choice: undo just the changes from the last time the file was installed or undo all the changes from every time the file was installed.

Uninstall files do not undo changes unless they are the same changes that Prism Deploy installed on the computer. For example, a Prism file may install a new version of a DLL; the Uninstall file stores a copy of the older DLL. If another program updates that DLL in the meantime and then you decide to uninstall the Prism file, Prism Deploy does not replace the DLL with the older version stored in the Uninstall file. The version now on the system is not the same version of the file that Prism Deploy installed.

Where Prism Deploy Stores Uninstall Files

- In the default location, based on your operating system and individual computer settings.
- In the folder that you specified under **Package | Options** on the Uninstall tab.
- In the folder that you specified in the registry of the target computers : HKCU\SOFTWARE\Lanovation\PictureTaker\Settings\Directories\Unapply Folder = [path]

Uninstalling Prism Files

Prism Deploy offers a variety of methods to uninstall Prism files. The best method depends in part on the mechanism you used to install the file initially.

Uninstalling Changes with the Console

In many cases, the Console is the best method for uninstalling Prism files. (See the *Deployment Guide* or Console online Help for more information.)

Alternate Methods for Uninstalling Changes

- Uninstalling Changes with the Editor
- Uninstalling Prism Files Using the Shell Extensions
- Uninstalling Prism Files Using the Command Line
- Uninstalling Prism Executable Packages

Uninstalling Changes with the Editor

If you used Prism Deploy's Editor to install new software, for example, you can quickly remove the software by uninstalling the changes.

Click ♻ in the toolbar to select the file you want to uninstall.

Note: This method is only applicable to files installed on your own computer.

Uninstalling Prism Files Using the Shell Extensions

You can select a Prism file to uninstall directly from Windows Explorer.

1. In Windows Explorer, right-click on a Picture file (.PWF), Package file (.PWC), or Prism Executable Package (.EXE).
2. Select **Uninstall** from the drop-down menu.

   Prism Deploy opens and displays a confirmation message asking if you want to uninstall the selected file.
3. Click **Yes** to continue. Click **No** to cancel the uninstall.

Uninstalling Prism Files Using the Command Line

You can also use the command line to uninstall Prism files. For example, use the command line to uninstall a Prism file for an application that is no longer needed. The command for uninstalling a Prism file is **/UninstallPackage** or **/UP**. (See /UninstallPackage in the *Deployment Guide* or Console Help for details on the supported command line options and how to use them.)

Examples

Here are two examples of using the command line to Uninstall files.

- The network administrator has rolled out a version of word processing software that now needs to be upgraded. The software manufacturer recommends uninstalling the older version before installing the new version.

  ```
  ptclient.exe /UninstallPackage oldsoftware.PWC
  ```

- You and five of your colleagues are evaluating three network analyzer packages for an upcoming purchase. You have built Prism files for the three different versions and each have taken turns evaluating the software. Now that you have made a decision, you need to uninstall the other two versions. Use the prompt switch to confirm the removal of the Prism files.

  ```
  ptclient.exe /UninstallPackage eval1.pwc /prompt
  ```

For More Information:

See "Scripting Prism Deploy Tasks" in the *Deployment Guide* or the Prism Console online Help for information about the command line options and syntax.

Uninstalling Prism Executable Packages

If you want to give the recipients of the Prism Executable Packages the ability to remove or uninstall the software or configuration changes you are delivering, you have two options:

- If you are delivering a software application, the application itself may create its own "uninstall" option when you originally build the Prism Executable Package. In that case, Prism Deploy captures the uninstall method and delivers it as part of the Prism Executable Package. Most software applications handle this uninstall option by putting an entry in the Add/Remove Programs area of the Control Panel and placing an uninstaller program on the hard drive.
- If you are delivering configuration changes, new files or folders, or software that does not automatically create an uninstaller, check Add an entry to Add/Remove Programs when the file is run check box in the Prism Executable Package's properties. (See the File Properties | Add/Remove Programs tab.) You must also enter the description to be entered in the Add/Remove Programs area of Control Panel.

When you use the second option to configure the rollback capability, Prism Deploy creates a folder called SIFXINST in one of the following folders on the target system:

C:\Program Files

C:\Program Files\Common

C:\Windows\System

C:\Windows

C:\

It then places an uninstaller program called SIFXINST.EXE and the Uninstall files for the Prism Executable Package in the folder. The names of the Uninstall files are the same as the Prism Executable Package with a number appended for each additional time the file is installed. The extension of the files is .PWR.

Syntax & Variables for Installation Rules

Rule Variables and Operators for Installation Requirements

The rules and variables covered in this section include those used for creating installation rules on the **File Properties | Requirements** tab in the Editor or creating scripts with the Prism Deploy /ScriptVer command. Rules can be built using the drop-down lists in the Rule Expert or typed directly in the Rules Editor. If you type a rule directly, the values for predefined variables must match those set up by New Boundary Technologies.

Whether it is created with the Rule Expert or typed directly in the Rules Editor, each rule is composed of some or all of these parts in the following order:

1. **Variable type**: Identifies the type of item you are evaluating—variable, registry key or value, INI value, file, file version, file name, folder name, and so on.
2. **Item**: If the Value Type requires an identifier (those value types enclosed in brackets, for example <File>), identify the item here. For example, with the Value Type <File>, enter the path and name of the file. For example:

   <File> "C:\Program Files\Acme Software\Acme.exe"
3. **Comparison Operator** or **Logical Operator**: Defines how the value on the target computer is compared to the rule. For example: =, Exists, Matches, and so on.
4. **Value** or **Constant**: The value to compare with the resolved variable's value on the target computer.

Notes on Rule Syntax

When typing rules and their values:

- Variables support registry value names containing back-slashes (\). In a registry value path, any characters following the first double-backslash (\\) are interpreted as part of the registry's name.

  If a registry value type is used and there is not a double-backslash, then the default value for the registry key is used. For example:

  "HKLM\Software\Foobar\\Some Value" : the registry value name is "Some Value"

  "HKLM\Software\Foobar\Another Value" : the registry value name is "Default"

- Rules are not case sensitive, except for size value abbreviations ("500 MB" is acceptable, but "500 mb" is not). They can by typed in all caps, lower case, or a combination of the two.
- A value must be enclosed in quotes (" ") in these cases:
  - the value contains spaces (for example, "c:\Program Files\Any Company\Any_exe.exe")
  - the value is a string (for example, "computername")
- Constants are not enclosed in quotes (" ").
- Quotes can be embedded by typing two quotes for every embedded quote.
  For example, the following rule evaluates to true if there is a registry value "Genuine" Intel in the key HKLM\hardware\description\system\centralprocessor:

  Exists <Registry Value> "HKLM\hardware\description\system\centralprocessor\\""Genuine"" Intel"

Constants

For your convenience, several of the variables have constants assigned. When using the Rule Expert, the constants are available in a drop-down list after you choose the variable type. If you are typing the rule directly, the constants must match the values set up by New Boundary Technologies.

See the discussion under each variable for information on the constants.

Comparison Operators

The following comparison operators are supported when creating installation rules on the **File Properties | Requirements** tab in the Editor, setting up user-defined configuration groups in the Console, or creating scripts with the Prism Deploy **/ScriptVer** command:

- All comparison operators must be followed by a value.
- All comparisons are case sensitive

Comparison Operators

=	The value on the computer is equal to or matches the stated value exactly. (See note below.)
<>	The value on the computer is not equal to or does not match the stated value exactly. (See note below.)
<	The value on the computer is less than the stated value.
<=	The value on the computer is less than or equal to the stated value.
>	The value on the computer is greater than the stated value.
>=	The value on the computer is greater than or equal to the stated value.
exists	The item exists. Use this option for identifying whether a specific file, folder, registry key, registry value, and so on exists on the computer. For example, use this option to determine whether an application is installed on the computer before sending it an update.
not	Reverses or inverts the meaning of the clause that follows it. For example: `NOT EXISTS <file> "C:\Oracle\DB.BIN"`
matches	The matches operator uses wild-card characters to determine whether or not an item matches the corresponding value. The portion of the statement that contains wild-cards must be surrounded by quotes. The DOS wild-card system is used: ? represents one character of any value * represents 0 or more characters of any value. For example: `IP MATCHES "*.183.*.*"`
grepmatches	Use a GrepMatch to identify a more complex pattern than those available with the other operators.

Note: The operators = = and ! = (used with previous versions of Prism Deploy are equivalent to = and < > respectively:

If you use = = in a rule or script, Prism Deploy translates it to =.

