

Application Management



The following sections provide information on the Application Management features in Novell® ZENworks® Desktop Management and the tasks you might need to perform to manage your applications:

- ◆ Chapter 19, “Novell Application Launcher: Components Overview,” on page 239
- ◆ Chapter 20, “Novell Application Launcher: Installing and Starting,” on page 249
- ◆ Chapter 21, “Novell Application Launcher: Configuring Settings,” on page 265
- ◆ Chapter 22, “Novell Application Launcher: Customizing Views,” on page 281
- ◆ Chapter 23, “Novell Application Launcher: Managing Authentication and File System Access,” on page 293
- ◆ Chapter 24, “Novell Application Launcher: Managing the Cache,” on page 303
- ◆ Chapter 25, “Novell Application Launcher: Organizing Applications,” on page 313
- ◆ Chapter 26, “ZENworks Launch Gadget: Configuring Settings,” on page 319
- ◆ Chapter 27, “Distribution: Overview,” on page 323
- ◆ Chapter 28, “Distribution: Simple Applications,” on page 325
- ◆ Chapter 29, “Distribution: Complex Applications,” on page 331
- ◆ Chapter 30, “Distribution: Terminal Server Applications,” on page 341
- ◆ Chapter 31, “Distribution: Web Applications,” on page 347
- ◆ Chapter 32, “Distribution: Applications to Terminal Servers,” on page 351
- ◆ Chapter 33, “Advanced Distribution: Creating Distribution Rules,” on page 353
- ◆ Chapter 34, “Advanced Distribution: Transferring Applications Using BITS,” on page 365
- ◆ Chapter 35, “Advanced Distribution: Pre-Installing Applications,” on page 377
- ◆ Chapter 36, “Advanced Distribution: Configuring Fault Tolerance, Load Balancing, and Site Lists,” on page 387
- ◆ Chapter 37, “Advanced Distribution: Configuring Application Dependencies and Chains,” on page 399
- ◆ Chapter 38, “Users: Supporting Terminal Server Users,” on page 407
- ◆ Chapter 39, “Users: Supporting Disconnected Users,” on page 411
- ◆ Chapter 40, “Users: Supporting Remote Users,” on page 415
- ◆ Chapter 41, “Controlling Rogue Processes,” on page 425
- ◆ Chapter 42, “Verifying Applications,” on page 433
- ◆ Chapter 43, “Uninstalling Applications,” on page 435
- ◆ Chapter 44, “Reporting Application Events,” on page 441
- ◆ Chapter 45, “Metering Software Licenses,” on page 469
- ◆ Chapter 46, “Reference: AdminStudio ZENworks Edition,” on page 473
- ◆ Chapter 47, “Reference: ZENworks SnAppShot,” on page 475
- ◆ Chapter 48, “Reference: Application Object Settings,” on page 481

- ◆ Chapter 49, “Reference: Macros,” on page 593
- ◆ Chapter 50, “Reference: Novell Application Launcher Tools,” on page 609
- ◆ Chapter 51, “Reference: Application Object Location,” on page 617
- ◆ Chapter 52, “Reference: Novell Application Launcher Authentication Hook,” on page 619
- ◆ Appendix J, “Documentation Updates,” on page 623

Novell Application Launcher: Components Overview

19

Novell® ZENworks® Desktop Management includes Novell Application Launcher™, workstation-based software that is used to deliver applications to the workstation.

Application Launcher is a 32-bit application that supports Windows 98 SE, Windows 2000, and Windows XP. When run on a workstation, Application Launcher reads Novell eDirectory™ to provide access to the applications the logged-in user and the workstation have been given rights to. If the user or workstation is not authenticated to eDirectory, Application Launcher uses the eDirectory information cached on the local workstation. It then controls every aspect of an application's use, from installing the application files, to mapping required drives, to uninstalling the application.

Application Launcher consists of three different user views that can be used to display, manage, and launch applications on user workstations: Application Window, Application Explorer, and Application Browser. If desired, all three views can be used at the same time on the same machine.

In addition to the three user views, Application Launcher includes the engine application, a Windows service, and a ZENworks Workstation Manager plug-in.

The following sections provide information to help you understand each of these components:

- ◆ [Section 19.1, “Application Window,” on page 239](#)
- ◆ [Section 19.2, “Application Explorer,” on page 241](#)
- ◆ [Section 19.3, “Application Browser,” on page 242](#)
- ◆ [Section 19.4, “Application Launcher Engine,” on page 246](#)
- ◆ [Section 19.5, “Application Launcher Service for Windows,” on page 246](#)
- ◆ [Section 19.6, “Application Launcher Workstation Helper,” on page 247](#)

19.1 Application Window

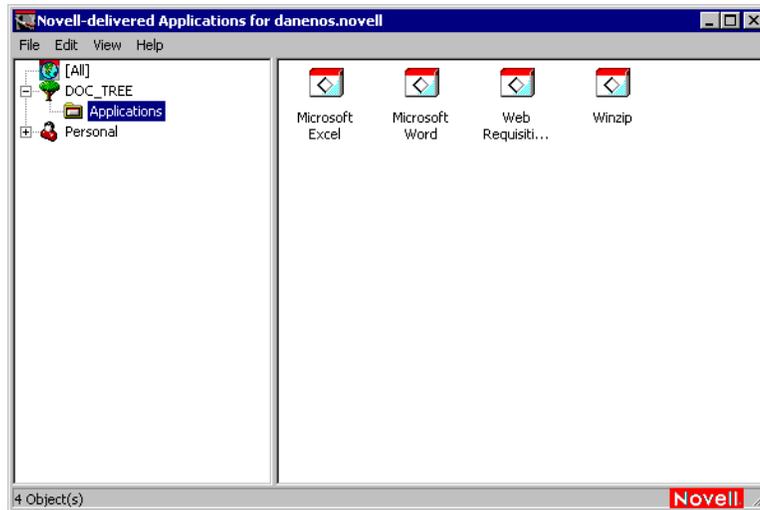
The Application Window is a standalone desktop window that provides greater administrative control of the user desktop than the other Application Launcher views.

The following sections contain additional information:

- ◆ [Section 19.1.1, “Application Window Description,” on page 240](#)
- ◆ [Section 19.1.2, “What Users Can Do with the Application Window,” on page 240](#)
- ◆ [Section 19.1.3, “Why Use the Application Window?,” on page 241](#)

19.1.1 Application Window Description

Figure 19-1 Application Window



The Application Window is divided into two panes. The left pane, referred to as the folder view, displays the following:

- ◆ **[All] folder:** Contains all applications that have been associated with the user or the workstation. This is an administrator-controlled feature. By default, it is enabled, which means the folder appears. You can disable the feature if desired. For instructions, see [Section 21.3, “Configuring User Settings,”](#) on page 268.
- ◆ **eDirectory trees:** Each tree contains the applications, located within the tree, that have been associated with the user or workstation. The Application Window displays only the trees to which the user and workstation are authenticated.
- ◆ **Personal folder:** Provides a location for the user to create personal folders for organizing applications. This is an administrator-controlled feature. By default, it is disabled, which means the folder does not appear. For information about enabling personal folders, see [Section 21.3, “Configuring User Settings,”](#) on page 268.

When a user selects a tree or folder in the left pane, the right pane displays the items (folders or applications) that are contained within the tree or folder.

You can also configure the Application Window to not include the folder view (the left pane).

19.1.2 What Users Can Do with the Application Window

Using the Application Window, users can do the following:

- ◆ Run an application by double-clicking the application's icon in the right pane. Depending on the application and how it is configured in eDirectory, the Application Launcher might install files to the workstation, map drives, or change workstation configuration files or settings.
- ◆ View the properties of an application. The properties include a description of the application, information about people to contact for help with the application, the times when the application is available for use, and the workstation requirements established for the application.

- ◆ Verify (fix) problems with an installed application. Verifying an application causes Application Launcher to redistribute the application to the workstation.
- ◆ Uninstall an application. This is an administrator-controlled feature. By default, it is disabled. You can enable it on a per-application basis. For information, see [Section 43.1, “Enabling an Application to be Uninstalled,” on page 435](#).
- ◆ Create personal folders to organize applications. This is an administrator-controlled feature. By default, it is disabled. For information about enabling personal folders, see [Section 21.3, “Configuring User Settings,” on page 268](#).
- ◆ Log in to eDirectory through the Novell Client™ or the ZENworks Middle Tier Server. This is an administrator-controlled feature. By default, it is disabled.
- ◆ Disconnect Application Launcher from eDirectory so that the user can work offline. Disconnecting Application Launcher from eDirectory does not log the user out of eDirectory.

19.1.3 Why Use the Application Window?

The Application Window provides greater administrative control of the user desktop than either of the other two views. You should use this view if you want to partially or completely lock down user workstations.

For example, the Application Window lets you ensure that all distributed applications are located only in the Application Window. At the same time, you can also determine the way the applications are organized in the window.

Or, if you want even greater control, you can replace the Windows desktop with the Application Window. This lets you restrict user access to only the applications in the Application Window. For instructions, see [Section 20.4, “Using Application Launcher As the Windows Shell,” on page 253](#).

19.2 Application Explorer

Application Explorer integrates with Windows Explorer to make applications available through a standalone window, similar to the Application Window, and from the Windows desktop, the Start menu, the system tray, and the Quick Launch bar.

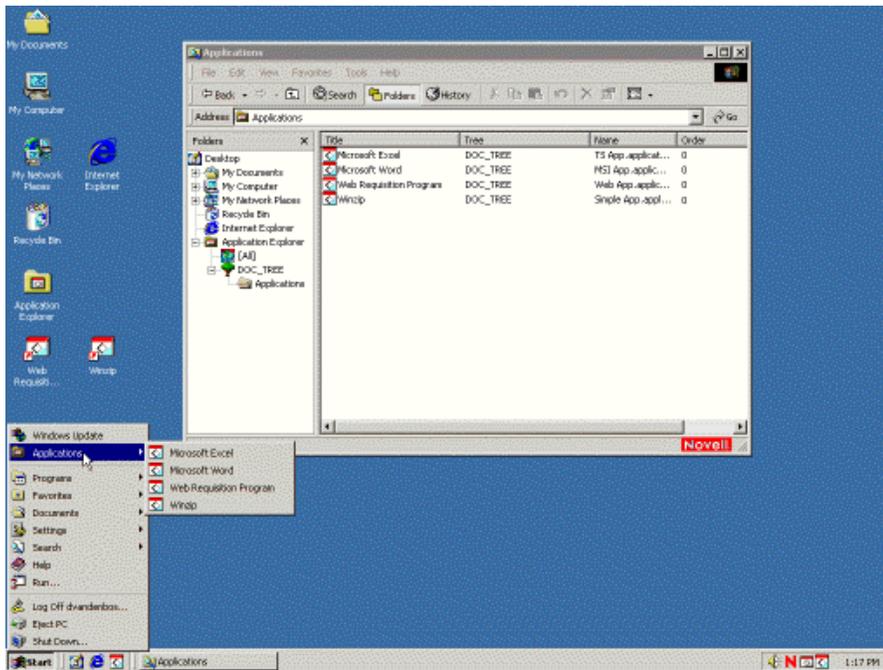
The following sections contain additional information:

- ◆ [Section 19.2.1, “Application Explorer Description,” on page 241](#)
- ◆ [Section 19.2.2, “Why Use Application Explorer?,” on page 242](#)

19.2.1 Application Explorer Description

In the following example screen, the Application Explorer window displays the applications contained in the DOC_TREE. In addition, these same four applications are also displayed on the Start menu, and two of the applications also appear on the desktop, one in the Quick Launch bar, and another in the system tray.

Figure 19-2 Application Explorer Window



The Application Explorer window contains the same features and functionality as the Application Window (see [Section 19.1, “Application Window,”](#) on page 239). In addition, because the Application Explorer window is an extension to Windows Explorer, it also includes functionality native to Windows Explorer, such as showing or hiding the folder view (left pane).

19.2.2 Why Use Application Explorer?

You should use Application Explorer if you don't require complete control of user desktops and you want take advantage of all the locations (Start menu, Windows desktop, and so forth) where application shortcuts can be placed.

19.3 Application Browser

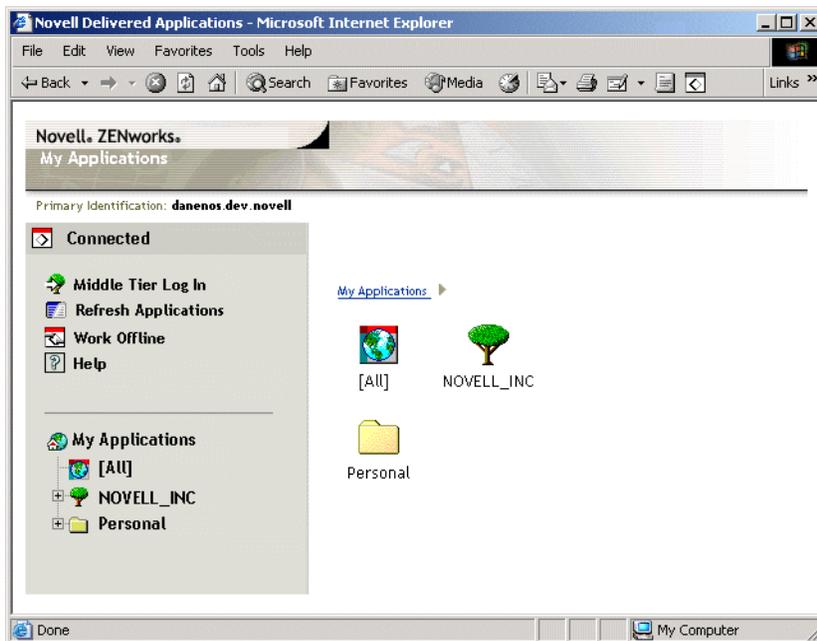
Application Browser, shown below, is a Web browser view similar to the Application Window and the Application Explorer window.

The following sections contain additional information:

- ◆ [Section 19.3.1, “Application Browser Description,”](#) on page 243
- ◆ [Section 19.3.2, “What Users Can Do with the Application Browser,”](#) on page 243
- ◆ [Section 19.3.3, “Why Use the Application Browser?,”](#) on page 244
- ◆ [Section 19.3.4, “Running the Application Browser Under Windows XP SP2 or Later,”](#) on page 244

19.3.1 Application Browser Description

Figure 19-3 Application Browser View



The Application Browser functionality is limited compared to the Application Window and Application Explorer. The view displays the following:

- ◆ **[All] folder:** Contains all applications that have been distributed to the user. This is an administrator-controlled feature. By default, it is enabled, which means the folder appears. You can disable the feature if desired. For instructions, see [Section 21.3, “Configuring User Settings,”](#) on page 268.
- ◆ **eDirectory trees:** Each tree contains the applications, located within the tree, that have been distributed to the user or workstation. Application Launcher displays only the trees to which the user is authenticated.
- ◆ **Personal folder:** Provides a location for the user to create personal folders for organizing applications. This is an administrator-controlled feature. By default, it is disabled, which means the folder does not appear. For information about enabling personal folders, see [Section 21.3, “Configuring User Settings,”](#) on page 268.

In the Application Browser, the Personal folder structure is for viewing and launching only. If users want to create or delete subfolders, add applications, or remove applications, they must use the Application Window or Application Explorer.