If you use ! = in a rule or script, Prism Deploy translates it to < >.

Examples Using Grepmatches

The following examples demonstrate the use of GrepMatch for the operator. For more information, visit the support area of the New Boundary Technologies Web site (www.newboundary.com/support/support.htm) for links to additional resources.

The group created with this match contains computers that have an IP Address within the range 192.168.1.*, 192.168.2.*, and 192.168.3.*:

```
IPAddress GREPMATCHES "192\.168\.[1-3].*"
```

The group created with this match contains computers with any of these letters as the first letter of their CD-ROM drive—A:, B: C:, or D:

```
FirstCDLetter GREPMATCHES "[A-D]"
```

The group created with this match contains computers where the registry value FileFilter contains *.olb as its first delimited value.

```
<Registry Value> "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\DbgClr\7.0\Object
Libraries\{18E32C04-58BA-4a1e-80DE-1C291634166A}\Filters\1\\FileFilter" GREPMATCHES
"\*\.olb.*"
```

Examples Using the Comparison Operators

These examples show the operator for a rule that identifies computers with less than 64MB of RAM and the operator that identifies computers that use any language other than U.S. English.

```
RAM < 64MB
SYSTEMFULLLANGUAGE <> "Language.Full.English.UnitedStates"
```

Conjunction

You can add additional or contrasting criteria to a rule with the following operators:

- **AND**: Prism Deploy identifies the computers that meet **both** or **all** sets of criteria defined by the rule. The Prism file is installed only on these computers or only these computers are selected for the group.
- **OR**: Prism Deploy identifies the computers that meet **either** or **any** of the criteria defined by the rule. The Prism file is installed only on these computers or only these computers are selected for the group.

Conjunctions are evaluated left to right, except if precedence operators—parentheses ( )—are used in the statement.

Note on Parentheses

Use parentheses to override the normal left-to-right evaluation of a rule. In some cases, use of parentheses changes the outcome of the evaluation. For example:

```
(CPU = "Pentium 5" AND OSVERSION = "Windows XP" ) OR (CPU = "Pentium 4" AND OSVERSION =
"Windows XP")
```

Values

The values are the sequences of text or numeric characters that are compared with the items on the target computers. Values must be enclosed in quotation marks if they contain spaces or are any current or future logical operator. For example, "here OR there".

Note: Comparison of values is not case sensitive, except for size abbreviations. The size abbreviations KB, MB, GB, and so on must be capitalized.

Prism Deploy rules support these types of values:

- **Number**: When a value must be compared against an item, the comparison is numeric if both the item and the value are numeric.
- **Size**: A size is the same as a number but it is followed by the abbreviation for a unit. For example 2.5MB for 2.5 megabytes.

  Unit abbreviations include:

  KB: kilobytes
  MB: megabytes
  GB: gigabytes
  TB: terabytes
- **Text**: Text values are characters, strings, or other values that do not fit the definition for Number or Size value.
- **Version Number**: The expected format for version numbers is 0.0.0.0. For example:

```
<file version> "%WINDIR%\FILE.DLL" <= 3.1.0.0
```

A file version can also be specified in these formats:

1

1.0

1.0.0

Quotes may be used around the version number, if you want the number to be evaluated as a string value or if you wish to use the wild-card characters "?" or "*". Quotes and wild-card characters should only be used with the logical operators MATCHES or GREPMATCHES.

Example

In both of the following examples, the rule evaluates to TRUE if PTCLIENT.EXE is greater than or equal to version 5 and less than version 6.

```
(<file version> "c:\Program Files\Prism Deploy\ClientInstallFiles\ptclient.exe" >= 5.0) AND
(<file version> "c:\Program Files\Prism Deploy\ClientInstallFiles\ptclient.exe " < 6.0)
<file version> "c:\Program Files\Prism Deploy\ClientInstallFiles\ptclient.exe " MATCHES "5.*"
```

Predefined Variables

Predefined Variables

Predefined variables include those that are part of Prism Deploy like PT_CD, and those defined as part of the Prism file installation rules. For some predefined variables, the possible values are provided by Prism Deploy, not the environment—these values are called constants. For example, Prism Deploy includes a list of potential values for the CPUFamily variable—486, PentiumPro/II/III-Athlon, Pentium4, and so on.

Predefined variables are used without special punctuation or text. For example, a rule to evaluate the amount of RAM on the target computer:

```
RAM >= 128MB
```

If the name of a predefined variable is enclosed in quotes, it is interpreted as a string value—not considered to be a predefined variable. For example, "CPUMaker" is a string with the value CPUMaker. CPUMaker is a predefined variable containing the CPU type.

Computer Active Directory (AD) Name

```
ComputerADName [comparison operator] ["Domain/Organizational Unit/Name"]
```

Active Directory (AD) name of the computer. The name must be written in canonical format, for example:

```
ComputerADName = "NewBoundary.local/Computers/pc1"
```

Variable Type

String

Computer DNS Name

```
ComputerDNSName [comparison operator] ["DNS Name"]
```

DNS name of the computer

Variable Type

String

Computer Domain

```
ComputerDomain [comparison operator] ["Domain"]
```

The domain or workgroup of the computer.

Variable Type

String

Computer Name

```
ComputerName [comparison operator] ["Computer Name"]
```

Host name of the computer. This variable resolves to %computername%.

Variable Type

String

CPU Family

```
CPUFamily [comparison operator] [Constant]
```

Family of the computer’s primary processor.

Variable Type

String

Constants

- Pre486
- 486
- Pentium_K5/K6
- PentiumPro/II/III_Athlon

- Pentium4
- Other

CPU Maker

```
CPUMaker [comparison operator] [Constant]
```

Manufacturer of the computer’s primary processor.

Variable Type

String

Constants

- AMD
- Centaur
- Cyrix
- Intel
- Unknown

CPU Model

```
CPUModel [comparison operator] [Constant]
```

Model of the computer’s primary processor.

Variable Type

String

Constants

- AMD_AM486
- AMD_K5_Model_0
- AMD_K5_Model_1
- AMD_K5_Model_2
- AMD_K5_Model_3
- AMD_K6_Model_6
- AMD_K6_Model_7
- AMD_K6-2_Model_8
- AMD_K6-3_Model_9
- AMD_K6-3_Model_10
- AMD_K6-3_Model_11
- AMD_K6-3_Model_12
- AMD_K6-3_Model_13
- AMD_K6-3_Model_14
- AMD_K6-3_Model_15
- AMD_Other_K5/K6_CPU
- AMD_ATHLON
- AMD_Other_CPU
- Cyrix_CPU
- Centaur_CPU
- Intel_486DX_Model_0
- Intel_486DX_Model_1
- Intel_486SX_Model_2
- Intel_486DX2_Model_3
- Intel_486SL_Model_4
- Intel_486SX2_Model_5
- Intel_486DX2E_Model_7
- Intel_486DX4_Model_8
- Intel_Other_486_CPU
- Intel_Pentium_Model_1
- Intel_Pentium_Model_2
- Intel_Pentium_Model_3
- Intel_Pentium_MMX_Model_4
- Intel_Other_Pentium(P5)_CPU
- Intel_Pentium-Pro_Model_1
- Intel_Pentium-II_Model_3
- Intel_Pentium-II_Model_5
- Intel_Celeron_Model_6
- Intel_Pentium-III_Model_7
- Intel_Pentium-III_Model_8
- Intel_Pentium-III_Model_10
- Intel_Pentium-III_Mobile_Model_11
- Intel_Other_Pentium(P6)_CPU
- Intel_Pentium-4_Model_0
- Intel_Other_Pentium(P15)_CPU
- Intel_Other_CPU

First CD Letter

```
FirstCDLetter [comparison operator] [Constant]
```

Drive letter of the first CD-ROM drive on the computer.

Variable Type

String

Constants

A, B, C, ... Z

IP Address

```
IPAddress [comparison operator] ["Value"]
```

The IP address of the primary network card in the computer. If the target computer has multiple IP addresses, this value actually contains multiple IP addresses—so you can compare to any of its IP addresses.

Variable Type

String

NIC Address

```
NICAddress [comparison operator] ["Value"]
```

The NIC address of the primary network card in the computer. If the target computer has multiple network cards, this value actually contains multiple NIC addresses—so you can compare to any of its NIC addresses.

Variable Type

String

Number of CPUs

```
NumCPUs [comparison operator] [Number]
```

The number of CPUs installed and supported on the computer.