19.3.2 What Users Can Do with the Application Browser

Using the Application Browser, users can do the following:

- ◆ Run an application by double-clicking the application's icon in the right pane. Depending on the application and how it is configured in eDirectory, the Application Launcher might install files to the workstation, map drives, or change workstation configuration files or settings.

- ◆ View the properties of an application. The properties include a description of the application, information about people to contact for help with the application, the times when the application is available for use, and the workstation requirements established for the application.
- ◆ Verify (fix) problems with an installed application. Verifying an application causes Application Launcher to redistribute the application to the workstation.
- ◆ Uninstall an application. This is an administrator-controlled feature. By default, it is disabled. You can enable it on a per-application basis. For information, see [Section 43.1, “Enabling an Application to be Uninstalled,” on page 435](#).

19.3.3 Why Use the Application Browser?

The primary purpose of the Application Browser is to provide applications in a Web browser environment. You can enable users to launch the Application Browser independently, or you can integrate it into a Web portal, such as Novell Portal Services or Novell exteNd Director™, so that your applications are presented alongside Web content you've made available to users.

19.3.4 Running the Application Browser Under Windows XP SP2 or Later

Because of security changes in Windows XP Service Pack 2 (SP2), the Application Browser behaves differently on Windows XP SP2 or later workstations. As an administrator, you should communicate these changes to affected users.

The following sections contain additional information:

- ◆ [“Internet Explorer Information Bar and Add-On Installs” on page 244](#)
- ◆ [“Internet Explorer Local Machine Zone Lockdown and Active Content Blocking” on page 245](#)

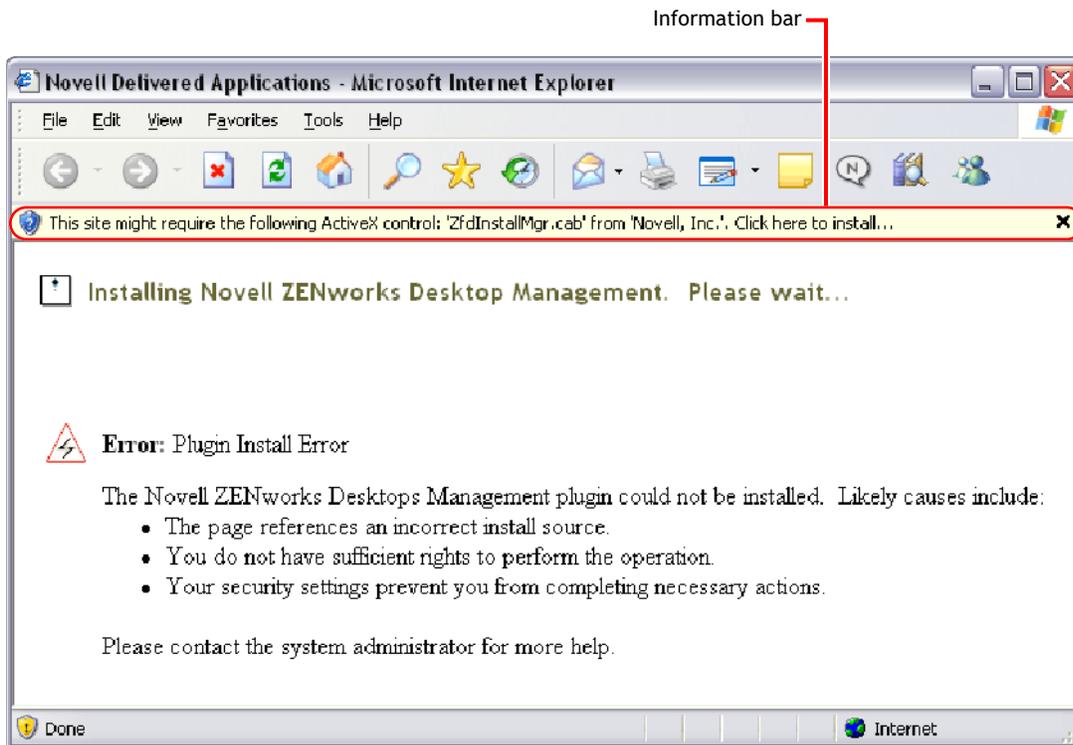
Internet Explorer Information Bar and Add-On Installs

When a Web page refers to an ActiveX control that is not currently on the workstation, users are asked whether or not they want the ActiveX control to be downloaded. In Windows XP SP2, this prompt is displayed in the information bar. The information bar displays in between the Internet Explorer toolbars and the Web page when a notification is present, and disappears on the next navigation.

NOTE: Displaying the information bar is enabled by default; however, the user can change the default setting so that the information bar does not display. Additionally, the ActiveX controls are installed automatically, without the information bar being displayed, if the publisher of the control (such as Novell) has been previously marked by the user as trusted.

When a user running Windows XP SP2 or later accesses the Application Browser for the first time (assuming that the user is using the default settings and has not marked Novell as trusted), the following error message displays in the Web page and the information bar displays directly above it.

Figure 19-4 Information Bar Displayed In Internet Explorer



Clicking the information bar, then clicking Install ActiveX Control displays the following dialog box, from which the user installs the ActiveX control.

Figure 19-5 Internet Explorer - Security Warning Dialog Box



Internet Explorer Local Machine Zone Lockdown and Active Content Blocking

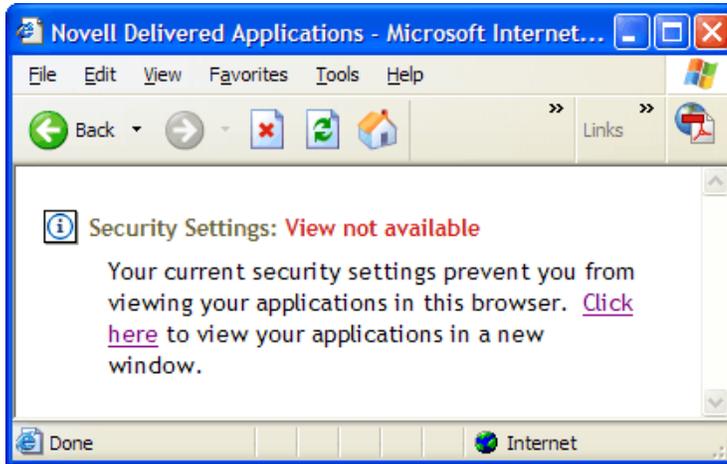
When Internet Explorer attempts to open a Web page, it places restrictions on what the page can do, based on the page's Internet Explorer security zone (in Internet Explorer, Tools > Internet Options). In Windows XP SP2 or later, the Local Machine Lockdown is more restrictive than in earlier versions of Windows XP.

Because the ZENworks Application Browser is a locally installed HTML control containing ActiveX controls, the Local Machine Lockdown feature, when combined with the new active content blocking feature, causes the Application Browser to behave differently on Windows XP SP2 or later workstations.

When a user types the URL to the Application Browser in the Address box or clicks a link to the Application Browser in the Favorites list, Internet Explorer shows an unformatted HTML page with the information bar showing the text “To help protect your security, Internet Explorer has restricted this file from showing active content that could access your computer. Click here for options.”

To avoid this situation and make it easier for users to access ZENworks Application Browser, the Application Browser view automatically detects if the Local Machine Lockdown feature is enabled on the workstation. If so, the following intermediate page displays:

Figure 19-6 Internet Explorer Security Warning Page



The user can then execute the link in this page, causing the Application Browser view to display in a new window, from which the user can access applications.

If the user accesses the Application Browser by clicking the *Application Browser* icon on the Standard Buttons toolbar in Internet Explorer, the intermediate page does not display and the Application Browser view displays without problems.

19.4 Application Launcher Engine

The Application Launcher engine, referred to simply as Application Launcher, performs the tasks required to manage applications associated with users or workstations. This includes such tasks as distributing, running, verifying, uninstalling, and caching applications.

Regardless of which view (Application Window, Application Explorer, or Application Browser) the user launches, Application Launcher is started in the background. It then accesses eDirectory (or the workstation's local cache directory if the user or workstation is not authenticated to eDirectory) to determine which applications to display to the user and to perform any other preconfigured tasks.

19.5 Application Launcher Service for Windows

On Windows 98, Application Launcher can perform all the tasks required to manage an application. On Windows 2000/XP, a user might not have all the workstation rights required by Application Launcher to perform its tasks. To ensure that it always has the necessary rights, Application

Launcher includes a Windows service (`nalntsrv.exe`), referred to as the NAL Service, that does the following:

- ♦ **Distribution, Caching, Uninstalling:** Ensures that applications can be distributed to, cached to, and uninstalled from the workstation regardless of the logged-in user's file system access. For example, if you've restricted user rights to specific local directories, the user might not have the file system and registry rights required to install an application to the workstation. The NAL Service, running in the system space, ensures that the application can be installed.
- ♦ **Launching:** Enables you to configure applications to launch and run under the System user's credentials rather than the logged-in user's credentials. This gives an application full rights to the file system and the registry, regardless of the logged-in user's rights.

You can choose from two security modes when launching the application as the System user: secure System user mode and unsecure System user mode.

Secure System user mode can be used if the user does not need to interact with the application (for example, you are applying a Service Pack); no interface is displayed to the user.

Unsecure System user mode can be used if the user requires interaction with the program (for example, a word processor); the normal interface is displayed to the user.

19.6 Application Launcher Workstation Helper

Application Launcher and the NAL Service manage all tasks for applications associated with the logged-in user. However, in addition to associating applications with users, you can associate applications with workstations. This requires that workstations be included in eDirectory as Workstation objects and that Workstation Manager be running on the workstations (see [Part III, “Automatic Workstation Import and Removal,” on page 125](#)).

When Workstation Manager starts, it loads the Application Launcher Workstation Helper (`zenappws.dll`). The Workstation Helper authenticates to eDirectory as the workstation (through the Workstation object). The Workstation Helper finds any applications associated with the workstation and performs any preconfigured management tasks associated with the applications. For example, if you've scheduled an application to be pre-installed in the middle of the night (also referred to as a “lights out” distribution), the application is distributed to the workstation by the Workstation Helper. Or, if you've configured an application to be launch immediately, the Workstation Helper launches the application.

The Workstation Helper does not have a user interface. Workstation-associated applications are only displayed when Application Launcher is running. When Application Launcher starts, it receives the list of workstation-associated applications from the Workstation Helper. Application Launcher then displays the workstation-associated applications just as it does the user-associated applications.

The Workstation Helper refreshes—that is, rereads eDirectory for changes to Application objects associated with the workstation—when 1) Application Launcher is started, 2) Application Launcher is manually refreshed, or 3) the Workstation Helper's scheduled refresh time occurs (see [Section 21.7, “Configuring Workstation Settings,” on page 277](#)).

Novell Application Launcher: Installing and Starting

20

The following sections provide information to help you install and start Novell® Application Launcher™:

- ◆ [Section 20.1, “Installing Application Launcher,” on page 249](#)
- ◆ [Section 20.2, “Installing the Application Launcher Plug-In,” on page 249](#)
- ◆ [Section 20.3, “Starting Application Launcher,” on page 250](#)
- ◆ [Section 20.4, “Using Application Launcher As the Windows Shell,” on page 253](#)
- ◆ [Section 20.5, “Application Window Command Line Switches,” on page 254](#)
- ◆ [Section 20.6, “Application Explorer Command Line Switches,” on page 261](#)

For additional information about Application Launcher, see [Chapter 19, “Novell Application Launcher: Components Overview,” on page 239](#).

20.1 Installing Application Launcher

The ZENworks® Desktop Management Agent installation program installs Application Launcher. All three Application Launcher views (Application Window, Application Explorer, and Application Browser) are installed, as well as the NAL Service for Windows and the Application Launcher Workstation Helper.

The Desktop Management Agent installation program must be run on each user's workstation. For instructions, see [“Installing and Configuring the Desktop Management Agent”](#) in the *Novell ZENworks 7 Desktop Management Installation Guide*.

NOTE: If you plan to use the Novell Client™ to enable Application Launcher to authenticate to Novell eDirectory™ and access NetWare® servers, you need to install it on your user workstations. For Novell Client requirements and installation instructions, see [“User Workstation Requirements”](#) in the *Novell ZENworks 7 Desktop Management Installation Guide*. If you plan to use the ZENworks Middle Tier Server rather than the Novell Client, and the Middle Tier Server is not already installed, see [“Installing the ZENworks Middle Tier Server”](#) in the *Novell ZENworks 7 Desktop Management Installation Guide*.

20.2 Installing the Application Launcher Plug-In

The Novell Application Launcher plug-in is a simplified version of Novell Application Launcher that can be used in place of the ZENworks Desktop Management Agent on workstations that only require distribution of user-associated applications. Or, it can be used to provide a Web-based roll-out of the Management Agent to workstations.

The Application Launcher plug-in includes only the Application Explorer and Application Browser views. It does not include the Application Window view, Application Launcher Service for Windows (nalntsrv.exe), or Application Launcher Workstation Helper (zenappws.dll). It

also does not include any other Desktop Management components, such as Workstation Manager, Workstation Inventory, Remote Management, or Workstation Imaging.

For information about installing the Application Launcher plug-in, see “[Installing the Novell Application Launcher Plug-In](#)” in the *Novell ZENworks 7 Desktop Management Installation Guide*.

20.3 Starting Application Launcher

The following sections explain how to manually start Application Launcher and how to automate starting the applications:

- ◆ [Section 20.3.1, “Windows 2000/XP File System Rights,”](#) on page 250
- ◆ [Section 20.3.2, “Manually Starting Application Launcher,”](#) on page 250
- ◆ [Section 20.3.3, “Automating Application Launcher Startup,”](#) on page 252

20.3.1 Windows 2000/XP File System Rights

To ensure that Application Launcher has the local file system access that it needs to distribute applications, make sure the user has the following rights on the workstation:

- ◆ At least Read access to the NAL cache directory (typically, `c:\nalcache`). For more information, see [Section 24.2, “File System Rights to the NAL Cache,”](#) on page 306.
- ◆ Full Control access to the user's temp directory (typically, `c:\documents and settings\username\local settings\temp`).
- ◆ Full Control access to the user's data encryption directory (typically, `c:\documents and settings\username\application data\microsoft\crypto`). This is required only if the user is using the Desktop Management Agent without a network client.
- ◆ Read\Write rights to the `HKEY_CURRENT_USER\Software\NetWare\NAL\1.0` registry key.
- ◆ Read rights to the `HKEY_LOCAL_MACHINE\Software\NetWare\NAL\1.0` registry key
- ◆ Read rights to the `HKEY_LOCAL_MACHINE\Software\Novell\ZENworks` registry key.

If you have not locked down the workstation, the appropriate rights can be granted through membership in the Users group.

In addition, the System user requires full access to all areas of the workstation so that the NAL Service (a component of Application Launcher) can distribute applications and launch applications configured to run in the “system” space. By default, this access is granted to the System user as a member of the Administrators group. Do not limit the default rights given to the Administrators group or the System user account.

For more information about Application Launcher file system access requirements, see [Chapter 23, “Novell Application Launcher: Managing Authentication and File System Access,”](#) on page 293.

20.3.2 Manually Starting Application Launcher

The view (Application Window, Application Explorer, or Application Browser) you want to use determines how you start Application Launcher.

The following sections contain additional information:

- ♦ [“Application Window” on page 251](#)
- ♦ [“Application Explorer” on page 251](#)
- ♦ [“Application Browser” on page 252](#)

Application Window

To start Application Launcher so that the Application Window view is displayed:

- 1 Click the *Start* menu > *Programs* > *Novell ZENworks Desktop Management* > *Application Window*.

or

Run `nalwin.exe` or `nalwin32.exe` from the `c:\program files\novell\zenworks` directory.

or

Run `nal.exe` from the `sys:\public` directory on the ZENworks Desktop Management Server.

In previous releases, `nal.exe` copied the Novell Application Launcher files to the workstation and then started the Application Window. Beginning with ZENworks for Desktops 4, `nal.exe` does not copy the files to the workstation; it only starts the Application Window if the Novell Application Launcher files are installed on the workstation. The primary purpose of the `nal.exe` file on the ZENworks Desktop Management Server is to maintain backward compatibility with login scripts created for previous versions.

IMPORTANT: You must ensure that you are not executing `nal.exe` from a server that has ZENworks for Desktops version 4.x or earlier installed. Running the ZENworks for Desktops 3.x `nal.exe` downgrades the Novell Application Launcher components on your workstation and causes error in the Application Launcher processes.

IMPORTANT: If you run `nal.exe` from the `sys:\public` directory and receive an error stating “The ordinal 6625 could not be located in the dynamic link library MFC42.DLL,” you need to update your server's `sys:\public\mfc42.dll` file. The correct version of `mfc42.dll` is installed to workstations by the ZENworks Desktop Management Agent. You can copy the file from a workstation's `c:\winnt\system32` directory to the server's `sys:\public\mfc42.dll` directory.

For information about command line switches that can be used when starting the Application Window, see [Section 20.5, “Application Window Command Line Switches,” on page 254](#).

Application Explorer

To start Application Launcher so that the Application Explorer view is displayed:

- 1 Click the *Start* menu > *Programs* > *Novell ZENworks Desktop Management* > *Application Explorer*.

or

Run `nalview.exe` or `naldesk.exe` from the `c:\program files\novell\zenworks` directory.

or

Run `nalexpld.exe` from the `sys:public` directory on the ZENworks Desktop Management Server.

In previous releases, `nalexpld.exe` copied the Novell Application Launcher files to the workstation and then started the Application Explorer. Beginning with ZENworks for Desktops 4, `nalexpld.exe` does not copy the files to the workstation; it only starts the Application Explorer if the Novell Application Launcher files are installed on the workstation. The primary purpose of the `nalexpld.exe` file on the Desktop Management Server is to maintain backward compatibility with login scripts created for previous versions.

IMPORTANT: You must ensure that you are not executing `nal.exe` from a server that has ZENworks for Desktops version 4.x or earlier installed. Running the ZENworks for Desktops 3.x `nal.exe` downgrades the Novell Application Launcher components on your workstation and causes error in the Application Launcher processes.

IMPORTANT: If you run `nalexpld.exe` from the `sys:\public` directory and receive an error stating “The ordinal 6625 could not be located in the dynamic link library MFC42.DLL,” you need to update your server's `sys:\public\mfc42.dll` file. The correct version of `mfc42.dll` is installed to workstations by the ZENworks Desktop Management Agent. You can copy the file from a workstation's `c:\winnt\system32` directory to the server's `sys:\public\mfc42.dll` directory.

For information about command line switches that can be used when starting the Application Window, see [Section 20.6, “Application Explorer Command Line Switches,” on page 261](#).

Application Browser

To start Application Launcher so that the Application Browser view is displayed:

- 1 Launch the Web browser, then click the *Application Browser* icon on the Standard Buttons toolbar.

20.3.3 Automating Application Launcher Startup

There are several ways to automatically start Application Launcher, including the following:

- ◆ Include the appropriate Application Window, Application Explorer, or Application Browser startup commands in the user's Windows or network login script.
- ◆ Add the Application Window or Application Explorer shortcut to the Windows Startup folder. The Desktop Management Agent installation program includes options to let you add either shortcut to the Startup folder.

20.4 Using Application Launcher As the Windows Shell

You can use Application Launcher in place of Windows Explorer (the default Windows shell) to further restrict user access to applications on workstations. When you do so, the Application Window view replaces the standard Windows desktop.

- ♦ [Section 20.4.1, “Setting Up Application Launcher As the Shell on Windows 98,” on page 253](#)
- ♦ [Section 20.4.2, “Setting Up Application Launcher As the Shell on Windows 2000/XP,” on page 253](#)

20.4.1 Setting Up Application Launcher As the Shell on Windows 98

- 1 Open the workstation's `system.ini` file (typically, `c:\windows\system.ini`) with a text editor.
- 2 Replace the `shell=explorer.exe` line with the following line:
`shell=c:\program files\novell\zenworks\nalwin.exe`
If you want the Application Window to be maximized when it comes up, add the `/max` switch to `nalwin.exe` (for example, `nalwin.exe /max`).
- 3 Save and close the `system.ini` file.
- 4 Restart Windows.

IMPORTANT: Starting Application Launcher under the Application Window shell is not supported. On a Windows 98 machine, doing so can cause general fault protection errors.

You should ensure that Application Launcher is not accidentally started. For example, remove the Application Explorer and Application Window shortcuts from the Start menu (*Start > Programs > ZENworks Desktop Management*), disable the user's ability to browse to the `c:\program files\novell\zenworks` directory, and verify that `nalwin.exe`, `nal.exe`, `nalwin32.exe`, and `naldesk.exe` are not being executed in the user's login scripts (Windows login script, Novell login script, etc.).

20.4.2 Setting Up Application Launcher As the Shell on Windows 2000/XP

- 1 On the Windows 2000/XP workstation, run `regedit.exe` and locate the following setting:
`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon`
- 2 Change the SHELL value from `explore.exe` to:
`c:\program files\novell\zenworks\nalwin.exe`
If you want the Application Window to be maximized when it comes up, add the `/max` switch to `nalwin.exe` (for example, `nalwin.exe /max`).
- 3 Close `regedit.exe`.
- 4 Restart Windows.

20.5 Application Window Command Line Switches

The following command line switches can be used when starting the Application Window.

The Application Window can be started by using the `nalwin.exe` or `nalwin32.exe` files, located in the workstation's `c:\program files\novell\zenworks` directory, or the `nal.exe` file, located in the ZENworks Desktop Management Server's `sys:public` directory. These switches work with all three files.

IMPORTANT: When you use a command line switch to distribute (`/a`), uninstall (`/l`), or verify (`/v`) an application, Application Launcher performs the action in the user space, not in the workstation space (even if the application is associated with the workstation). Therefore, the logged-in user must have the eDirectory rights and file system rights required to perform the distribution, uninstall, or verification of the application.

Table 20-1 Application Window Command Line Switches

Switch	Description
:	Skips the initial splash screen.
EXAMPLE: <code>nalwin :</code>	
<code>/a="tree:.app_object_dn"</code>	Distributes and launches the specified Application object, regardless of whether or not it has been associated with the user or workstation, as long as the user has the file system rights needed to distribute and launch the application and the rights to read Application object properties.
EXAMPLE: <code>nalwin /a="nov:.ms word.app"</code>	In addition to distributing and launching the specified application, Application Launcher continues running and displays the user and workstation-associated applications in the Application Window. If you don't want Application Launcher to do this, you can use the <code>/f</code> switch to instruct it not to read eDirectory for associations and the <code>/h</code> switch to hide the Application Window.
	<i>Tree</i> is the eDirectory tree where the Application object resides. The default tree is assumed if no tree name is specified.
	<i>App_object_dn</i> is the fully distinguished name of the Application object.
	NOTE: The quotation marks are required only if a space is used in the tree name or Application object name.

Switch	Description
<pre>/c="window_title"</pre> <p>EXAMPLE: <pre>nalwin /c="%cn%'s Apps"</pre></p>	<p>Uses the specified text as the title for the Application Window. The default window title is "Novell-delivered Applications for <i>User</i>," where <i>User</i> is the distinguished name of the user.</p> <p>The text specified can contain macros (variables) to display eDirectory information. The example, <code>/c="%cn%'s Apps"</code> substitutes the user's common name in the title (for example, JSmith's Apps).</p> <hr/> <p>NOTE: The quotation marks are required only if a space is used in the window title.</p> <hr/> <p>For information about macros, see Chapter 49, "Reference: Macros," on page 593.</p>
<pre>/f</pre> <p>EXAMPLE: <pre>nalwin /f /h</pre></p>	<p>Instructs Application Launcher not to read eDirectory to get the list of applications that have been associated with the logged-in user or the workstation. This is useful when using the <code>/a</code>, <code>/v</code>, or <code>/l</code> switches to simply distribute, verify, or uninstall a specific application. The example causes Application Launcher to start with the Application Window hidden (<code>/h</code>) not read eDirectory for all of the applications associated to the logged-in user or workstation (<code>/f</code>), and then distribute and launch the Word application (<code>/a=.word.apps</code>).</p> <p>This switch also requires you to use the <code>/h</code> switch.</p>
<pre>/h</pre> <p>EXAMPLE: <pre>nalwin /h</pre></p>	<p>Hides the Application Window.</p> <p>This switch is required when using the <code>/f</code> switch and is useful with the <code>/a</code>, <code>/v</code>, and <code>/l</code> switches.</p>
<pre>/l="tree:.app_object_dn"</pre> <p>EXAMPLE: <pre>nalwin /l="nov:.ms word.app"</pre></p>	<p>Uninstalls the specified Application object.</p> <p>In addition to uninstalling the specified application, Application Launcher continues running and displays the user and workstation-associated applications in the Application Window. If you don't want Application Launcher to do this, you can use the <code>/f</code> switch to instruct it not to read eDirectory for associations and the <code>/h</code> switch to hide the Application Window.</p> <p><i>Tree</i> is the name of the tree where the Application object resides. The default tree is assumed if no tree name is specified.</p> <p><i>App_object_dn</i> is the fully distinguished name of the Application object.</p> <hr/> <p>NOTE: The quotation marks are required only if a space is used in the tree name or Application object name.</p> <hr/>
<pre>/max</pre> <p>EXAMPLE: <pre>nalwin /max</pre></p>	<p>Displays the Application Window maximized when first loaded, overriding the window state (size and position) that was saved when exiting the previous Application Window session.</p>
<pre>/min</pre> <p>EXAMPLE: <pre>nalwin /min</pre></p>	<p>Displays the Application Window minimized when first loaded, overriding the window state (size and position) that was saved when exiting the previous Application Window session.</p>

Switch	Description
<p>/nd</p> <p>EXAMPLE: nalwin /nd</p>	<p>Skips the warning dialog box that appears when a dial-up connection is detected.</p>
<p>/norm</p> <p>EXAMPLE: nalwin /norm</p>	<p>Displays the Application Window in its original state (size and position) when first loaded, as long as the Save Window Size and Position setting (<i>User object > ZENworks tab > Launcher Configuration page > Add button > Window tab</i>) is turned off. By default, the Save Window Size and Position setting is turned off.</p>
<p>/nq</p> <p>EXAMPLE: nalwin.exe /a=.word.app / p="file 1.txt" /nq</p>	<p>Used only with the /p switch, instructs Application Launcher to remove the enclosing quotes from the parameter string. In the example, if /nq is not used, the parameters passed to the application would be "file 1.txt". With the /nq switch, the parameter is passed as file 1.txt.</p>

Switch	Description
<p><code>/p=param</code> <code>/p="param1 param2 param3"</code></p> <p>EXAMPLE: <code>nalwin /a=.word.app /</code> <code>p=file.txt</code></p>	<p>Passes the specified parameters to the launched application. It is only effective when used with <code>/a</code> or <code>/v</code>. The specified parameters are added to any that are already defined in the Application object (<i>Run Options</i> tab > <i>Application</i> page > <i>Parameters</i> field). The following rules apply when using the switch:</p> <ul style="list-style-type: none"> ◆ Application Launcher reads everything after <code>/p=</code> until the first space. At that point, it assumes the end of the parameter string. If the parameter string includes spaces, you must enclose the entire string in quotes. <p>Example 1: <code>nalwin.exe /a=.word.app /p=file.txt</code> Passed to word.app: <code>file.txt</code></p> <p>Example 2: <code>nalwin.exe /a=.word.app /p="/</code> <code>f=file.txt /m / w"</code> Passed to word.app: <code>"/f=file.txt /m /w"</code></p> <p>Example 3: <code>nalwin.exe /a=.word.app /p=file 1.txt</code> Passed to word.app: <code>file</code></p> <p>In Example 1, the parameter string does not include spaces so no quotes are required. In Example 2, the parameter string includes three parameters, separated by spaces, so quotes are required. In Example 3, the parameter string includes a filename with spaces, but no quotes have been used, so only the first part of the filename is passed to the application.</p> <ul style="list-style-type: none"> ◆ Use the <code>/nq</code> (no quotes) switch if you don't want the enclosing quotes to be passed to the application. <p>Example 4: <code>nalwin.exe /a=.word.app /p="file</code> <code>1.txt" /nq</code> Passed to word.app: <code>file 1.txt</code></p> <p>Example 5: <code>nalwin.exe /a=.word.app /p="/</code> <code>f=file.txt /m / w" /nq</code> Passed to word.app: <code>/f=file.txt /m /w</code></p> <p>Example 5 is the same as Example 2. However, Example 5 includes <code>/nq</code> so that the enclosing quotes are not passed.</p> <ul style="list-style-type: none"> ◆ Use escaped quotes (<code>\</code>) for any parameters that must include quotes in order to be accepted by the application. <p>Example 6: <code>nalwin.exe /a=word.app /p="/f=\"file</code> <code>1.txt\" /r /q" /nq</code> Passed to word.app: <code>/f="file 1.txt" /r /q</code></p> <p>Example 7: <code>nalwin.exe /a=word.app /p="/f=file</code> <code>1.txt /r /q" /nq</code> Passed to word.app: <code>/f=file 1.txt /r /q</code></p> <p>Example 6 assumes that the application requires file <code>1.txt</code> to be enclosed in quotes because the filename includes spaces. Example 7 assumes that the application does not require filenames with spaces to be enclosed in quotes.</p> <ul style="list-style-type: none"> ◆ The total characters allowed for all parameters is 256.

Switch	Description
/r EXAMPLE: nalwin /r	Resets the Application Launcher launch flag to 0. This occurs automatically when Application Launcher terminates normally, However, if Application Launcher terminates abnormally (for example, a user uses Ctrl+Alt+Del to shut down <code>nalwin.exe</code>), the launch flag is not reset and Application Launcher cannot be restarted until the workstation is rebooted or this switch is used.

Switch	Description
<pre>/restrictonline=0 1 2 3</pre> <p>EXAMPLE: <pre>nalwin /restrictonline=1</pre></p>	<p>Determines the Application Launcher state (online or offline) as eDirectory connections become available. This switch is supported only when eDirectory authentication occurs directly through the ZENworks Desktop Management Agent; it is not supported when eDirectory authentication occurs through the ZENworks Middle Tier Server.</p> <p>The switch's four possible values are described below, with each value having both a startup behavior and a running behavior. The running behavior applies when, after initial startup, Application Launcher detects a change in eDirectory connections.</p> <p><code>/restrictonline=0</code></p> <ul style="list-style-type: none"> ♦ At startup: Application Launcher goes into online mode if it detects an eDirectory connection, unless the connection is through dial-up. In that case, Application Launcher prompts the user whether to go online or stay offline. ♦ Running: Regardless of the connection type (direct or dial-up), Application Launcher automatically goes into online mode if it detects an eDirectory connection. <p>The <code>/restrictonline=0</code> value results in the same behavior as Application Launcher's default behavior. In other words, using the 0 value is the same as not using the switch at all.</p> <p><code>/restrictonline=1</code></p> <ul style="list-style-type: none"> ♦ At startup: Application Launcher goes into online mode if it detects an eDirectory connection, unless the connection is through dial-up. In that case, Application Launcher stays in offline mode. ♦ Running: Same as the startup behavior. Application Launcher goes into online mode if it detects an eDirectory connection, unless the connection is through dial-up. In that case, Application Launcher stays in offline mode. <p><code>/restrictonline=2</code></p> <ul style="list-style-type: none"> ♦ At startup: Value not used at this time. ♦ Running: Value not used at this time. <p><code>/restrictonline=3</code></p> <ul style="list-style-type: none"> ♦ At startup: Application Launcher goes into online mode if it detects an eDirectory connection, otherwise, Application Launcher stays in offline mode. ♦ Running: If Application Launcher started in online mode, it remains in online mode as long as an eDirectory connection is available. If Application Launcher started in offline mode, it remains in offline mode even if an eDirectory connection becomes available.