Variable Type

Number

Operating System Bits

```
OSBits [comparison operator] [Constant]
```

Indicates if the operating system is 32-bits or 64-bits.

Variable Type

String

Constants

- 32-Bit
- 64-Bit

Operating System Components

```
OSComponent [comparison operator] [Constant]
```

The extra components installed on the client operating system. For example, Windows XP Home Edition or Terminal Services.

Variable Type

String

Constants

- Windows Small Business Server
- Windows 2000 Advanced Server or Windows .NET Enterprise Server
- Microsoft Backoffice
- Terminal Services
- Microsoft Small Business Server (restrictive client license)
- Windows 2000 or Windows .NET Datacenter Server
- Non-application server mode running one interactive session
- Windows XP Home Edition
- Windows .NET Web Server
- Windows Storage Server 2003
- Windows Server 2003 Compute Cluster Edition
- Windows Home Server

Operating System Edition

```
OSEdition [comparison operator] [Constant]
```

The edition of the operating system.

Variable Type

String

Constants

- Edition Undefined
- Ultimate Edition
- Home Basic Edition
- Home Premium Edition
- Enterprise Edition
- Home Basic Edition N
- Business Edition
- Server Standard Edition (full installation)
- Server Datacenter Edition (full installation)
- Small Business Server
- Server Enterprise Edition (full installation)
- Starter Edition
- Server Datacenter Edition (core installation)
- Server Standard Edition (core installation)
- Server Enterprise Edition (core installation)
- Server Enterprise Edition for Itanium-based Systems
- Business Edition N
- Web Server Edition
- Cluster Server Edition
- Home Server Edition
- Storage Server Express Edition
- Storage Server Standard Edition

- Storage Server Workgroup Edition
- Storage Server Enterprise Edition
- Server for Small Business Edition
- Small Business Server Premium Edition
- Home Premium Edition N
- Enterprise Edition N
- Ultimate Edition N
- Windows Server 2003 R2
- Unlicensed

Operating System Family

OSFamily [comparison operator] [Constant]

The operating system family on the computer.

Variable Type

String

Constants

- Windows NT Family

Operating System Revision

OSRevision [comparison operator] [Constant]

The service-pack revision of the operating system.

Variable Type

String

Constants

- Windows 2000, Initial Release
- Windows 2000, Service Pack 1
- Windows 2000, Service Pack 2
- Windows 2000, Service Pack 3
- Windows 2000, Service Pack 4
- Windows Server 2000, Initial Release
- Windows Server 2000, Service Pack 1
- Windows Server 2000, Service Pack 2
- Windows Server 2000, Service Pack 3
- Windows Server 2000, Service Pack 4
- Windows XP, Initial Release
- Windows XP, Service Pack 1
- Windows XP, Service Pack 2
- Windows XP, Service Pack 3
- Windows XP x64, Initial Release
- Windows XP x64, Service Pack 1
- Windows XP x64, Service Pack 2
- Windows Server 2003, Initial Release
- Windows Server 2003, Service Pack 1
- Windows Server 2003, Service Pack 2
- Windows Vista, Initial Release
- Windows Vista, Service Pack 1
- Windows Vista, Service Pack 2
- Windows Server 2008, Initial Release

- Windows Server 2008, Service Pack 2
- Windows Server 2008 R2, Initial Release
- Windows 7, Initial Release

Operating System Role

```
OSRole [comparison operator] [Constant]
```

Role in which the computer’s operating system operates. For example, Workstation, Server, or Domain Controller.

Variable Type

String

Constants

- Domain Controller
- Workstation
- Server

Operating System Version

```
OSVersion [comparison operator] [Constant]
```

The specific operating system on the computer.

Variable Type

String

Constants

- Windows 2000
- Windows Server 2000
- Windows XP
- Windows XP x64
- Windows Server 2003
- Windows Vista
- Windows Server 2008
- Windows Server 2008 R2
- Windows 7
- Future Versions

RAM

```
RAM [comparison operator] [Number]
```

Amount of ram on the computer, in bytes

Variable Type

Size / Number

System Base Language

```
SystemBaseLanguage [comparison operator] [Constant]
```

The base language identifier (LANGID) for the managed computer. The base language identifiers, both SystemBaseLanguage and UserBaseLanguage, identify a family of languages—English, French, and so on.

Variable Type

String

Constants

Value*	**Descriptive Name****
Language.Base.Arabic	Arabic
Language.Base.Bulgarian	Bulgarian
Language.Base.Catalan	Catalan
Language.Base.Chinese	Chinese
Language.Base.Czech	Czech
Language.Base.Danish	Danish
Language.Base.German	German
Language.Base.Greek	Greek
Language.Base.English	English
Language.Base.Spanish	Spanish
Language.Base.Finnish	Finnish
Language.Base.French	French
Language.Base.Hebrew	Hebrew
Language.Base.Hungarian	Hungarian
Language.Base.Icelandic	Icelandic
Language.Base.Italian	Italian
Language.Base.Japanese	Japanese
Language.Base.Korean	Korean
Language.Base.Dutch	Dutch
Language.Base.Norwegian	Norwegian
Language.Base.Polish	Polish
Language.Base.Portuguese	Portuguese
Language.Base.Romanian	Romanian
Language.Base.Russian	Russian
Language.Base.Croatian	Croatian
Language.Base.Serbian	Serbian
Language.Base.Slovak	Slovak
Language.Base.Albanian	Albanian
Language.Base.Swedish	Swedish
Language.Base.Thai	Thai
Language.Base.Turkish	Turkish
Language.Base.Urdu	Urdu
Language.Base.Indonesian	Indonesian

Language.Base.Ukrainian	Ukrainian
Language.Base.Belarusian	Belarusian
Language.Base.Slovenian	Slovenian
Language.Base.Estonian	Estonian
Language.Base.Latvian	Latvian
Language.Base.Lithuanian	Lithuanian
Language.Base.Farsi	Farsi
Language.Base.Vietnamese	Vietnamese
Language.Base.Armenian	Armenian
Language.Base.Azeri	Azeri
Language.Base.Basque	Basque
Language.Base.FYROMacedonian	FYRO Macedonian
Language.Base.Afrikaans	Afrikaans
Language.Base.Georgian	Georgian
Language.Base.Faeroese	Faeroese
Language.Base.Hindi	Hindi
Language.Base.Malay	Malay
Language.Base.Kazak	Kazak
Language.Base.Kyrgyz	Kyrgyz
Language.Base.Swahili	Swahili
Language.Base.Uzbek	Uzbek
Language.Base.Tatar	Tatar
Language.Base.Punjabi	Punjabi
Language.Base.Gujarati	Gujarati
Language.Base.Tamil	Tamil
Language.Base.Telugu	Telugu
Language.Base.Kannada	Kannada
Language.Base.Marathi	Marathi
Language.Base.Sanskrit	Sanskrit
Language.Base.Mongolian	Mongolian
Language.Base.Galician	Galician
Language.Base.Konkani	Konkani
Language.Base.Syriac	Syriac
Language.Base.Divehi	Divehi
Language.Base.Catalan	Catalan
Language.Base.Chinese	Chinese
Language.Base.Czech	Czech

Language.Base.Danish	Danish
Language.Base.German	German
Language.Base.Greek	Greek
Language.Base.English	English
Language.Base.Spanish	Spanish
Language.Base.Finnish	Finnish
Language.Base.French	French
Language.Base.Hebrew	Hebrew
Language.Base.Hungarian	Hungarian
Language.Base.Icelandic	Icelandic
Language.Base.Italian	Italian
Language.Base.Japanese	Japanese
Language.Base.Korean	Korean
Language.Base.Dutch	Dutch
Language.Base.Norwegian	Norwegian
Language.Base.Polish	Polish
Language.Base.Portuguese	Portuguese
Language.Base.Romanian	Romanian
Language.Base.Russian	Russian
Language.Base.Croatian	Croatian
Language.Base.Serbian	Serbian
Language.Base.Slovak	Slovak
Language.Base.Albanian	Albanian
Language.Base.Swedish	Swedish
Language.Base.Thai	Thai
Language.Base.Turkish	Turkish
Language.Base.Urdu	Urdu
Language.Base.Indonesian	Indonesian
Language.Base.Ukrainian	Ukrainian
Language.Base.Belarusian	Belarusian
Language.Base.Slovenian	Slovenian
Language.Base.Estonian	Estonian
Language.Base.Latvian	Latvian
Language.Base.Lithuanian	Lithuanian
Language.Base.Farsi	Farsi
Language.Base.Vietnamese	Vietnamese
Language.Base.Armenian	Armenian

Language.Base.Azeri	Azeri
Language.Base.Basque	Basque
Language.Base.FYROMacedonian	FYRO Macedonian
Language.Base.Afrikaans	Afrikaans
Language.Base.Georgian	Georgian
Language.Base.Faeroese	Faeroese
Language.Base.Hindi	Hindi
Language.Base.Malay	Malay
Language.Base.Kazak	Kazak
Language.Base.Kyrgyz	Kyrgyz
Language.Base.Swahili	Swahili
Language.Base.Uzbek	Uzbek
Language.Base.Tatar	Tatar
Language.Base.Punjabi	Punjabi
Language.Base.Gujarati	Gujarati
Language.Base.Tamil	Tamil
Language.Base.Telugu	Telugu
Language.Base.Kannada	Kannada
Language.Base.Marathi	Marathi
Language.Base.Sanskrit	Sanskrit
Language.Base.Mongolian	Mongolian
Language.Base.Galician	Galician
Language.Base.Konkani	Konkani
Language.Base.Syriac	Syriac
Language.Base.Divehi	Divehi

* If you are typing a constant directly in the Rules Editor, use the constant as it is given in the first column.