Switch	Description
<pre>/singletree="tree"</pre> <p>EXAMPLE: nalwin /singletree=nov</p>	<p>Reads the specified eDirectory tree only, regardless of any other trees the user might be authenticated to. This option applies only to user-associated applications, not workstation-associated applications. For example, assume that the workstation has a Workstation object in TREE1. The user authenticates to TREE2 and starts Application Launcher against TREE2 (nalwin /singletree=TREE2). Application Launcher displays the user-associated application from TREE2 and the workstation-associated applications from TREE1.</p> <hr/> <p>NOTE: The quotation marks are required only if a space is used in the tree name.</p>
<pre>/s</pre> <p>EXAMPLE: nalwin /s</p>	<p>Makes the Application Window act like the Windows shell. For example, on the File menu, the Exit Application Launcher option changes to Shutdown and gives the user the standard Windows shutdown options.</p> <p>This is not a true replacement for the Windows shell. If users minimize the Application Window, they have access to the normal desktop. If you want the Application Window to replace the Windows shell, see Section 20.4, "Using Application Launcher As the Windows Shell," on page 253.</p> <hr/> <p>NOTE: On a Windows 98 workstation, when a user selects Shutdown > Close All Programs and Log On as a Different User, the user's Windows desktop remains displayed while the login dialog box is displayed; normally, the desktop is removed.</p>
<pre>/time=n</pre> <p>or</p> <pre>/timen</pre> <p>EXAMPLE: nalwin /time=30</p>	<p>Delays starting of Application Launcher for the specified number of seconds.</p> <p>Use this switch only if you want Application Launcher to wait extra time before loading. For example, if several other programs are starting from the login script and they all display splash screens, you might not want all of them displayed at the same time. Or, if you are using the <i>/a</i> switch to launch an application from a login script, you might want to delay the start until all other commands in the login script have been executed.</p>
<pre>/u</pre> <p>EXAMPLE: nalwin /u</p>	<p>Terminates all applications, exits the Application Window, and unloads Application Launcher from memory.</p>

Switch	Description
<code>/v="app_object_dn"</code>	Verifies and launches the specified application.
<p>EXAMPLE: <code>nalwin /v="nov:.snap.apps"</code></p>	<p>In addition to verifying and launching the specified application, Application Launcher continues running and displays the user and workstation-associated applications in the Application Window. If you don't want Application Launcher to do this, you can use the <code>/f</code> switch to instruct it not to read eDirectory for associations and the <code>/h</code> switch to hide the Application Window.</p> <p><i>Tree</i> is the name of the tree where the Application object resides. The default tree is assumed if no tree name is specified.</p> <p><i>Application_object_dn</i> is the fully distinguished name of the Application object.</p>
	<p>NOTE: The quotation marks are required only if a space is used in the tree name or Application object name.</p>

20.6 Application Explorer Command Line Switches

The following command line switches can be used when starting Application Explorer.

Application Explorer can be started by using the `nalview.exe` or `naldesk.exe` files, located in the workstation's `c:\program files\novell\zenworks` directory, or the `nalexpld.exe` file, located in the ZENworks Desktop Management Server's `sys:public` directory. These switches work with all three files.

Table 20-2 Application Explorer Command Line Switches

Switch	Description
<p><code>/nd</code></p> <p>EXAMPLE: <code>nalview /nd</code></p>	Skips the warning dialog box that appears when a dial-up connection is detected.
<p><code>/ns</code></p> <p>EXAMPLE: <code>nalview /ns</code></p>	Skips the initial splash screen.

Switch	Description
<pre>/restrictonline=0 1 2 3</pre> <p>EXAMPLE: <pre>nalview /restrictonline=1</pre></p>	<p>Determines the Application Launcher state (online or offline) as eDirectory connections become available. This switch is supported only when eDirectory authentication occurs directly through the ZENworks Desktop Management Agent; it is not supported when eDirectory authentication occurs through the ZENworks Middle Tier Server.</p> <p>The switch's four possible values are described below, with each value having both a startup behavior and a running behavior. The running behavior applies when, after initial startup, Application Launcher detects a change in eDirectory connections.</p> <p><code>/restrictonline=0</code></p> <ul style="list-style-type: none"> ♦ At startup: Application Launcher goes into online mode if it detects an eDirectory connection, unless the connection is through dial-up. In that case, Application Launcher prompts the user whether to go online or stay offline. ♦ Running: Regardless of the connection type (direct or dial-up), Application Launcher automatically goes into online mode if it detects an eDirectory connection. <p>The <code>/restrictonline=0</code> value results in the same behavior as Application Launcher's default behavior. In other words, using the 0 value is the same as not using the switch at all.</p> <p><code>/restrictonline=1</code></p> <ul style="list-style-type: none"> ♦ At startup: Application Launcher goes into online mode if it detects an eDirectory connection, unless the connection is through dial-up. In that case, Application Launcher stays in offline mode. ♦ Running: Same as the startup behavior. Application Launcher goes into online mode if it detects an eDirectory connection, unless the connection is through dial-up. In that case, Application Launcher stays in offline mode. <p><code>/restrictonline=2</code></p> <ul style="list-style-type: none"> ♦ At startup: Value not used at this time. ♦ Running: Value not used at this time. <p><code>/restrictonline=3</code></p> <ul style="list-style-type: none"> ♦ At startup: Application Launcher goes into online mode if it detects an eDirectory connection, unless the connection is through dial-up. In that case, Application Launcher stays in offline mode. Same startup behavior as <code>/restrictonline=1</code>. ♦ Running: If Application Launcher started in online mode, it remains in online mode as long as an eDirectory connection is available. If Application Launcher started in offline mode, it remains in offline mode even if an eDirectory connection becomes available.

Switch	Description
<pre data-bbox="188 260 440 281">/singletree="tree"</pre> <p data-bbox="188 310 310 331">EXAMPLE:</p> <pre data-bbox="188 340 521 361">nalview /singletree=now</pre>	<p data-bbox="578 260 1284 520">Reads the specified eDirectory tree only, regardless of any other trees the user might be authenticated to. This option applies only to user-associated applications, not workstation-associated applications. For example, assume that the workstation has a Workstation object in TREE1. The user authenticates to TREE2 and starts Application Launcher against TREE2 (<code>nalview /singletree=TREE2</code>). Application Launcher displays the user-associated application from TREE2 and the workstation-associated applications from TREE1.</p>
	<p data-bbox="578 554 1284 613">NOTE: The quotation marks are required only if a space is used in the tree name.</p>

Novell Application Launcher: Configuring Settings

21

You can use the Novell® Application Launcher™ settings to configure how Application Launcher works. The following sections provide information and instructions for using the configuration settings:

- ◆ [Section 21.1, “How Configuration Settings Are Applied,” on page 265](#)
- ◆ [Section 21.2, “Accessing the Application Launcher Configuration Settings,” on page 265](#)
- ◆ [Section 21.3, “Configuring User Settings,” on page 268](#)
- ◆ [Section 21.4, “Configuring Window Settings,” on page 273](#)
- ◆ [Section 21.5, “Configuring Explorer Settings,” on page 275](#)
- ◆ [Section 21.6, “Configuring Browser Settings,” on page 276](#)
- ◆ [Section 21.7, “Configuring Workstation Settings,” on page 277](#)
- ◆ [Section 21.8, “Designating the Top of a Configuration Tree,” on page 279](#)

21.1 How Configuration Settings Are Applied

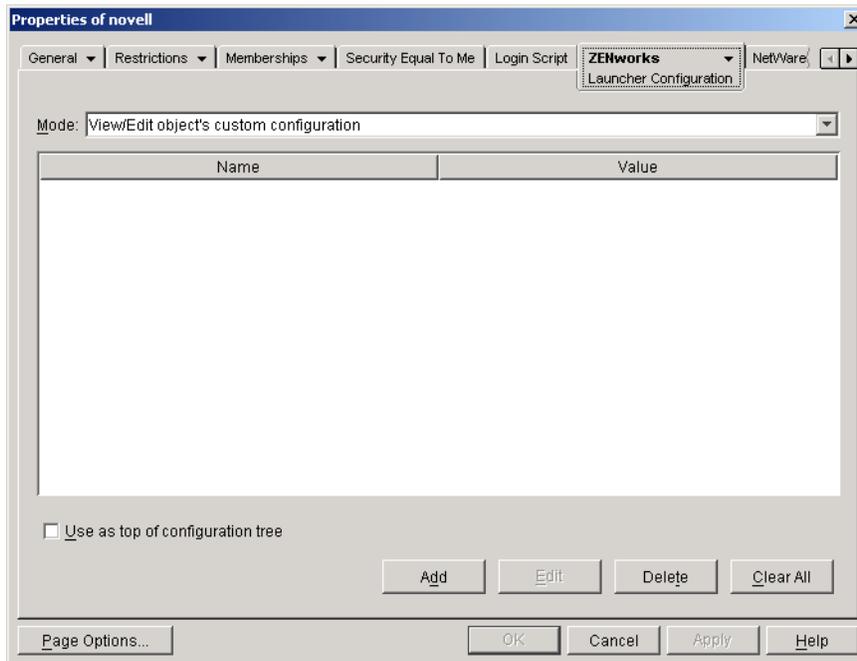
You can configure settings at the object or container level. When Application Launcher starts, it searches the Novell eDirectory™ tree, starting with the User object, for the settings that should be applied to the logged-in user. If a setting has not been defined at the User object, Application Launcher looks at the User object's parent container to see if the setting has been defined at that level. If not, Application Launcher continues up the eDirectory tree until it reaches the container object that has been designated as the top of the configuration tree (see [Section 21.8, “Designating the Top of a Configuration Tree,” on page 279](#)). If the setting is still not defined, it is considered undefined, or “unset,” and the preset default value is applied. Every configuration setting has a preset default value.

Application Launcher and the Application Launcher Workstation Helper follow the same process for determining configuration settings for the workstation, starting with the Workstation object.

21.2 Accessing the Application Launcher Configuration Settings

- 1 In ConsoleOne®, right-click the User, Workstation, or container object to which you want to apply new settings, then click *Properties*.

- 2 Click the *ZENworks* tab, then click *Launcher Configuration* to display the Launcher Configuration page.



The Launcher Configuration page provides three modes you can use to view the configuration settings for the current object. By default, the *View/edit object's custom configuration* mode is selected.

- 3 Make sure the *View/edit object's custom configuration* mode is selected. This is the mode that lets you set configuration options for the object. All three modes are described below.

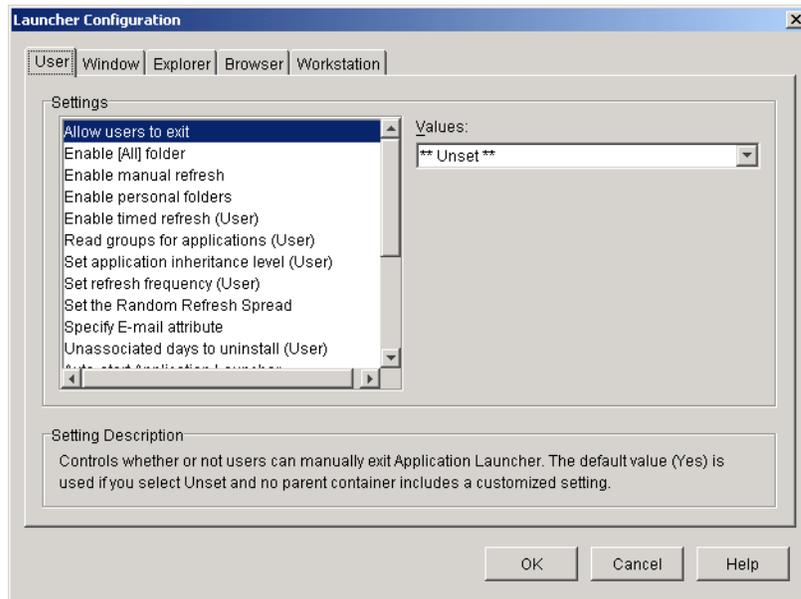
View/edit object's custom configuration (default mode): In this mode, the window lists all custom configuration settings that have been added and lists the settings values. Only customized settings are displayed; settings that are using the default or are receiving their value from their parent container (or higher) are not displayed.

This is the mode you must use to customize a configuration setting or edit a customized configuration setting.

View object's effective settings: In this mode, the window lists all configuration options with their effective settings. Effective settings might be from custom settings applied directly to the object, from settings inherited from a parent container, or from settings determined by the default settings values. This mode is useful when you want to see all settings that are being applied to the object.

View configuration tree: In this mode, the window displays the portion of the eDirectory tree that is being used to determine the configuration settings for the option. Only configuration options that are using a custom setting (either from the current object or from its parent container) are displayed. This mode is useful when you want to see where a setting is being inherited from.

- 4 Click *Add* to display the Launcher Configuration dialog box.



The Launcher Configuration dialog box has four or five tabs, depending on the type of object:

User: The User settings apply to Application Launcher regardless of the view (Application Window, application Explorer, Application Browser) that the user has open.

Window: The Window settings apply to the Application Window view.

Explorer: The Explorer settings apply to the Application Explorer view.

Browser: The Browser settings apply to the Application Browser view.

Workstation: The Workstation settings apply to the Application Launcher Workstation Helper. These settings are only used when the workstation has been imported into eDirectory as a Workstation object and Workstation Manager is running on the workstation.

- 5 To configure general settings for Application Launcher, see [Section 21.3, “Configuring User Settings,”](#) on page 268.

or

To configure settings for the Application Window view, see [Section 21.4, “Configuring Window Settings,”](#) on page 273.

or

To configure settings for the Application Explorer view, see [Section 21.5, “Configuring Explorer Settings,”](#) on page 275.

or

To configure settings for the Application Browser view, see [Section 21.6, “Configuring Browser Settings,”](#) on page 276.

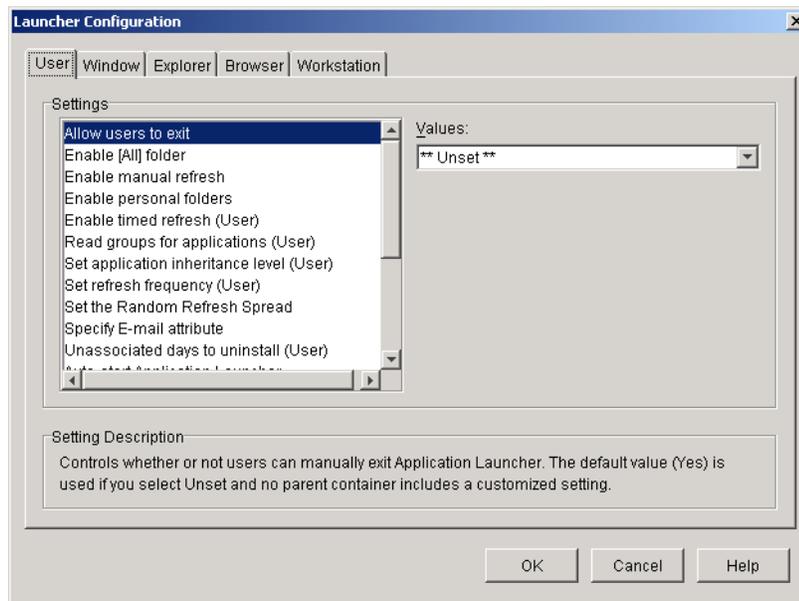
or

To configure settings for the Workstation Helper, see [Section 21.7, “Configuring Workstation Settings,”](#) on page 277.

21.3 Configuring User Settings

The User settings are general configuration settings that apply to Application Launcher regardless of the view (Application Window, Application Explorer, Application Browser) the user has open.

- 1 Make sure the Launcher Configuration dialog box is open. If it is not, see [Section 21.2, “Accessing the Application Launcher Configuration Settings,”](#) on page 265.
- 2 Click *User* to display the general configuration settings that apply to Application Launcher.



- 3 Configure the following settings:

Allow users to exit: Specify whether or not to allow the user to exit Application Launcher. If you choose *No*, the Exit Application Launcher option is removed from the File menu in the Application Window and the Application Explorer window.

The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

Enable [All] folder: Specify whether or not the user can see the [All] folder. The [All] folder displays all applications the user has access to, regardless of the application's eDirectory tree or folder.

The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

Enable manual refresh: By default, when Application Launcher starts on a user's workstation it reads the eDirectory information for all applications associated with the user. If an application's eDirectory information changes after Application Launcher starts, Application Launcher must refresh its information before the changes show up on the user's workstation.

Specify whether or not users can manually refresh Application Launcher to distribute any changes made to application information since the last time Application Launcher read eDirectory.

The *Enable manual refresh* option and *Enable timed refresh* option are not connected in any way except that they both control refresh. One option does not need to be selected for the other to work.

The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

Enable personal folders: Specify whether or not Application Launcher displays the Personal folder. If the Personal folder is displayed, users can organize their most used applications by moving them into the folder. They can even create subfolders.

When this option is enabled, the Application Window, Application Explorer window, and Application Browser all display the Personal folder. However, the Application Browser does not support creating personal folders. Personal folders created in the Application Window or the Application Explorer window display in the Application Browser, but the user cannot create additional personal folders.

The setting values are *Yes*, *No*, and *Unset*. The default value (*No*) is used if you select *Unset* and no parent container includes a customized setting.

Enable timed refresh: By default, when Application Launcher starts it reads the eDirectory information for all applications associated with the user. If an application's eDirectory information changes after Application Launcher starts, Application Launcher must refresh its information before the changes show up on the user's workstation.

Specify whether or not Application Launcher automatically refreshes applications so the user doesn't need to manually refresh them.

The setting values are *Yes*, *No*, and *Unset*. The default value (*No*) is used if you select *Unset* and no parent container includes a customized setting. If you set the option to *Yes*, the *Set refresh frequency* option determines how often Application Launcher refreshes applications.

Read group objects for applications: Specify whether or not you want Application Launcher to read Group objects to see if the user has been associated with any applications through membership in a group. Although groups are a convenient way of indirectly associating the user with applications, requiring Application Launcher to read Group objects can also decrease performance.

The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

Set application inheritance level: Specify how many parent containers Application Launcher searches for applications associated with the user.

The setting values are *-1*, *1* through *999*, and *Unset*. Entering *-1* instructs Application Launcher to search to the root of the eDirectory tree. The default value (*1*) is used if you select *Unset* and no parent container includes a customized setting.

Set refresh frequency: This option applies only if Application Launcher is configured to perform a timed refresh (see the *Enable timed refresh* option).

Specify how often you want Application Launcher to search eDirectory for new or changed Application objects associated with the user. A short timed refresh interval is very useful in situations where you want changes to refresh quickly. However, a short timed refresh interval can cause higher network traffic.

The setting values are *0* through *999,999,999* seconds and *Unset*. The default value (*43200* seconds or 12 hours) is used if you select *Unset* and no parent container includes a customized setting.

Set the random refresh spread: By default, when Application Launcher starts it immediately reads eDirectory to get information about the applications associated with the user. You can use this option to instruct Application Launcher to retrieve its application information from the user's local cache directory during startup and then refresh that information from eDirectory at

a later time. This allows you to reduce network traffic (generated by Application Launcher reading eDirectory) during peak times such as the morning login period.

The setting values are *Custom* and *Unset*. If you select *Custom*, specify a range from which Application Launcher determines the initial refresh time. For example, if you specify *3600* seconds (1 hour), Application Launcher randomly selects a number between 0 and 3600 and does the initial refresh that long after starting. Entering 0 in the range field causes Application Launcher to refresh immediately upon starting.

If you select *Unset* and no parent container includes a customized setting, the default value (0) is used, which means that Application Launcher refreshes at startup.

Specify e-mail attribute: Specify the eDirectory attribute you want to use to display e-mail names on the Help Contacts tab of the Application object's Properties dialog box. If the user has problems with applications, he or she can contact people by e-mail to get help. The e-mail name that appears is pulled from the eDirectory attribute you specify here. Users must have the eDirectory rights required to read the selected attribute for all users defined as contacts.

The setting values include all the eDirectory attributes and *Unset*. The default value (*Internet e-mail address*) is used if you select *Unset* and no parent container includes a customized setting.

Unassociated days to uninstall: Specify the number of days after which you want Application Launcher to uninstall an application that is no longer associated with the user.

The setting values are *-1* through *730* and *Unset*. Specify *0* if you want the application to be uninstalled as soon as the user is no longer associated with it. You should use *0* only for users who are located in a high-availability environment such as a LAN environment. You should not use this setting for users whose workstations are using a wireless network connection or a LAN environment that is unreliable. Using *0* as the setting in either of these environments can result in applications being uninstalled if the workstation loses its network connection during an application refresh; the best practice is to use the *-1* setting or a setting of 1 or greater.

Specify *-1* if you don't want the application uninstalled when unassociated with the user. The default value (*-1*) is used if you select *Unset* and no parent container includes a customized setting.

NOTE: This option requires you to have enabled the application to be uninstalled. For information, see [Chapter 43, "Uninstalling Applications,"](#) on page 435.

Auto-start Application Launcher: This setting applies only to pre-ZENworks 6.5 versions of Application Launcher. Beginning with ZENworks 6.5, this setting is replaced by the ability to designate a startup option in the ZENworks Desktop Management Agent installation program.

If you have pre-ZENworks 6.5 versions of Application Launcher, specify whether or not you want Application Launcher to be included in the user's Startup folder so that it automatically starts when the user logs on to the Windows workstation. When Application Launcher starts, it displays the same view (Application Window or Application Explorer) that was being used at the time it was added to the Startup folder.

The setting values are *Yes*, *No*, and *Unset*. The default value (*No*) is used if you select *Unset* and no parent container includes a customized setting.

Enable reading from removable cache: Specify whether or not you want to enable Application Launcher's ability to read information from the cache directory on removable media such as a CD, Jaz*, or Zip* drive.

If you set this option to *No*, you disable a user's ability to launch applications or install applications from removable media only. Application Launcher continues to read the cache on the local hard drive.

The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

Enable writing to the cache: Specify whether or not you want to enable Application Launcher's ability to write information to the cache directory.

The primary purpose of the cache directory is to enable users to work offline, disconnected from eDirectory. Setting this option to *No* disables the local cache, forcing users to be connected to eDirectory or a removable cache in order to access applications. To ensure that users don't manually go offline, the *Work offline* feature in Application Launcher is also disabled.

The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

Enable the checkpoint restart postpone button: Distribution of a large application across a slow link can require a significant amount of time. Specify whether you want Application Launcher to display a *Postpone* button that enables the user to postpone the distribution of an application to his or her workstation. If you enable the *Postpone* button, it is only displayed when Application Launcher detects that the user's workstation is running in remote mode (see the **Configure remote access detection method** option).

The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

Always evaluate referrals: Specify whether you want Application Launcher, when making a call to eDirectory, to evaluate the location of the eDirectory replicas and use the most accessible one at the time, or to force all eDirectory calls to go the first eDirectory replica found.

The setting values are *Yes*, *No*, and *Unset*. The default value (*No*) forces all eDirectory calls to the same replica and is used if you select *Unset* and no parent container includes a customized setting.

Enable automatic icon cleanup: This setting applies only when Application Launcher is running on a terminal server (Microsoft Windows Terminal Server or Citrix* MetaFrame*).

Specify whether or not you want Application Launcher, when exited, to remove application icons from the user's terminal server session desktop. Generally, you would set this option to *Yes* to have Application Launcher clean up. However, if you have multiple terminal server users who log in with the same username to run applications, you should disable this option. Otherwise, when one user exits Application Launcher, the application icon's disappears from all users' terminal server session desktops.

The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

Configure remote access detection method: If Application Launcher detects an active dial-up connection, it starts in remote mode. In the case where an active dial-up connection is not detected, you can specify how you want Application Launcher to detect if the user's workstation is running in local or remote mode. The setting values are:

- ♦ **User will always be local:** Application Launcher functions as if the user is local.
- ♦ **User will always be remote:** Application Launcher functions as if the user is remote.
- ♦ **Prompt:** Application Launcher prompts the user to select local or remote.
- ♦ **Auto detect using max interface speed:** Application Launcher detects the maximum speed of the network interface card and determines from that speed whether or not the user

is local or remote. If you select this option, you must establish the connection speed (threshold) that determines local or remote status.

IMPORTANT: This functionality uses the Windows `sensapi.dll`. On Windows NT 4, this DLL is installed only with Internet Explorer 6. If this functionality is not working on Windows NT 4 workstations, make sure the DLL exists by updating to IE 6.

- ♦ **Detect using network ID:** Application Launcher uses the workstation's network ID (also known as the network address) to establish whether or not the user is local or remote.

If you select this option, you must enter the network ID used to establish whether the user is local or remote. To determine the network ID, take the bit-wise logical AND comparison of the 32-bit IP address and 32-bit subnet mask, then convert the resulting 32-bit network ID to dotted decimal notation. In an AND comparison, the result of the two bits being compared is true (1) only when both bits are 1; otherwise, the result is false (0). For example:

10000001 00111000 10111101 00101001	(129.56.189.41 IP address)
11111111 11111111 11110000 00000000	(255.255.240.0 subnet mask)
10000001 00111000 10110000 00000000	(129.56.176.0 network ID)

If you want workstations whose network IDs match the specified network ID to be considered local, select *Network ID is equal to this network ID*.

If you want workstations whose network IDs do not match the specified network ID to be considered local, select *Network ID is not equal to this network ID*.

- ♦ **Unset:** The default value (*User will always be local*) is used if you select *Unset* and no parent container includes a customized setting.

Bring all popup windows to the front: Specify whether or not you want Application Launcher to ensure that a launched application appears in front of any other currently-opened desktop windows. The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

This setting has no effect on Windows 98 workstations because the default system configuration already forces pop-up windows to the front. However, the default system setting for Windows 2000 and Windows XP causes all pop-up windows to appear behind the current application's active window. With Application Launcher, this causes the launched application to appear behind the Application Launcher window. Enabling this option changes the Windows registry setting so that all pop-up windows, even the ones not generated by launching an application, appear as the front window. This is similar to using the Microsoft Tweak UI utility (`tweakui.exe`) to change the registry setting.

Enable Middle Tier login: Specify whether or not you want to add an Application Launcher menu item to enable users to log in through a ZENworks Middle Tier server (if they don't already have a Middle Tier session running). The setting values are *Yes*, *No*, and *Unset*. The default setting (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

Attempt to go online during refresh: Specify whether or not you want the logged-in user's workstation to attempt to go online during a refresh. The default setting is *Yes*.

This setting helps avoid long refresh times if the workstation is not connected or is connected over a slow link.

Enable BITS: Specify whether or not you want Application Launcher to be able to use the Microsoft Background Intelligent Service (BITS) to transfer a user's applications to the workstation. Unlike Application Launcher's standard transfer method which competes with other applications for available network bandwidth, BITS uses idle network bandwidth to transfer files, increasing or decreasing the rate at which files are transferred based on the amount of idle network bandwidth available. For example, if a network application begins to consume more bandwidth, BITS decreases its transfer rate.

The setting values are *Yes*, *No*, and *Unset*. The default setting (*No*) is used if you select *Unset* and no parent container includes a customized setting.

For additional configuration tasks you must complete to enable Application Launcher to use BITS, see [Chapter 34, "Advanced Distribution: Transferring Applications Using BITS," on page 365](#).

Allow User to Override BITS Transfer: Because BITS uses idle network bandwidth to transfer applications, an application might not be available when a user attempts to launch it. Specify the action you want to occur in this case:

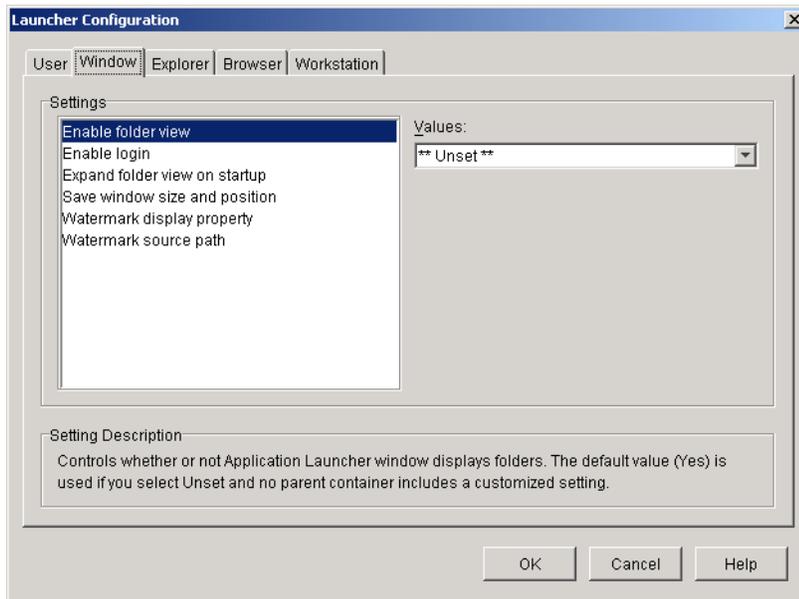
- ♦ *Yes* (default): Causes Application Launcher to cancel the BITS job and immediately transfer the application itself.
- ♦ *No*: Causes BITS to maintain control of the transfer; the user is unable to launch the application until the transfer is complete.
- ♦ *Prompt*: Prompts the user as to whether or not to interrupt the BITS transfer and have Application Launcher immediately download the application.

21.4 Configuring Window Settings

The Window settings are configuration settings that apply only to the Application Window view. When a user opens the Application Window view, these settings, along with any settings configured on the User page (see [Section 21.3, "Configuring User Settings," on page 268](#)), are applied.

- 1 Make sure the Launcher Configuration dialog box is open. If it is not, see [Section 21.2, "Accessing the Application Launcher Configuration Settings," on page 265](#).