** The descriptive name is displayed in the drop-down list in the Rules Editor.

System Full Language

```
SystemFullLanguage [comparison operator] [Constant]
```

The full language identifier (LANGID) for the managed computer. The full language identifiers, both SystemFullLanguage and UserFullLanguage, identify a specific language variant—English (United Kingdom), German (Germany), German (Austria), and so on.

Variable Type

String

Constants

Value*	**Descriptive Name****
Language.Full.Afrikaans	Afrikaans
Language.Full.Albanian	Albanian
Language.Full.Arabic.SaudiArabia	Arabic (Saudi Arabia)
Language.Full.Arabic.Iraq	Arabic (Iraq)
Language.Full.Arabic.Egypt	Arabic (Egypt)
Language.Full.Arabic.Libya	Arabic (Libya)
Language.Full.Arabic.Algeria	Arabic (Algeria)
Language.Full.Arabic.Morocco	Arabic (Morocco)
Language.Full.Arabic.Tunisia	Arabic (Tunisia)
Language.Full.Arabic.Oman	Arabic (Oman)
Language.Full.Arabic.Yemen	Arabic (Yemen)
Language.Full.Arabic.Syria	Arabic (Syria)
Language.Full.Arabic.Jordan	Arabic (Jordan)
Language.Full.Arabic.Lebanon	Arabic (Lebanon)
Language.Full.Arabic.Kuwait	Arabic (Kuwait)
Language.Full.Arabic.UAE	Arabic (U.A.E.)
Language.Full.Arabic.Bahrain	Arabic (Bahrain)
Language.Full.Arabic.Qatar	Arabic (Qatar)
Language.Full.Armenian	Armenian
Language.Full.Azeri.Latin	Azeri (Latin)
Language.Full.Azeri.Cyrillic	Azeri (Cyrillic)
Language.Full.Basque	Basque
Language.Full.Belarusian	Belarusian
Language.Full.Bulgarian	Bulgarian
Language.Full.Burmese	Burmese
Language.Full.Catalan	Catalan
Language.Full.Chinese.Taiwan	Chinese (Taiwan)
Language.Full.Chinese.PRC	Chinese (PRC)
Language.Full.Chinese.HongKong	Chinese (Hong Kong SAR, PRC)
Language.Full.Chinese.Singapore	Chinese (Singapore)
Language.Full.Chinese.Macau	Chinese (Macau SAR)
Language.Full.Croatian	Croatian
Language.Full.Czech	Czech
Language.Full.Danish	Danish
Language.Full.Divehi	Divehi

Language.Full.Dutch.Netherlands	Dutch (Netherlands)
Language.Full.Dutch.Belgium	Dutch (Belgium)
Language.Full.English.UnitedStates	English (United States)
Language.Full.English.UnitedKingdom	English (United Kingdom)
Language.Full.English.Australian	English (Australian)
Language.Full.English.Canadian	English (Canadian)
Language.Full.English.NewZealand	English (New Zealand)
Language.Full.English.Ireland	English (Ireland)
Language.Full.English.SouthAfrica	English (South Africa)
Language.Full.English.Jamaica	English (Jamaica)
Language.Full.English.Caribbean	English (Caribbean)
Language.Full.English.Belize	English (Belize)
Language.Full.English.Trinidad	English (Trinidad)
Language.Full.English.Zimbabwe	English (Zimbabwe)
Language.Full.English.Philippines	English (Philippines)
Language.Full.Estonian	Estonian
Language.Full.Faeroese	Faeroese
Language.Full.Farsi	Farsi
Language.Full.Finnish	Finnish
Language.Full.French.Standard	French (Standard)
Language.Full.French.Belgian	French (Belgian)
Language.Full.French.Canadian	French (Canadian)
Language.Full.French.Switzerland	French (Switzerland)
Language.Full.French.Luxembourg	French (Luxembourg)
Language.Full.French.Monaco	French (Monaco)
Language.Full.Galician	Galician
Language.Full.Georgian	Georgian
Language.Full.German.Standard	German (Germany)
Language.Full.German.Switzerland	German (Switzerland)
Language.Full.German.Austria	German (Austria)
Language.Full.German.Luxembourg	German (Luxembourg)
Language.Full.German.Liechtenstein	German (Liechtenstein)
Language.Full.Greek	Greek
Language.Full.Gujarati	Gujarati
Language.Full.Hebrew	Hebrew
Language.Full.Hindi	Hindi
Language.Full.Hungarian	Hungarian

Language.Full.Icelandic	Icelandic
Language.Full.Indonesian	Indonesian
Language.Full.Italian.Standard	Italian (Standard)
Language.Full.Italian.Switzerland	Italian (Switzerland)
Language.Full.Japanese	Japanese
Language.Full.Kannada	Kannada
Language.Full.Konkani	Konkani
Language.Full.Korean	Korean
Language.Full.Korean.Johab	Korean.Johab
Language.Full.Kyrgyz	Kyrgyz
Language.Full.Latvian	Latvian
Language.Full.Lithuanian	Lithuanian
Language.Full.Lithuanian.Classic	Lithuanian (Classic)
Language.Full.FYROMacedonian	FYRO Macedonian
Language.Full.Malay.Malaysian	Malay (Malaysian)
Language.Full.Malay.BruneiDarussalam	Malay (Brunei Darussalam)
Language.Full.Marathi	Marathi
Language.Full.Mongolian	Mongolian
Language.Full.Norwegian.Bokmal	Norwegian (Bokmal)
Language.Full.Norwegian.Nynorsk	Norwegian (Nynorsk)
Language.Full.Polish	Polish
Language.Full.Portuguese.Brazil	Portuguese (Brazil)
Language.Full.Portuguese.Portugal	Portuguese (Portugal)
Language.Full.Punjabi	Punjabi
Language.Full.Romanian	Romanian
Language.Full.Russian	Russian
Language.Full.Sanskrit	Sanskrit
Language.Full.Serbian.Cyrillic	Serbian (Cyrillic)
Language.Full.Serbian.Latin	Serbian (Latin)
Language.Full.Slovak	Slovak
Language.Full.Slovenian	Slovenian
Language.Full.Spanish.Spain.TraditionalSort	Spanish (Spain, Traditional Sort)
Language.Full.Spanish.Mexican	Spanish (Mexican)
Language.Full.Spanish.Spain.ModernSort	Spanish (Spain, Modern Sort)
Language.Full.Spanish.Guatemala	Spanish (Guatemala)
Language.Full.Spanish.CostaRica	Spanish (Costa Rica)
Language.Full.Spanish.Panama	Spanish (Panama)

Language.Full.Spanish.DominicanRepublic	Spanish (Dominican Republic)
Language.Full.Spanish.Venezuela	Spanish (Venezuela)
Language.Full.Spanish.Colombia	Spanish (Colombia)
Language.Full.Spanish.Peru	Spanish (Peru)
Language.Full.Spanish.Argentina	Spanish (Argentina)
Language.Full.Spanish.Ecuador	Spanish (Ecuador)
Language.Full.Spanish.Chile	Spanish (Chile)
Language.Full.Spanish.Uruguay	Spanish (Uruguay)
Language.Full.Spanish.Paraguay	Spanish (Paraguay)
Language.Full.Spanish.Bolivia	Spanish (Bolivia)
Language.Full.Spanish.ElSalvador	Spanish (El Salvador)
Language.Full.Spanish.Honduras	Spanish (Honduras)
Language.Full.Spanish.Nicaragua	Spanish (Nicaragua)
Language.Full.Spanish.PuertoRico	Spanish (Puerto Rico)
Language.Full.Sutu	Sutu
Language.Full.Swahili.Kenya	Swahili (Kenya)
Language.Full.Swedish	Swedish
Language.Full.Swedish.Finland	Swedish (Finland)
Language.Full.Syriac	Syriac
Language.Full.Tamil	Tamil
Language.Full.Tatar.Tatarstan	Tatar (Tatarstan)
Language.Full.Telugu	Telugu
Language.Full.Thai	Thai
Language.Full.Turkish	Turkish
Language.Full.Ukrainian	Ukrainian
Language.Full.Urdu.Pakistan	Urdu (Pakistan)
Language.Full.Urdu.India	Urdu (India)
Language.Full.Uzbek.Latin	Uzbek (Latin)
Language.Full.Uzbek.Cyrillic	Uzbek (Cyrillic)
Language.Full.Vietnamese	Vietnamese

* If you are typing a constant directly in the Rules Editor, use the constant as it is given in the first column.