- 2 Click *Window* to display the configuration settings that apply to the Application Window view.



- 3 Configure the following settings:

Enable folder view: Specify whether or not to display a folder list in the Application Window. When this option is enabled, the Application Window is divided into two panes: the right pane contains the folder list and the left pane displays the applications located in the selected folder. When this option is disabled, the Application Window consists of one pane only with all applications displayed in that pane.

Changes to this setting do not take effect until after a user exits and restarts the Application Window.

The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

Enable log in: Specify whether or not you want to add a Client32 Login option on the File menu in the Application Window so the user can run the Novell Client™ to log in to eDirectory. Before you enable the Log In option, make sure that Application Launcher can find the login program (`loginw32.exe`) on the user's workstation. For example, you could make sure that the login program's directory (typically `c:\winnt\system32`) is included in the PATH environment variable.

The setting values are *Yes*, *No*, and *Unset*. The default value (*No*) is used if you select *Unset* and no parent container includes a customized setting.

Expand folder view on startup: This option applies only if the folder view is enabled (see the *Enable folder view* option).

Specify whether or not you want to expand the entire folder tree when the Application Window opens.

The setting values are *Yes*, *No*, and *Unset*. The default value (*No*) is used if you select *Unset* and no parent container includes a customized setting.

Save window size and position: Specify whether or not to save the Application Window size and position settings.

The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

Watermark display property: This setting applies only if you've specified a watermark (see the *Watermark source path* option). Select *Default* to have the watermark placed in the upper left corner in the Application Window. Select *Tile* to have the watermark repeated to cover the entire window. The default setting is *Default*.

Watermark source path: Specify the location of the graphic file you want to use as the background wallpaper in the Application Launcher window. Graphics files larger than 172 Kb cannot be used as the background wallpaper. You can use any valid file path convention (for example, URL, mapped drive, or UNC path) and wallpaper graphics type (for example, *.bmp*, *.gif*, or *.jpg*).

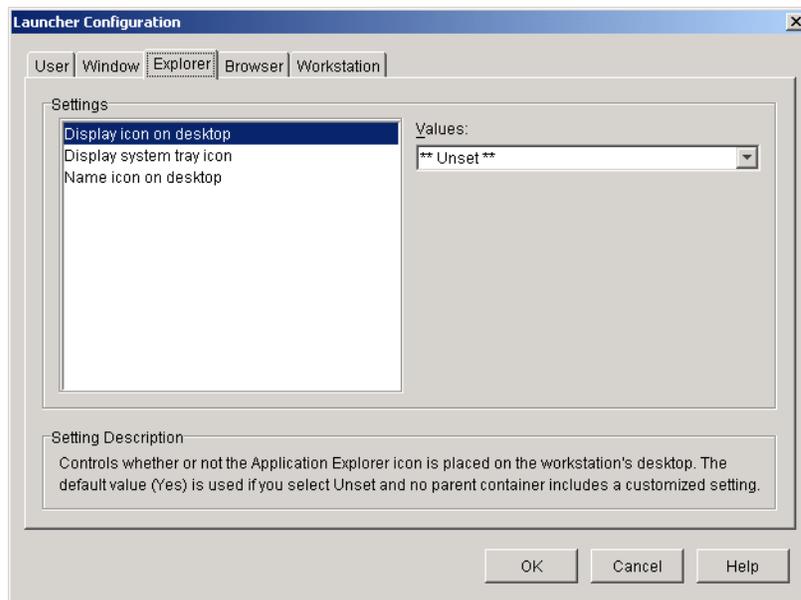
The ZENworks Desktop Management Agent does not support network drive mappings or UNC paths. You should only use these mapping types if users have a network client installed that enables access to the watermark source files. If this is not possible, you can also use an Application object to distribute the watermark to each workstation and use this Watermark Source Path field to enter the path to the local source.

The setting values are *Unset* and *Custom*. The default value (*Unset*) causes the setting to be inherited from the user's container. To override this setting, select *Custom* and then specify the appropriate file path.

21.5 Configuring Explorer Settings

The Explorer settings are configuration settings that apply only to the Application Explorer view. When a user opens the Application Explorer view, these settings, along with any settings configured on the User page (see [Section 21.3, "Configuring User Settings," on page 268](#)), are applied.

- 1 Make sure the Launcher Configuration dialog box is open. If it is not, see [Section 21.2, "Accessing the Application Launcher Configuration Settings," on page 265](#).
- 2 Click *Explorer* to display the configuration settings that apply to the Application Explorer view.



3 Configure the following settings:

Display icon on desktop: Specify whether or not you want the Application Explorer icon displayed on the user's desktop. The Application Explorer icon lets users open the Application Explorer window.

The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

Display system tray icon: Specify whether or not you want the Application Explorer icon displayed in the Windows system tray. The Application Explorer icon lets users open the Application Explorer window.

The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

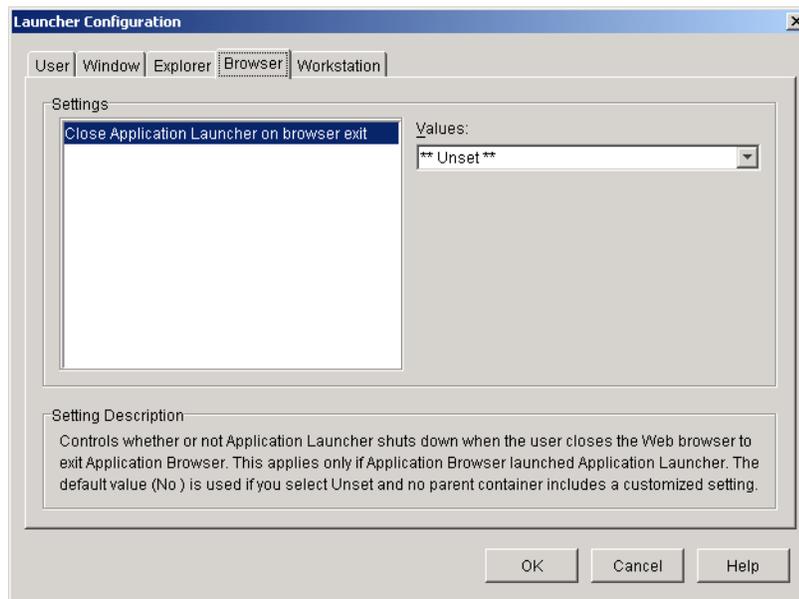
Name icon on desktop: Use this setting to change the name of the Application Explorer icon. The default name is Application Explorer.

The setting values are *Custom* and *Unset*. The default value (*Application Explorer*) is used if you select *Unset* and no parent container includes a customized setting. To use a different name, select *Custom* and enter the icon title.

21.6 Configuring Browser Settings

The Browser settings are configuration settings that apply only to the Application Browser view. When a user opens the Application Browser view, these settings, along with any settings configured on the User page (see [Section 21.3, “Configuring User Settings,” on page 268](#)), are applied.

- 1 Make sure the Launcher Configuration dialog box is open. If it is not, see [Section 21.2, “Accessing the Application Launcher Configuration Settings,” on page 265](#).
- 2 Click *Browser* to display the configuration settings that apply to the Application Browser view.



3 Configure the following settings:

Close Application Launcher on browser exit: This setting applies only to pre-ZENworks 7 versions of Application Launcher. Beginning with ZENworks 7, Application Launcher does not use this setting. Instead, it keeps track of the number of times its been called and shuts down only after the last view (Application Explorer, Application Window, or Application Browser) is exited.

With pre-ZENworks 7 versions, when a user opens the Application Browser view in his or her Web browser, Application Launcher is started if necessary. Specify whether or not you want Application Launcher to shut down when the user closes the Web browser to exit the Application Browser.

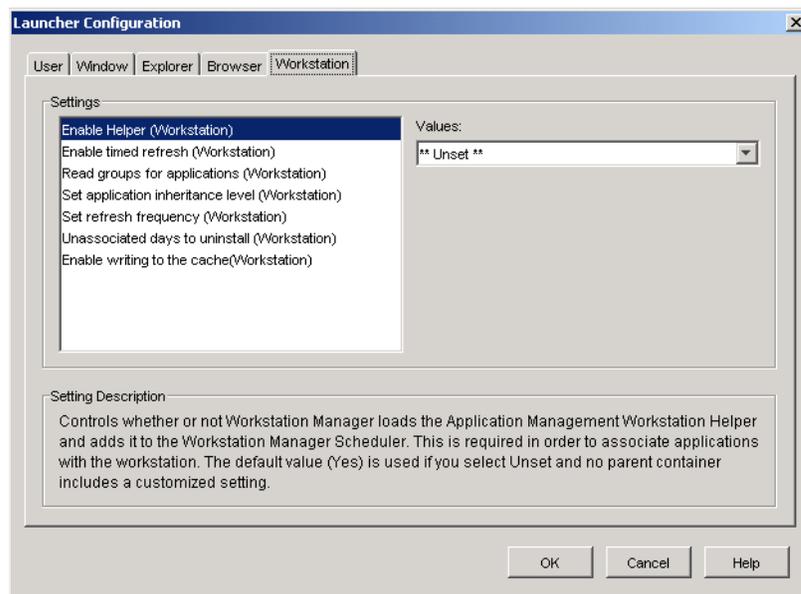
You should set this option to *Yes* only when users are using the Application Browser and neither of the other views (Application Window or Application Explorer). If users are also using the Application Window view or Application Explorer view, having Application Launcher close when the Web browser is exited also causes the Application Window and Application Explorer views to close.

The setting values are *Yes*, *No*, and *Unset*. The default value (*No*) is used if you select *Unset* and no parent container includes a customized setting.

21.7 Configuring Workstation Settings

The Workstation settings are configuration settings that apply to the Application Launcher Workstation Helper.

- 1 Make sure the Launcher Configuration dialog box is open. If it is not, see [Section 21.2, “Accessing the Application Launcher Configuration Settings,”](#) on page 265.
- 2 Click *Workstation* to display the configuration settings that apply to the Application Launcher Workstation Helper.



- 3 Configure the following settings:

Enable Helper: Specify whether or not you want to enable the Workstation Helper. Application Launcher reads eDirectory on behalf of logged-in users, but the Workstation Helper, running under Workstation Manager Scheduler, reads eDirectory on behalf of the

workstation and passes the application information to Application Launcher. You must enable the Workstation Helper if you want to be able to distribute workstation-associated applications and have Application Launcher display them.

The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

Enable Timed Refresh: Specify whether or not you want the Workstation Helper to periodically refresh application information from eDirectory. This enables the Workstation Helper to know about any new Application objects associated with the workstation or any updated information for currently-associated objects.

The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting. The *Set refresh frequency* option determines how often Workstation Helper refreshes applications.

NOTE: A Workstation Helper timed refresh does not cause Application Launcher to display new workstation-associated applications discovered by the Workstation Helper timed refresh. If Application Launcher is running on the workstation, Application Launcher only displays the newly discovered workstation-associated applications after the user performs a manual refresh or after Application Launcher performs its own timed refresh as defined by the Enable Time Refresh (User) setting on the User tab (see [Enable Timed Refresh](#) under [Section 21.3, “Configuring User Settings,”](#) on page 268). During its refresh, Application Launcher gets information about workstation-associated applications from Workstation Helper.

Read group objects for applications: Specify whether or not you want Workstation Helper to read Group objects to see if the workstation has been associated with any applications through membership in a group. Although groups are a convenient way of indirectly associating workstations with applications, reading Group objects can also decrease performance.

The setting values are *Yes*, *No*, and *Unset*. The default value (*No*) is used if you select *Unset* and no parent container includes a customized setting.

Set application inheritance level: Specify how many parent containers you want Workstation Helper to search for applications associated with the workstation.

The setting values are *-1* through *999* and *Unset*. The default value (*1*) is used if you select *Unset* and no parent container includes a customized setting.

Set refresh frequency: This setting applies only if Workstation Helper is configured to perform a timed refresh (see the *Enable timed refresh* option).

Specify how often you want the Workstation Helper to search eDirectory for new or changed Application objects associated with the workstation. A short timed refresh interval is very useful in situations where you want changes to refresh quickly. However, a short timed refresh interval can cause higher network traffic.

The setting values are *0* through *999,999,999* (seconds) and *Unset*. The default value (*43200*, which equates to 12 hours) is used if you select *Unset* and no parent container includes a customized setting.

Unassociated days to uninstall: Specify the number of days after which you want to uninstall an application that is no longer associated with the workstation.

The setting values are *-1* through *730* and *Unset*. Specify *0* if you want the application to be uninstalled as soon as the workstation is no longer associated with it. You should use *0* only for workstations that are located in a high-availability environment such as a LAN environment. You should not use this setting for workstations that are using a wireless network connection or a LAN environment that is unreliable. Using *0* as the setting in either of these environments can

result in applications being uninstalled if the workstation loses its network connection during an application refresh; the best practice is to use the *-1* setting or a setting of 1 or greater.

Specify *-1* if you don't want the application uninstalled when unassociated with the workstation. The default value (*-1*) is used if you select *Unset* and no parent container includes a customized setting.

Enable writing to the cache: Specify whether or not you want to enable Workstation Helper's ability to write information to the cache directory. The main purpose of this option is to disable the local cache, forcing users to access their applications through eDirectory or a cache directory on removable media.

The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

Enable reading from removable cache: Specify whether or not you want the Application Launcher to read information from the cache directories on removable media, such as a CD, Jaz*, or Zip* drive.

If you set this option to *No*, you disable a user's ability to launch applications or install applications from removable media only. Application Launcher continues to read the cache on the local hard drive.

The setting values are *Yes*, *No*, and *Unset*. The default value (*Yes*) is used if you select *Unset* and no parent container includes a customized setting.

Attempt to go online during refresh: Specify whether or not you want the logged-in workstation to attempt to go online during a refresh. The default setting is *Yes*.

This setting helps avoid long refresh times if the workstation is not connected or is connected over a slow link.

Enable BITS: Specify whether or not you want to be able to use the Microsoft Background Intelligent Service (BITS) to transfer a user's applications to the workstation. Unlike Application Launcher's standard transfer method which competes with other applications for available network bandwidth, BITS uses idle network bandwidth to transfer files, increasing or decreasing the rate at which files are transferred based on the amount of idle network bandwidth available. For example, if a network application begins to consume more bandwidth, BITS decreases its transfer rate.

The setting values are *Yes*, *No*, and *Unset*. The default setting (*No*) is used if you select *Unset* and no parent container includes a customized setting.

For additional configuration tasks you must complete to enable Application Launcher to use BITS, see [Chapter 34, "Advanced Distribution: Transferring Applications Using BITS,"](#) on [page 365](#).

21.8 Designating the Top of a Configuration Tree

When Application Launcher searches the eDirectory tree for a user's Application Launcher configuration settings, it starts with the User object. If the option has not been set for the User object, Application Launcher looks at the object's parent container to see if the option has been set at that level. If not, Application Launcher continues up the eDirectory tree searching for the option's setting until it reaches an object that has been designated as the top of the tree or the [ROOT]. The Application Launcher Workstation Helper follows the same process, starting at the Workstation object, to discover the configuration settings for the workstation.

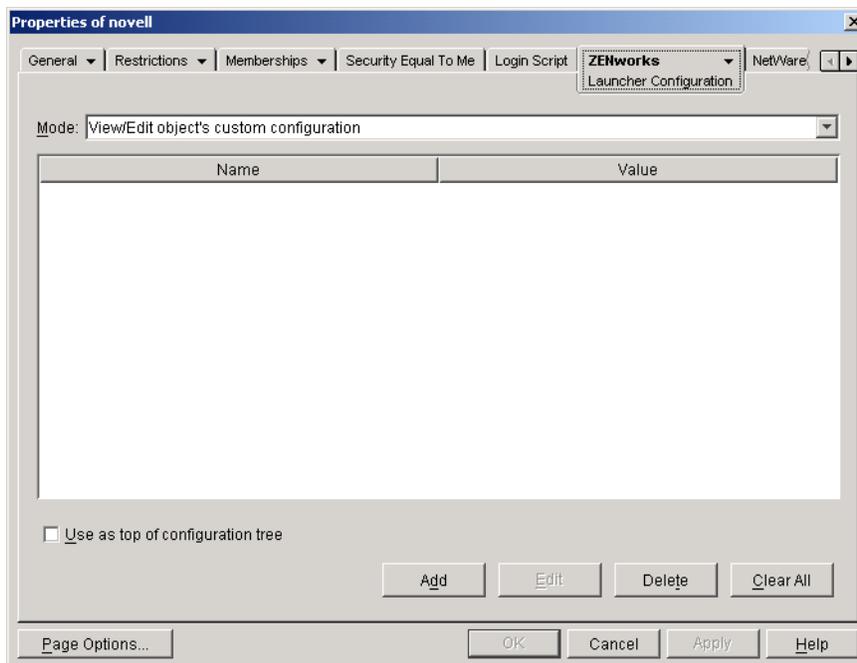
You can designate User, Workstation, or container objects as the top of a configuration tree. For example, if you want Application Launcher to read only the user's object for its configuration

settings, you would designate the User object as the top of the configuration tree. Or, if you want Application Launcher to read the user's object and its parent container object for its configuration settings, you would designate the parent container object as the top of the configuration tree.

Not all users and workstations must use the same top for the configuration tree. For example, for one user you might want Application Launcher to read the user's configuration settings only, but you want to use the parent container as the top of the configuration tree for all other users in the container. In this case, you would specify the User object as the top of the configuration tree for the one user, and specify the parent container as the top of the configuration tree for the other users.

To designate an object as the top of the configuration tree:

- 1 In ConsoleOne, right-click the User, Workstation, or container object you want to designate as the top of a configuration tree, then click *Properties*.
- 2 Click the *ZENworks* tab, then click *Launcher Configuration* to display the Launcher Configuration page.



- 3 Select the *Use as top of configuration tree* check box to designate this object as the top of a configuration tree.
- 4 Click *OK*.

Novell Application Launcher: Customizing Views

22

The following sections provide information about changes you can make to the Application Window, Application Explorer, and Application Browser views to customize their appearance:

- ♦ [Section 22.1, “Customizing the Application Window and Application Explorer Views,” on page 281](#)
- ♦ [Section 22.2, “Customizing the Application Browser View,” on page 282](#)

22.1 Customizing the Application Window and Application Explorer Views

You can customize the appearance of the Application Window and Application Explorer views by replacing the following graphics elements:

- ♦ The startup screen (Application Window and Application Explorer)
- ♦ The Application Explorer desktop icons used to represent connected mode and disconnected mode
- ♦ The Application Explorer system tray icons used to represent idle, offline, and refreshing
- ♦ The AVI file used during the distribution and caching processes (Application Window and Application Explorer)
- ♦ The AVI file used during the uninstall process (Application Window and Application Explorer)
- ♦ The Novell® banner in the lower-right corner of the Application Window and Application Explorer window
- ♦ The 12 system tray icons used to animate a refresh (Application Window and Application Explorer)

Each graphics file must have a specific name and must be placed in the `c:\program files\novell\zenworks\graphics` directory.

- 1 Create the new graphics elements you want, using the information in the following table.

Element	File Name
Startup screen	splash.bmp
Desktop icon - connected mode	desktop.ico
Desktop icon - disconnected mode	offlinedesktop.ico
System tray icon - idle	systray.ico
System tray icon - disconnected mode	offlinesystray.ico
Distribution and caching process	progress.avi
Uninstall process	uninstall.avi

Element	File Name
Novell banner	minibanner.bmp
System tray icons - refresh animation	refresh1.bmp - refresh12.bmp

- 2 Create a `\graphics` directory under the `c:\program files\novell\zenworks` directory on each user's workstation (for example, `c:\program files\novell\zenworks\graphics`).
- 3 Copy the new graphics files to the `\graphics` directory.

TIP: To avoid manually creating the directory and copying the files to each user's workstation, create a simple Application object that creates the `\graphics` directory and copies the files to it. Then associate the Application object with each user (or workstation) and mark it as *Force run*.

22.2 Customizing the Application Browser View

You can use the `myapps.html`, `axnalview.js`, `refresh.html`, and `hf_style.css` files to customize the functionality and look of the Application Browser view. The following table lists what each file controls:

Table 22-1 Application Browser Files

File	Controls...
<code>myapps.html</code> <code>axnalview.js</code>	<ul style="list-style-type: none"> ◆ Which ZENworks® Middle Tier Server is accessed ◆ Whether the view is Web portal enabled or not ◆ The banner graphic ◆ The banner height ◆ Whether or not the view includes the folder tree ◆ How the application icons are displayed ◆ The navigation elements that are displayed
<code>refresh.html</code>	<ul style="list-style-type: none"> ◆ The message displayed in the Application Browser view while Application Launcher starts and retrieves application information
<code>hf_style.css</code>	<ul style="list-style-type: none"> ◆ All styles that are applied to the view's HTML elements (heading, text, etc.)

The following sections provide information to help you use the files to customize the Application Browser view:

- ◆ [Section 22.2.1, “File Location,” on page 283](#)
- ◆ [Section 22.2.2, “Modifying the Myapps.html and Axnalview.js Files,” on page 284](#)
- ◆ [Section 22.2.3, “Creating a Refresh.html File,” on page 286](#)
- ◆ [Section 22.2.4, “Creating an Hf_style.css File,” on page 287](#)

22.2.1 File Location

The `myapps.html`, `axnalview.js`, `refresh.html`, and `hf_style.css` files reside, or can reside, in two locations, as shown in the following table:

Table 22-2 Application Browser File Locations

File	Location
<code>myapps.html</code> <code>axnalview.js</code>	<ul style="list-style-type: none">♦ <code>web_server_root\nwdocs</code> directory on the ZENworks Middle Tier Server♦ <code>c:\program files\novell\zenworks\nls\language</code> directory on the local workstation
<code>refresh.html</code>	This file does not exist until you create it. You must then place it in one of the following directories: <ul style="list-style-type: none">♦ <code>web_server_root\nwdocs\nal_html</code> directory on the ZENworks Middle Tier Server♦ <code>c:\program files\novell\zenworks\nal_html</code> directory on the local workstation
<code>hf_style.css</code>	This file does not exist until you create it. You must then place it in one of the following directories: <ul style="list-style-type: none">♦ <code>web_server_root\nwdocs\nal_html</code> directory on the ZENworks Middle Tier Server♦ <code>c:\program files\novell\zenworks\nal_html</code> directory on the local workstation

Myapps.html and Axnalview.js Files

The `myapps.html` and `axnalview.js` files contain the parameter settings that determine the appearance and functionality of the Application Browser view. They are installed with either of the following components:

- ♦ **Desktop Management Agent:** The files are installed by the Desktop Management Agent installation program as part of Application Launcher. For information about installing the Desktop Management Agent, see “[Installing and Configuring the Desktop Management Agent](#)” in the *Novell ZENworks 7 Desktop Management Installation Guide*.
- ♦ **NAL plug-in:** The NAL plug-in is a simplified version of Application Launcher that includes only the Application Explorer and Application Browser views. The NAL plug-in, including the `myapps.html` and `axnalview.js` files, is installed to the ZENworks Middle Tier Server. When a user accesses the `myapps.html` file on the Web server for the first time, the NAL plug-in installation manager (`zfdwebinstallmgr.dll`), which is installed to the Middle Tier Server along with the `myapps.html` file, installs the NAL plug-in (including `myapps.html` and `axnalview.js`) to the user's workstation.

For information about installing the Middle Tier Server, see “[Installing the ZENworks Middle Tier Server](#)” in the *Novell ZENworks 7 Desktop Management Installation Guide*. For information about installing and using the NAL plug-in, see “[Installing the Novell Application Launcher Plug-In](#)” in the *Novell ZENworks 7 Desktop Management Installation Guide*.

If you modify the `myapps.html` file and `axnalview.js` file on the Middle Tier Server, users who access the Application Browser from the server have the modified files downloaded to the

c:\program files\novell\zenworks\nls\language directory on their workstations. This ensures that the same Application Browser settings are used regardless of whether a user accesses the Application Browser from the Middle Tier Server or the local drive.

If you have users who only launch the Application Browser by accessing the `myapps.html` file on the workstation, you need to modify the `myapps.html` file and `axnalview.js` file on their workstations. You might want to modify single copies of the files and then use an Application object to push them to each user's workstation.

Refresh.html and Hf_style.css Files

The `refresh.html` and `hf_style.css` files do not exist until you create them. After you create them and place them in the appropriate directory, they are used in place of the Application Browser's internal settings.

If you add the `refresh.html` and `hf_style.css` files to the ZENworks Middle Tier Server, when a user accesses the `myapps.html` file on the server, the files are downloaded to the user's workstation to maintain consistency.

If users are not accessing the `myapps.html` file from a ZENworks Middle Tier Server, you need to add the files to each user's workstation. You might want to distribute them through Application Launcher at the same time you distribute the `myapps.html`.

22.2.2 Modifying the Myapps.html and Axnalview.js Files

Both the `myapps.html` file and `axnalview.js` files include the same nine parameters, shown below.

```
<!--param name="SingleTree\" value="ZENWORKS_TREE"-->
<!--param name="PortalView\" value="false"-->
<!--param name="BannerURL\" value="http://www.company.com/
banner.html"-->
<!--param name="BannerHeight\" value="80"-->
<!--param name="ShowTree\" value="true"-->
<!--param name="ShowTasks\" value="false"-->
<!--param name="AppDisplayType\" value="0"-->
<!--param name="ShowAppFrameNavigation\" value="true"-->
<!--param name="ShowIEToolbarButton\" value="true"-->
```

Even though the files contain the same parameters, you need to modify both files. This ensures that Internet Explorer can successfully launch the Application Browser with the correct settings, regardless of its Local Machine Lockdown and Active Content Blocking security settings.

If you install the Application Browser from a Middle Tier Server (see [“Installing the Novell Application Launcher Plug-In”](#) in the *Novell ZENworks 7 Desktop Management Installation Guide*.) rather than through the ZENworks Desktop Management Agent installation, you can customize two additional `myapps.html` parameters that are used by the installation manager:

```
<!--param name="MiddleTierAddress\" value="$$IPADDR$$"-->
<!--param name="Nt4PluginVersion\" value="4,0,1,0"-->
```

By default, the parameters are commented out, which results in the Application Browser using its preset internal values.

To modify a parameter:

- 1 Open the `myapps.html` file and `axnalview.js` files in an editor.

For information about the location of these files, see [Section 22.2.1, “File Location,” on page 283](#).

- 2 Remove the `!--` (beginning comment) and the `--` (ending comment) to activate the parameter.

For example:

```
<param name=\"SingleTree\" value=\"novell_tree\"
```

- 3 Modify the parameter's VALUE. Each parameter is described below.

SingleTree: This parameter lets you specify a single Novell eDirectory™ tree from which to read application information. If this parameter is used, Application Launcher ignores any other trees to which the user authenticates.

This parameter applies only at installation time. After installation, changes to this parameter have no effect.

PortalView: This parameter lets you better support portals by removing the banner section of the Application Browser view. The VALUE settings are *True* or *False*. *True* removes the banner section.

BannerURL: This parameter applies only if the *PortalView* parameter is set to *False*.

You can use this parameter to specify an alternate banner. For example, you could use a banner that contains your company's logo instead of the Novell logo. The VALUE setting must be a URL to an HTML page or graphics file (GIF, JPEG, etc.). If you specify an HTML page, the page is cropped to the height specified by the *BannerHeight* parameter. If you specify a graphics file, the banner section scrolls to fit the entire graphics file. Unless you want the banner section to be scrollable, you should ensure that the graphic's height is not greater than the height specified in the *BannerHeight* parameter.

BannerHeight: This parameter applies only if *PortalView* parameter is set to *False* and the default Novell banner is being overridden by the *BannerView* parameter.

You can use this parameter to determine the height of the banner section. The VALUE setting must be from 5 to 200. Any number less than 5 is rounded up to 5. Any number greater than 200 is rounded down to 200.

ShowTree: This parameter determines whether or not the Application Browser view includes the left pane (referred to as the folder view). The VALUE settings are *True* and *False*. *True* causes the folder view to display, and *False* causes it to be removed.

ShowTasks: This parameter determines whether or not the folder view (left pane) includes the *Work online/work offline*, *MiddleTier log in/Middle Tier log out*, *Refresh applications*, and *Help* options. The VALUE settings are *True* and *False*. *True* causes the options to display, and *False* causes them to be removed.

AppDisplayType: This parameter determines how the applications are displayed in the right pane of the Application Browser view. The VALUE settings are *0* and *1*. The default setting, *0*, causes the application icons to display as large icons, similar to the Large Icons view in Windows Explorer. The *1* setting causes the applications to be listed in table format, similar to the List view in Windows Explorer.

ShowAppFrameNavigation: This parameter determines whether or not the right pane of the Application Browser view includes navigation. The VALUE settings are *True* and *False*. *True*

causes the right pane to include navigation. The type of navigation depends on the setting for the `AppDisplayType` parameter:

- ◆ When the `AppDisplayType` parameter is set to *0* (large icons), navigation is displayed as a breadcrumb trail (for example, ZENworks Tree > Application Folder > Application A).
- ◆ When the `AppDisplayType` parameter is set to *1* (small icons), navigation is displayed as an Up arrow at the top of the applications list.

Setting this parameter's value to `False` causes the Application Browser view to start with the All folder open, meaning that all application icons are displayed in the right pane. If the All folder is disabled in the user's Launcher Configuration settings in ConsoleOne, this parameter is ignored (in other words, the default value of `True` is used).

ShowIEToolBarButton: This parameter determines whether or not the Application Browser button is added to the Internet Explorer toolbar. The Application Browser button launches the local version of the `myapps.html` file. The VALUE settings are `True` and `False`. The `True` setting is the default setting and causes the Application Browser button to be added to the toolbar. The `False` setting causes the Application Browser button to be removed from the toolbar. Whenever this setting is changed, the user must close Internet Explorer and then reopen it for the change to take affect.

MiddleTierAddress: This parameter applies only if you've configured a ZENworks Middle Tier Server for access to eDirectory.

The `MiddleTierAddress` parameter lets you specify the IP address of the Middle Tier Server that the user authenticates to. It is used only if a Middle Tier Server address is not included in the Windows registry; the registry includes the address if the user entered it when running the ZENworks Desktop Management Agent installation program.

Nt4PluginVersion: The ZENworks 7 NAL plug-in is not supported on Windows NT 4 workstations (see “[Interoperability with Windows NT 4 Workstations](#)” in “[Interoperability with ZENworks for Desktops 4.x](#)” in the *Novell ZENworks 7 Desktop Management Installation Guide* for more information about Windows NT 4 support). You can, however, continue to use the ZfD 4.0.1 NAL plug-in on those workstations. This parameter specifies the minimum version of the ZfD 4.0.1 NAL plug-in that runs on Windows NT workstations.