** The descriptive name is displayed in the drop-down list in the Rules Editor.

Time Zone

TimeZone [comparison operator] [Constant]

The Current Time Zone specified in the computer's date and time properties. For example Central Standard Time.

Variable Type

String

Constants

Value*	**Descriptive Name****
TimeZone.DatelineStandardTime	(GMT-12:00) International Date Line West
TimeZone.SamoaStandardTime	(GMT-11:00) Midway Island, Samoa
TimeZone.HawaiianStandardTime	(GMT-10:00) Hawaii
TimeZone.AlaskanStandardTime	(GMT-09:00) Alaska
TimeZone.PacificStandardTime	(GMT-08:00) Pacific Time (US & Canada); Tijuana
TimeZone.MountainStandardTime	(GMT-07:00) Mountain Time (US & Canada)
TimeZone.MexicoStandardTime2	(GMT-07:00) Chihuahua, La Paz, Mazatlan
TimeZone.USMountainStandardTime	(GMT-07:00) Arizona
TimeZone.CentralStandardTime	(GMT-06:00) CentralTime (US & Canada)
TimeZone.CanadaCentralStandardTime	(GMT-06:00) Saskatchewan
TimeZone.MexicoStandardTime	(GMT-06:00) Guadalajara, Mexico City, Monterrey
TimeZone.CentralAmericaStandardTime	(GMT-06:00) CentralAmerica
TimeZone.EasternStandardTime	(GMT-05:00) Eastern Time (US & Canada)
TimeZone.USEasternStandardTime	(GMT-05:00) Indiana (East)
TimeZone.SAPacificStandardTime	(GMT-05:00) Bogota, Lima, Quito
TimeZone.AtlanticStandardTime	(GMT-04:00) Atlantic Time (Canada)
TimeZone.PacificSAStandardTime	(GMT-04:00) Santiago
TimeZone.SAWesternStandardTime	(GMT-04:00) Caracas, La Paz
TimeZone.NewfoundlandStandardTime	(GMT-03:30) Newfoundland
TimeZone.GreenlandStandardTime	(GMT-03:00) Greenland
TimeZone.SAEasternStandardTime	(GMT-03:00) Buenos Aires, Georgetown
TimeZone.E.SouthAmericaStandardTime	(GMT-03:00) Brasilia
TimeZone.Mid-AtlanticStandardTime	(GMT-02:00) Mid-Atlantic
TimeZone.CapeVerdeStandardTime	(GMT-01:00) Cape Verde Is.
TimeZone.AzoresStandardTime	(GMT-01:00) Azores
TimeZone.GMTStandardTime	(GMT) Greenwich Mean Time : Dublin, Edinburgh, Lisbon, London

TimeZone.GreenwichStandardTime	(GMT) Casablanca, Monrovia
TimeZone.RomanceStandardTime	(GMT+01:00) Brussels, Copenhagen, Madrid, Paris
TimeZone.W.EuropeStandardTime	(GMT+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
TimeZone.CentralEuropeStandardTime	(GMT+01:00) Belgrade, Bratislava, Budapest, Ljubljana,Prague
TimeZone.CentralEuropeanStandardTime	(GMT+01:00) Sarajevo, Skopje, Warsaw, Zagreb
TimeZone.W.CentralAfricaStandardTime	(GMT+01:00) West Central Africa
TimeZone.GTBStandardTime	(GMT+02:00) Athens, Istanbul, Minsk
TimeZone.SouthAfricaStandardTime	(GMT+02:00) Harare, Pretoria
TimeZone.E.EuropeStandardTime	(GMT+02:00) Bucharest
TimeZone.EgyptStandardTime	(GMT+02:00) Cairo
TimeZone.FLEStandardTime	(GMT+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius
TimeZone.IsraelStandardTime	(GMT+02:00) Jerusalem
TimeZone.ArabicStandardTime	(GMT+03:00) Baghdad
TimeZone.ArabStandardTime	(GMT+03:00) Kuwait, Riyadh
TimeZone.RussianStandardTime	(GMT+03:00) Moscow, St. Petersburg, Volgograd
TimeZone.E.AfricaStandardTime	(GMT+03:00) Nairobi
TimeZone.IranStandardTime	(GMT+03:30) Tehran
TimeZone.ArabianStandardTime	(GMT+04:00) Abu Dhabi, Muscat
TimeZone.CaucasusStandardTime	(GMT+04:00) Baku, Tbilisi, Yerevan
TimeZone.AfghanistanStandardTime	(GMT+04:30) Kabul
TimeZone.EkaterinburgStandardTime	(GMT+05:00) Ekaterinburg
TimeZone.WestAsiaStandardTime	(GMT+05:00) Islamabad, Karachi, Tashkent
TimeZone.IndiaStandardTime	(GMT+05:30) Chennai, Kolkata, Mumbai, New Delhi
TimeZone.NepalStandardTime	(GMT+05:45) Kathmandu
TimeZone.N.CentralAsiaStandardTime	(GMT+06:00) Almaty, Novosibirsk
TimeZone.CentralAsiaStandardTime	(GMT+06:00) Astana, Dhaka
TimeZone.SriLankaStandardTime	(GMT+06:00) Sri Jayawardenepura
TimeZone.MyanmarStandardTime	(GMT+06:30) Rangoon
TimeZone.SEAsiaStandardTime	(GMT+07:00) Bangkok, Hanoi, Jakarta

TimeZone.NorthAsiaStandardTime	(GMT+07:00) Krasnoyarsk
TimeZone.ChinaStandardTime	(GMT+08:00) Beijing, Chongqing, Hong Kong, Urumqi
TimeZone.NorthAsiaEastStandardTime	(GMT+08:00) Irkutsk, Ulaan Bataar
TimeZone.SingaporeStandardTime	(GMT+08:00) Kuala Lumpur, Singapore
TimeZone.W.AustraliaStandardTime	(GMT+08:00) Perth
TimeZone.TaipeiStandardTime	(GMT+08:00) Taipei
TimeZone.TokyoStandardTime	(GMT+09:00) Osaka, Sapporo, Tokyo
TimeZone.KoreaStandardTime	(GMT+09:00) Seoul
TimeZone.YakutskStandardTime	(GMT+09:00) Yakutsk
TimeZone.Cen.AustraliaStandardTime	(GMT+09:30) Adelaide
TimeZone.AUSCentralStandardTime	(GMT+09:30) Darwin
TimeZone.AUSEasternStandardTime	(GMT+10:00) Canberra, Melbourne, Sydney
TimeZone.E.AustraliaStandardTime	(GMT+10:00) Brisbane
TimeZone.WestPacificStandardTime	(GMT+10:00) Guam, Port Moresby
TimeZone.TasmaniaStandardTime	(GMT+10:00) Hobart
TimeZone.VladivostokStandardTime	(GMT+10:00) Vladivostok
TimeZone.CentralPacificStandardTime	(GMT+11:00) Magadan, Solomon Is., New Caledonia
TimeZone.NewZealandStandardTime	(GMT+12:00) Auckland, Wellington
TimeZone.FijiStandardTime	(GMT+12:00) Fiji, Kamchatka, Marshall Is.
TimeZone.TongaStandardTime	(GMT+13:00) Nuku'alofa

* If you are typing a constant directly in the Rules Editor, use the constant as it is given in the first column.

** The descriptive name is displayed in the drop-down list in the Rules Editor.

User Active Directory (AD) Name

```
UserADName [comparison operator] ["Domain/Organizational Unit/user name"]
```

Active Directory (AD) name of the current user. The name must be written in canonical format, for example:

```
UserADName = "NewBoundary.local/marketing/val"
```

Variable Type

String

User Base Language

```
UserBaseLanguage [comparison operator] [Constant]
```

The base language identifier (LANGID) for the current user.

Variable Type

String

Constants

See the list of constants for System Base Language

User Domain

```
UserDomain [comparison operator] ["Value"]
```

The domain or workgroup of the current user.

Variable Type

String

User Full Language

```
UserFullLanguage [comparison operator] [Constant"]
```

The full language identifier (LANGID) for the current user.

Variable Type

String

Constants

See the list of constants for System Full Language

User Name

```
UserName [comparison operator] ["Name"]
```

The name of the user currently logged on to the computer. If no one is logged on, Username is <unattended>. For example, create a new user-defined group and type UserName = "Julie", then assign a Task to this subgroup. The subgroup is populated with a new member each time Julie logs on to a computer that is on the network. From one Task assignment, Julie receives the Task on each computer she logs on to, without taking time to identify each computer and assigning the Task to each of these computers.

This variable resolves to %username%.

Variable Type

String

User-Defined Variables

User-Defined Variables

The user-defined variables are enclosed in angle brackets (< >) in the **Value type** field of the Rule Expert. These values can be used with any appropriate value. Unlike the predefined variables, you are not limited to the values defined by Prism Deploy.

Custom

Environment Variable

File

File Version

Folder

Free Space

INI Value

Registry Key

Registry Value

Total Space on Path

WMI

Custom

```
<Custom> ["ProgID"] ["Parameter string"] [comparison operator] ["Value"]
```

String value supplied by a custom plug-in. To indicate an empty parameter string use empty quotes (""). For more detailed information, see **www.newboundary.com/support/support.htm**.

Variable Type

String

Environment Variable

```
<Env Var> [%variable name%] [comparison operator] [Value]
```

Identifies a specific environment variable in the user's session. Environment variable items must specify an environment variable, a comparison operator, and a value. For example:

```
<Env Var> "%SESSIONNAME%" = "Console"
```

Variable Type

String

File Version

```
<File Version> [File Path] [comparison operator] [Version_number]
```

The version of a file on the target computer, for example:

```
<File Version> "%WINDIR%\MS.DLL" >= 6.1.0.0
```

The "File Version" item type is a version number, and is considered "0.0.0.0" if the given file does not exist, or if it has no version information. (See Values.)

Variable Type

Version

File

```
[logical operator] <File> ["File Path"]
```

The name of a file on the target computer, for example:

```
NOT EXISTS <File> "C:\DATA\STUFF.DAT"
```

Variable Type

String

Folder

```
[logical operator] <Folder> ["Folder Path"]
```

The path and folder name for a folder on the target computer, for example:

```
EXISTS <Folder> "C:\Program Files\Acme Software\Acme App"
```

Variable Type

String

Free Space

```
<FreeSpace> ["File Path"] [comparison operator] ["Number of TB/GB/MB/KB"]
```

The amount of free space available on the specified drive. Specify the unit of measure.

For example:

```
<FreeSpace> "C:\" > 500MB
```

Variable Type

Size/Number

INI Value

```
<INI Value> ["Filepath\\section\\key"] [comparison operator] [Value]
```

INI Value items must have three double-backslash-separated items: the file path, the section name, and the key name. For example:

```
<INI Value> "%WINDIR%\Acme.ini\\Audio Section\\Sample Rate" = "InetProtocol file"
```

Variable Type

String

Registry Key

```
[logical operator] <Registry Key> ["Registry Key Path"]
```

Identifies a specific registry key on the target computer. For example:

```
EXISTS <Registry key> "HKEY_CLASSES_ROOT\exefile"
```

Variable Type

String

Registry Value

```
<Registry Value> ["Registry Key Path\\Value"] [comparison operator] [Value]
```

Identifies a specific registry value on the target computer. Two items, one for the key and one for the value name, must be provided and separated by two backslashes (both parts within the same set of quotes, not separately quoted). For example:

```
<Registry Value>
"HKEY_LOCAL_MACHINE\SOFTWARE\Adobe\Acrobat\Reader\5.0\AdobeViewer\\Trusted
Mode" = 2
```

To query default registry values when building rules based on <Registry Value>, use this syntax:

```
<Registry Value> "HKLM\Software\\" = <value>
```

Variable Type

String

Total Space on Path

```
<TotalSpace> ["File Path"] [comparison operator] ["Number of TB/GB/MB/KB"]
```

Total disk space on the drive where "path" is located. Specify the unit of measure.

```
<TotalSpace> "C:\" > 500MB
```

Variable Type

Size/Number

WMI

```
<WMI> ["WMI Class"] [comparison operator] [Value]
```

WMI items must specify a WMI class, a comparison operator, and a value. For example:

```
<WMI> "Win32_OSRecoveryConfiguration.WriteToSystemLog" = "True"
```

Variable Type

Depends on the WMI variable.

Rules.INI File

Overview of the Rules.INI File

Windows operating systems have been around for a long time, and New Boundary Technologies has been designing, building, and supporting software-distribution applications since the early Windows days. This means we know all the tricks pertaining to supporting Windows.

Much of Prism's "smart" behavior is defined by its rules file, RULES.INI. This file defines how Prism handles special cases in the following categories when building distribution Packages:

- Registry
- Special text files such as AUTOEXEC.BAT and CONFIG.SYS
- INI files
- Files and folders

RULES.INI is available for you to view, edit, and customize to your environment or specific applications.

Note: If you modify the RULES.INI file and you have the Console and Editor installed on separate computers, copy the modified RULES.INI file to the other computer. For example, if you modify the RULES.INI file on the Console machine, copy it to the Editor machine.

Prism and File Updates

When you install a Prism file on a computer, Prism handles changes to INI files, the registry, and special text files in one of two ways:

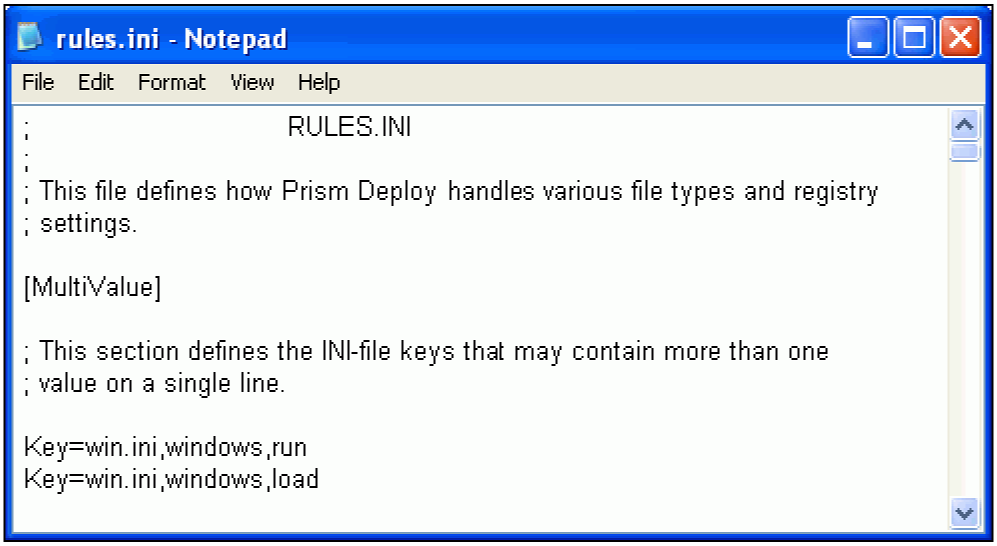
- **Add**: This action is performed when you install a Package or Prism Executable Package on a computer. If the Prism file contains an INI file, a registry setting, or fragments of a special text file, the lines of the file on the computer are compared with the lines of the file of the same name in the Prism file. Lines in the computer's file are only replaced if the same lines exist in the Prism file. That is, information not contained in the Prism file is left untouched on the computer. The purpose of rules.ini is to handle cases where Prism's default replace or merge behavior is not appropriate.
- **Replace**: This action is performed when you install a Picture on a computer (which is rarely done). A Picture replaces files with the same name on the computer. Files on a computer that do not have a counterpart in the Prism file are unaffected. Shortcuts and folders on the desktop or in the Start menu, and Application Data that are not in the Picture will be removed.

  The exception to this is the registry. Registry keys that are in both the Picture and in the computer's current registry will contain the Picture's value after the file is installed on the computer. Registry keys that are only in the Picture will be added to the computer's registry. Registry keys that are only in the computer's current registry will not be affected.

RULES.INI

The RULES.INI file is stored in the Prism folder. To view or edit RULES.INI, double-click on the file from Windows Explorer to open it in your default text editor, usually Notepad.

Note: If you edit RULES.INI, the changes do not take effect until you exit from and restart Prism.



```
;                        RULES.INI
;
; This file defines how Prism Deploy handles various file types and registry
; settings.

[MultiValue]

; This section defines the INI-file keys that may contain more than one
; value on a single line.

Key=win.ini,windows,run
Key=win.ini,windows,load
```

Like many sections in RULES.INI, the lines shown in the [MultiValue] section of the RULES.INI file conform to the following convention: Each line starts with Key=, which is followed by a file name, section name, and line name, separated by commas.

Any RULES.INI option that is discussed here can be updated in a text editor. The following discussion outlines some of the sections in RULES.INI along with Prism's default settings. RULES.INI supports the use of wildcard characters * and ?. Use * in place of a line name or section name. Use ? in place of a single character.

INI Files

INI files are standard Windows files that are used to control Windows settings and a user's working environment for specific applications. INI files are basic text files with a standard format:

```
[section]
key=value
```

Sections are set in square brackets [ ]. Keys contain letters and/or numbers and must be followed by an equal sign (no spaces). Values follow the equal sign (again, no spaces) and can be an integer, string, text, file name, path, or empty. Each key in a section is unique and has only one value.

Prism's default behavior is to merge new sections or keys contained in the Prism file into the target's INI file. If a key's value already exists, it is replaced by the value in the Prism file.

MultiValue Keys

An exception to the standard INI file format is in the common Windows file WIN.INI in the [WINDOWS] section. The lines run= and load= can both have multiple values separated by a space or a comma. Prism knows that, when installing a Prism file, the values following run= and load= in the WINDOWS section of WIN.INI can contain more than one value. New values should be added to the end of existing lines rather than replacing them. For example, when the line "run=prog3.exe" is added to a WIN.INI file, Prism changes "run=prog1.exe prog2.exe" to "run=prog1.exe prog2.exe prog3.exe".

MultiKeys

Another exception to the standard INI file format is found in the common Windows file SYSTEM.INI in the [386Enh] section. This section contains multiple keys called device=. Prism ignores existing keys of the same name when installing a Prism file. In this case, installing a Prism file with the line "device=vtcpip" to the 386Enh section of a SYSTEM.INI file causes the line to be appended to the end of the section and does not replace any existing lines unless a "device=vtcpip" line already exists.

File Definitions

Most INI files actually end with the extension .INI. However, there is at least one file with this extension that does not follow standard INI format. In fact, it is a binary file. The file is WINWORD.INI, which is an Office file. If a Prism file contains this file, it will not be editable within the Editor. Likewise, during a distribution, the Prism file will Copy rather than Merge the file.

The Registry

Like INI files, the registry contains Windows settings and controls a user's environment for specific applications. Windows 2000/XP or higher computers store the registry in the binary files SYSTEM.DAT and NTUSER.DAT. Backups of these files are stored with the extension .DA0.

Ignore Registry

Prism automatically ignores the specified registry keys when building Prism files. These entries are:

```
Software\LANovation\PictureTaker
Software\LANovation\PictureTaker Express
Software\LANovation\PictureTaker Personal Edition
Software\Lanovation\Prism
Software\Lanovation\Prism Pack
Software\LANovation\PTCC
Explorer\RecentDocs
Explorer\StreamMRU
Explorer\Streams
Explorer\DesktopStreamMRU
Explorer\DesktopStreams
UuidPersistentData
HKEY_LOCAL_MACHINE\System\ControlSet???
HKEY_LOCAL_MACHINE\System\Clone
HKEY_LOCAL_MACHINE\Clone
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager,PendingFileRenameOperations
HKEY_LOCAL_MACHINE\SAM
HKEY_LOCAL_MACHINE\SECURITY
HKEY_LOCAL_MACHINE\Software\Microsoft\Cryptography\RNG
HKEY_LOCAL_MACHINE\Hardware\ResourceMap\PNP Manager
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Enum\*\*\*\Control,DeviceReference
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer
Software\Lanovation\Installed Products
HKEY_LOCAL_MACHINE\Software\Microsoft\RPC,UuidSequenceNumber
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Prefetcher,ExitTime
```

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Prefetcher,StartTime

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\DHCP

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\*\Parameters\TCPIP

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TrkWks\Parameters

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Perflib

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Diagnostics\Performance

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Search

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Component Based Servicing

HKEY_LOCAL_MACHINE\BCD00000000

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\SideBySide

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Reliability

HKEY_LOCAL_MACHINE\COMPONENTS

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PrismXL\Data

HKEY_CURRENT_USER\Software\Microsoft\Windows\Shell

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam

HKEY_CURRENT_USER\Volatile Environment

Note that Prism ignores any registry key that *ends* with the specified key. These are registry branches that you do not want to distribute to other computers. The above entries contain very specific information about the current computer or information about Prism and other New Boundary Technologies products.

The following ignored registry key ignores a specific registry value (PendingFileRenameOperations), rather than ignoring the entire key.

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager,PendingFileRenameOperations

Registry Counters

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\SharedDLLs

Since many applications share common DLLs and drivers, this area of the registry is used to keep track of which files are shared and how many applications depend on each file. The registry key contains a registry value for each registered system file. Each file keeps a counter (a number) for the applications that depend on it. When an application installs a new DLL or driver, it often records the file in this area of the registry. If a program attempts to install a file that already exists and is registered, it increments the counter in the registry. Likewise, if an application is uninstalled, it decrements this counter.

If Prism finds that an application has incremented a number for a registered file, it will pull that file into the Prism files, even if the file was on your computer prior to installation of the application. Prism will also increment or decrement the counter as appropriate for each computer when a Prism file is installed or rolled back. If an application install or uninstall decrements the counter to zero (0), the resulting system file will be deleted from the target computer.

Merge Registry Values

This section defines which registry values are to be treated as a list of discrete pieces of text Prism merges new entries with existing values rather than replacing them.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment,Os2LibPath

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment,Path
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment,PATHEXT
HKEY_CURRENT_USER\Environment,Os2LibPath
HKEY_CURRENT_USER\Environment,Path
HKEY_CURRENT_USER\Environment,PATHEXT
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application,Sources
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\VirtualDeviceDrivers,VDD
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\ServiceGroupOrder,List
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\NetworkCards\*\NetRules,Block
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\*\Linkage,Bind
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\*\Linkage,Export
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\*\Linkage,Route
HKEY_LOCAL_MACHINE\Software\Microsoft\Java VM,ClassPath
HKEY_LOCAL_MACHINE\Software\Microsoft\Java VM,TrustedClassPath
HKEY_LOCAL_MACHINE\Software\Microsoft\Java VM,TrustedLibsDirectory
HKEY_LOCAL_MACHINE\Software\Microsoft\Java VM,LibsDirectory
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security,Sources
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System,Sources
```

Files and Folders

Prism maintains a great deal of control over which files are stored in Prism files. The rules described below only pertain to the initial step in the Expert when scanning the current state of your system. These rules are not used when distributing Prism files to target computers. If you add any of the mentioned files to a Prism file, the file will be distributed.

Exclude Files

```
*.SFTCH?
386SPART.PAR
AVCONSOL.INI
CLASSES.DAT
METABASE.BIN
NTUSER.DA?
NTUSER.DAT.LOG
PAGEFILE.SYS
PICTAKER.LOG
SCH2DLGU.TXT
SHELLICONCACHE
SYSTEM.DA?
SYSTEM.DAT.LOG
USER.DA?
```

WIN386.SWP

WININIT.INI

SCHEDLGU.TXT

hiberfil.sys

wpa.dbl

USRCLASS.DAT

USRCLASS.DAT.LOG

The files listed above are those files that Prism ignores when building a Prism file. These files are ignored wherever they are located on the computer. If any of the above files are manually copied into a Prism file, the file will be distributed to the destination computer.

In addition to this list, files that are flagged "temporary" are automatically ignored.

Exclude Files in Certain Folders

Prism will automatically exclude all files found in the following folders. You rarely would want to find the changes made to your temporary files or the binary files Windows uses for the registry.

%RECENT%

%TEMP%

%WINSYSDIR%\CONFIG

%WINSYSDIR%\NtmsData

TEMPORARY INTERNET FILES

\RECYCLED

\RECYCLER

_RESTORE

%WINDIR%\Security

%WINDIR%\Debug

%WINDIR%\Prefetch

\System Volume Information

%pt_datadir%

%pt_defdatadir%

%pt_uninstalldir%

%pt_defuninstalldir%

%COOKIES%

%HISTORY%

%WINDIR%\CSC

%WINDIR%\TASKS\SA.DAT

%WINSYSDIR%\WBEM\Repository

%WINSYSDIR%\WBEM\Logs

%WINSYSDIR%\CATROOT2

%WINDIR%\SoftwareDistribution\DataStore\Logs

%WINDIR%\RESCACHE

\Boot

\$Recycle.Bin

%Local AppData%\Microsoft\Feeds Cache

%WINDIR%\WINSXS\Backup

%WINDIR%\WINSXS\Catalogs

%WINDIR%\WINSXS\FileMaps

%WINDIR%\WINSXS\InstallTemp

Busy Files

If a Prism file attempts to replace a file that is busy, Prism places the file in a temporary location and uses Windows' method of updating the busy file during the next boot process. Windows 2000/XP or higher add an entry to the HKEY_LOCAL_MACHINE branch of the registry called PendingFileRenameOperations, notifying the system of the update.

Prism does not automatically force a reboot if it encounters a busy file. If an install requires a reboot (a good hint is that the initial install required a reboot before completion), we recommend setting a property on the Prism file to inform the user that a reboot is necessary to finish the installation process. (See the File Properties tabs in the Editor for information on setting properties on Prism files.)

Text Files: AUTOEXEC.BAT and CONFIG.SYS

Prism interprets and parses AUTOEXEC.BAT and CONFIG.SYS when building and distributing Prism files. The changes are stored as partial files containing special instructions for adding the partial text file to an existing text file of the same name.

When Prism finds changes to text files other than AUTOEXEC.BAT and CONFIG.SYS, Prism includes the entire text file in the resulting Prism file. This means that existing text files of the same name are *copied* to the target system and overwritten depending on the Replace options set for the Prism file.

Sections in Text Files

Prism finds changes and merges those changes into existing text files whether or not the files have sections. As with INI files, Prism keeps separate all text that is prefaced by a word or phrase in square brackets. Prism treats the block of text as a section. For example, you may use sections in CONFIG.SYS referring to the DOS Startup menu.

How Prism Knows Where to Put Information in an Existing Text File

When Prism finds a change in a text file, it notes the placement of the line. In order to "remember" the placement, Prism includes one of several commands listed below in the text file it creates. When the Prism file is installed, Prism interprets the command to place text file information where it should be in the current file.

The following section describes the commands Prism uses. You will need these commands only if you decide to modify Prism's suggestions or create a customized Prism file that includes changes to non-INI text files.

Important: Because different computers may have different entries in CONFIG.SYS and AUTOEXEC.BAT, the commands that Prism suggests for the computer where you created the Prism file may not be correct for a target computer. For this reason, we suggest that you review the contents of text files that are included in a Prism file. You may need to edit the commands so that the text file changes will be installed correctly to multiple computers.

All of the Prism text file commands start with **Merge Command** followed by a specific command. The text on the line that follows this command is the actual line to be inserted or updated in the text file. The text may either be a single line or a block of lines. A block is a sequence of lines that are not interrupted by any of the Prism commands listed below.

Prism treats a block of lines as a single entry. If the block does not exist in the destination text file *exactly* as it appears in the Prism file, the block will be prepended to the text file. For example:

Adding:

Merge Command: Prepend to Section
Line1
Line2

To a line with:

Line2 (or just Line1, or just Line2)
Line1

Will result in:

Line1
Line2
Line2
Line1

If this is not the desired behavior, break the block into single lines within the Prism file, such as:

Merge Command: Prepend to Section
Line1
Merge Command: Prepend to Section
Line2

Note: Prism finds new, modified, or deleted blank lines in text files. The text file in a Prism file may include merge commands for inserting or deleting blank lines. Blank lines may also be considered part of a block of lines.

Merge Command: Merge text

This line will always be the first Prism command in a text file within a Prism file. It is usually followed by one or more of the commands listed below along with the actual text file line that is to be merged or deleted.

Merge Text tells Prism that the file is to be merged into the existing file following the rules described below. If this line is not the first line in a file, Prism overwrites the existing text file with the file in the Prism file.

Merge Command: Prepend to section

This command tells Prism to add the following line or block to the beginning of the section (if any) or file. The line or block will not be added if it already exists in the destination file. Adding lines to the top of a section or file is Prism’s default behavior.

Merge Command: Append to section

This command tells Prism to add the following line or block to the end of the section or file. The line or block will not be added if it already exists in the destination file.

Merge Command: Append after line x

This command tells Prism to add the following line or block after the line "x." If the line "x" does not exist, the line or block will not be added. If the line defined by "x" ends in "…", Prism will find any line that starts with "x." For example, a file to be merged into AUTOEXEC.BAT might contain:

```
Merge Command: Append after line PATH...
C:\DOS\DOSKEY
```

When a Prism file containing this command is executed, it will add the line C:\DOS\DOSKEY after the existing path statement no matter what follows the word "PATH." Prism will not add the line or block if it already exists.

Note Prism does not automatically insert "…" into a text file. Prism will always suggest the entire line that was modified or deleted in the script command. For example:

```
Merge Command: Append after line Path=c:\windows;c:\windows\system.
```

If you want to append the following line or block after the path statement, no matter what the path statement may be, you must edit the contents of the text file to replace the suggested line with "…". For example:

```
Merge Command: Append after line Path=…
```

Merge Command: If possible append after line x

This command is the same as Merge Command: Append after line x except that Prism will add the following line or block to the end of the section or file if the line "x" is not present.

Merge Command: Prepend to line x

This command prepends the following line to the current line "x." If the line "x" does not exist, the line or block will not be added. If the line defined by "x" ends in "…", Prism will find any line that starts with "x." For example, a file to be merged into AUTOEXEC.BAT might contain:

```
Merge Command: Prepend to line PATH=...
Path=C:\windows\system
```

The Prism file will check for the existence of C:\WINDOWS\SYSTEM in the current path statement. If it exists, the line will not be prepended. If C:\WINDOWS\SYSTEM does not exist in the current path statement, it will be added to the beginning of the current path statement. Prism inserts a semicolon (;) after the new insertion.

Merge Command: If possible prepend to line x

This command is the same as Merge Command: Prepend to line x, except that the line following it will be added to the destination file whether or not "x" exists.

Merge Command: Append to line x

This command appends the following line to the current line "x." If the line "x" does not exist, the line or block will not be added. If the line defined by "x" ends in "…", Prism will find any line that starts with "x." For example, a file to be merged into AUTOEXEC.BAT might contain:

```
Merge Command: Append to line PATH=...
Path=c:\windows\system
```

The Prism file will check for the existence of C:\WINDOWS\SYSTEM in the current path statement. If it exists, the line will not be appended. If C:\WINDOWS\SYSTEM does not exist in the current path statement, it will be appended to the end of the current path statement. Prism inserts a semicolon (;) before the new insertion.

Merge Command: If possible append to line x

This command is the same as Merge Command: Append to line x, except that the line following it will be added to the destination whether or not "x" exists.

Merge Command: Delete line x

Prism deletes the line "x" from the destination file, if the line exists. When "x" ends in "…", Prism deletes the first line it encounters that starts with "x."

Merge Command: Replace line x

If the line "x" exists, Prism replaces it with the line that follows. If "x" ends in "…",Prism replaces the first line it encounters that starts with "x."

Merge Command: If possible replace line x

This command is the same as Merge Command: Replace line x, except that the line following it will be added to the destination file whether or not "x" exists.

Examples of Text File Commands

```
CONFIG.SYS
  Merge Command: Merge Text
  Merge Command: Replace line c:\windows\emm386.exe...
  C:\windows\emm386.exe /4096 RAM
  Merge Command: If possible replace line Files=40
  Files=80

AUTOEXEC.BAT
  Merge Command: Merge Text
  Merge Command: Append to line Path=...
  Path=C:\program files\newdir
  Merge Command: Delete line Set Display=VGA
  Merge Command: Append to section
  Set Display=SVGA
  Merge Command: Append to section
  C:\WINDOWS\COMMAND\DOSKEY
```

Index