For example, the default parameter setting (4,0,1,0) requires only that the original ZENworks for Desktops 4.0.1 NAL plug-in be installed on the workstation. If you place an updated ZENworks for Desktops 4.0.1 NAL plug-in (`zfd40.cab`) with a version number higher than 4,0,1,0 (for example, 4,0,1,1 or 4,0,1,2) on the Middle Tier Server, workstations that have the 4.0.1.0 NAL plug-in are upgraded to the new version only if you change this parameter to the new version. In other words, upgrades do not happen automatically when you place a newer NAL plug-in version on the Middle Tier Server; if you want workstations upgraded, you must force the upgrade by bumping up the version number specified in this parameter.

- 4 Save the file.

22.2.3 Creating a Refresh.html File

When Application Launcher first starts, it reads eDirectory and the local cache for information about the applications it needs to display. If Application Launcher is started by opening the Application Browser view, the following message is displayed in the view while Application Launcher retrieves its application information:

Figure 22-1 Application Progress Message

 **In Progress: Your application list is being generated.**

You can replace this message by creating a `refresh.html` file that contains the message and graphics you want displayed during startup.

- 1 Create the file you want to use. Be aware of the following guidelines:
 - ♦ The file can contain text, graphics, and other standard HTML elements.
 - ♦ Make sure referenced files are available. If the `refresh.html` file is being accessed from a workstation rather than the ZENworks Middle Tier Server, use an absolute URL rather than a relative URL. For example, when referencing an image, use `` rather than ``. Any files referenced from the `refresh.html` file, regardless of the location, are downloaded to user's workstations along with the `refresh.html` file. In addition, the references are modified to point to the files on the workstation. Only referenced files one level deep are downloaded. For example, if `refresh.html` references `message.html`, which references `banner.jpg`, `message.html` is downloaded but not `banner.jpg`.
- 2 Name the file `refresh.html`.
- 3 Create a `nal_html` directory in one of the following two locations:
 - ♦ Under the `web_server_root\nwdocs` directory on the ZENworks Middle Tier Server (for example, `apache\nwdocs\nal_html`). The `nal_html` directory and its contents are copied to each user's workstation the same way the `myapps.html` file is copied. For information, see “[File Location](#)” on page 283.
 - ♦ Under the `c:\program files\novell\zenworks` directory on each user's workstation, if users won't access the `myapps.html` page from the Middle Tier Server.
- 4 Copy the `refresh.html` file to the `nal_html` directory.

22.2.4 Creating an `Hf_style.css` File

The Application Browser uses an internal default cascading style sheet (`.css`) to determine the look of elements such as headings and paragraph text. You can replace the default style sheet by creating an `hf_style.css` file that contains the style definitions you want used.

- 1 Create the cascading style sheet. You can use the default Application Browser cascading style sheet, shown below, as an example or template.
- 2 Name the file `hf_style.css`.
- 3 Create a `nal_html` directory in one of the following two locations:
 - ♦ Under the `web_server_root\nwdocs` directory on the ZENworks Middle Tier Server (for example, `apache\nwdocs\nal_html`). The `\nal_html` directory, and its contents, is copied to each user's workstation the same way the `myapps.html` file is copied. For information, see “[File Location](#)” on page 283.
 - ♦ Under the `c:\program files\novell\zenworks` directory on each user's workstation, if users won't access the `myapps.html` page from the ZENworks Middle Tier Server.
- 4 Copy the `hf_style.css` file to the `\nal_html` directory.

Default Application Browser Cascading Style Sheet

```
a { color: #039; font-family: "Trebuchet MS", Arial, Helvetica,
Geneva, Swiss, SunSans-Regular; text-decoration: underline }
body { font family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss,
SunSans-Regular }
p, option, li, ol, ul { color: black; font-size: 0.9em; font-family:
"Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-Regular }
th { font-family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss,
SunSans-Regular }
.datalink { color: #039; text-decoration:underline; margin-top: 2px }
.errorlink { color: #c82727; text-decoration: underline; margin-top:
2px }
.errorhead { color: #c82727; font-style: normal; font-weight: 800;
font-size: 1.2em; line-height: 1.5em; margin-bottom: 1.5em }
.formcolumnhead1 { color: #556c79; font-weight: bold; font-size:
0.7em; text-align: center }
.formhead1 { color: white; font-weight: bold; font-size: 1em; line-
height: 1.2em; background-color: #6b8899; text-align: left; text-
indent: 0.5em; letter-spacing: 0.1em; vertical-align: middle }
.formhead1b { color: white; font-weight: bold; font-size: 1em; line-
height: 1.2em; background-color: #6b8899; text-align: left; text-
indent: 0.5em; letter-spacing: 0.1em; vertical-align: middle }
.formhead2 { color: black; font-weight: bold; font-size: 1em; line-
height: 1.2em; text-align: left; vertical-align: middle }
.formdescriptext { color: #355263; font-size: 0.8em; margin-left:
1em }
.head1 { color: black; font-weight: bold; font-size: 1.3em; line-
height: 1.3em; font-family: "Trebuchet MS", Arial, Helvetica, Geneva,
Swiss, SunSans-Regular; margin-bottom: 0.6em }
.head1a { color: #663; font-weight: bold; font-size: 1.3em; line-
height: 1.3em; font-family: "Trebuchet MS", Arial, Helvetica, Geneva,
Swiss, SunSans-Regular; margin-bottom: 0.6em; margin-left: 1em }
.head1b { color: black; font-weight: bold; font-size: 1.3em; line-
height: 1.3em; font-family: "Trebuchet MS", Arial, Helvetica, Geneva,
Swiss, SunSans-Regular }
.head1w { color: white; font-weight: bold; font-size: 1.3em; line-
height: 1.3em; font-family: "Trebuchet MS", Arial, Helvetica, Geneva,
Swiss, SunSans-Regular }
.head2 { color: black; font-weight: bold; font-size: 1.2em;
font-family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss,
SunSans-Regular; margin-top: 0.4em }
.head2a { color: #663; font-weight: bold; font-size: 1.2em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular; margin-top: 0.4em }
.head2b { color: black; font-weight: 600; font-size: 1.2em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular; letter-spacing: 0.1em; margin-top: 0.4em; margin-bottom:
0.3em; border-bottom: 2px solid #6c8899 }
.head2tm6 { color: black; font-weight: bold; font-size: 1.2em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular; margin-top: 0.6em }
.head2w { color: #fff; font-weight: bold; font-size: 1.2em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
```

```

Regular }
.head2indent { color: black; font-weight: bold; font-size: 1.2em;
font-family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular; margin-top: 0.4em; margin-left: 0.5em }
.head3 { color: black; font-weight: bold; font-size: 1em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular; margin-top: 0.3em }
.head3a { color: #663; font-weight: bold; font-size: 1em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular; margin-top: 0.3em }
.head3b { color: black; font-weight: bold; font-size: 1em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, "Trebuchet MS",
Arial, Helvetica, Geneva, Swiss, SunSans-Regular; }
.head3center { color: black; font-weight: bold; font-size: 1em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular; text-align: center }
.head3indent { color: black; font-weight: bold; font-size: 1em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular; margin-top: 0.3em; margin-left: 1em }
.head3tm6 { color: black; font-weight: bold; font-size: 1em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular; margin-top: 0.6em }
.head3w { color: #fff; font-weight: bold; font-size: 1em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular }
.head4 { color: black; font-weight: bold; font-size: 0.85em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular; margin-top: 0.2em }
.head4b { color: #663; font-weight: bold; font-size: 0.85em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, "Trebuchet MS",
Arial, Helvetica, Geneva, Swiss, SunSans-Regular; margin-top: 0.2em }
.head4tm6 { color: black; font-weight: bold; font-size: 0.85em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular; margin-top: 0.6em }
.head5 { color: black; font-weight: 600; font-size: 0.75em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular; margin-top: 0.2em }
.head5b { color: #663; font-weight: 600; font-size: 0.75em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, "Trebuchet MS",
Arial, Helvetica, Geneva, Swiss, SunSans-Regular; margin-top: 0.2em }
.head5tm6 { color: black; font-weight: 600; font-size: 0.75em; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular; margin-top: 0.6em; margin-left: 2em }
.hint1 { color: #663; font-size: 12px; line-height: 14px; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular }
.hint1centered { color: #663; font-size: 12px; line-height: 14px;
font-family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular; text-align: center }
.hint1right { color: #663; font-size: 12px; line-height: 14px; font-
family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular; text-align: right }
.inactive { color: #738c9c }
.indent1 { margin-top: 0.3em; margin-left: 1em }

```

```

.indent1b { line-height: 18pt; margin-top: 0.5em; margin-left: 1em }
.indent1c { margin-top: 0.8em; margin-left: 1em }
.iconindent1 { margin-left: 18px }
.iconindent2 { margin-left: 34px }
.indent2 { margin-top: 0.3em; margin-left: 2em }
.indent3 { margin-top: 0.3em; margin-left: 3em }
.indentall { margin-left: 0.5em }
.listhead1 { color: black; font-style: normal; font-weight: 800;
font-size: 12pt; text-indent: .3em; }
.mainnavlink { font-weight: bold; font-size: 15pt; line-height: 18pt;
}
.margintop1 { margin-top: 0.3em }
.margintop2 { margin-top: 0.5em }
.margintop3 { margin-top: 1em }
.marginleft1 { margin-left: 1em }
.marginleft1b { font-size: 0.85em; margin-left: 1em }
.marginleft2 { margin-left: 2em }
.marginleft3 { margin-left: 3em }
.marginleft4 { margin-left: 4em }
.navlink { color: #030; font-weight: normal; text-decoration:
underline }
.nonproportional { color: black; font-size: 12pt; font-family:
"Courier New", Courier, Monaco }
.rowaltcolor { color: #efeee9 }
.serverhead1 { color: #c82727; font-weight: 800; font-size: 1.3em;
font-family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular }
.smalltext { font-size: 0.7em; line-height: 1em; font-family:
"Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-Regular }
.mediumtext { font-size: 0.85em; line-height: 1em; font-family:
"Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-Regular }
.smalltext2c { color: #355263; font-size: 0.7em; line-height: 1.1em;
font-family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular; width: 400px }
.textblue1 { color: #355263; font-size: 1em; font-family: "Trebuchet
MS", Arial, Helvetica, Geneva, Swiss, SunSans-Regular }
.smalltext2a { color: #663; font-size: 0.9em; line-height: 1.1em;
font-family: "Trebuchet MS", Arial, Helvetica, Geneva, Swiss, SunSans-
Regular }
.smalltext2b { color: black; font-weight: bold; font-size: 0.8em;
line-height: 1.1em; font-family: "Trebuchet MS", Arial, Helvetica,
Geneva, Swiss, SunSans-Regular }
.subtitle1 { color: black; font-weight: bold; font-size: 14px; line-
height: 14px; font-family: "Trebuchet MS", Arial, Helvetica, Geneva,
Swiss, SunSans-Regular }
.subtitle2 { color: white; font-weight: bold; font-size: 14px; line-
height: 14px; font-family: "Trebuchet MS", Arial, Helvetica, Geneva,
Swiss, SunSans-Regular }
.headpb { color: black; font-weight: bold; font-size: 15px; text-
align: left; vertical-align: top; letter-spacing:2px;}
.tabs { color: white; font-weight: bold; font-size: 12px; line-
height: 17px; text-decoration: none; background-color: #6b8899; text-
align: center }
.tablu { color: black; background-color:#DFDDD5; font-size:12px;

```

```

text-decoration:none; font-weight:bold; text-align:center }
.tab2s { color: white; font-weight: bold; font-size: 11px; text-
decoration: none; background-color: #6b8899 }
.tab2u { color: white; font-weight: normal; font-size: 11px; text-
decoration: none; background-color: #6b8899 }
span.tab1u { color: black }
span.tab2u { text-decoration: underline }
span.tab2s {text-decoration: underline }
.tablecolumnhead1 { color: #663; font-weight: bold; font-size: 0.7em;
text-align: center }
.tablecolumnhead2 { color: #663; font-weight: 600; font-size:
0.75em}
.tablehead1 { font-weight: bold; font-size: 1em; line-height: 1.1em;
background-color: #cc9; text-align: left; text-indent: 0.5em; letter-
spacing: 0.1em }
.tablehead1b { font-weight: bold; font-size: 1em; line-height:
1.1em; background-color: #cc9; text-align: left; text-indent: 0.5em }
.tablesubhead1 { color: black; font-weight: bold; font-size: 0.85em;
background-color: #efeee9; text-align: left; text-indent: 0.5em }
.tablehead3 { color: black; font-weight: bold; margin-left: .5em }
.tabb { font-weight: bold; font-size: 14pt;}
.tabw { color: white; font-weight: bold; font-size: 14pt; }
.task1 { margin-top: 0.3em; margin-left: 1em; font-size: 0.8em }
.task2 { margin-top: 0.3em; margin-left: 2em; font-size: 0.8em }
.windowheadgrey { color: #5a5958; font-weight: bold; font-size:
18pt; line-height: 18pt; letter-spacing: 1px }
.windowheadblack { color: #5a5958; font-weight: bold; font-size:
18pt; line-height: 18pt; letter-spacing: 1px }
.windowheadwhite { color: white; font-weight: bold; font-size: 18pt;
line-height: 18pt; letter-spacing: 1px }
.width350 { width: 350px }
.width400 { width: 400px }
.width450 { width: 450px }
.width500
width: 500px }

```


Novell Application Launcher: Managing Authentication and File System Access

23

In order to manage applications, Novell® Application Launcher™ requires access to Novell eDirectory™, the workstation's local file system, and, in some cases, a network server's file system.

The following sections provide information to help you manage the authentication and file system rights required by Application Launcher to distribute, launch, cache, and uninstall applications:

- ♦ [Section 23.1, “Novell eDirectory Authentication Overview,” on page 293](#)
- ♦ [Section 23.2, “File System Access Overview,” on page 294](#)
- ♦ [Section 23.3, “Authentication and File System Access for User-Associated Applications,” on page 295](#)
- ♦ [Section 23.4, “Authentication and File System Access for Workstation-Associated Applications,” on page 299](#)

23.1 Novell eDirectory Authentication Overview

Application Launcher authenticates to eDirectory to get information about the applications available to the logged-in user or the workstation.

Application Launcher can use the Novell Client™ to authenticate to eDirectory. Because the Novell Client uses NCP™ (NetWare® Core Protocol™) to access eDirectory, it works best for workstations that are running on a local area network inside a firewall.

If you do not have the Novell Client rolled out to workstations in your local area network, or you have users who must authenticate to eDirectory through a firewall, Application Launcher can also use the ZENworks Middle Tier Server. Novell Application Launcher uses the standard HTTP or HTTPS protocol to communicate with the ZENworks Middle Tier Server inside your firewall, which then accesses eDirectory on behalf of the user. For information about setting up a Middle Tier server, see [“Installing the ZENworks Middle Tier Server”](#) in the *Novell ZENworks 7 Desktop Management Installation Guide*.

A Note About the Application Launcher Cache

To enable users to run applications while disconnected from eDirectory, Application Launcher creates a cache directory, referred to as the NAL cache, on the local workstation.

The NAL cache contains the information required to display and launch applications. When the user is not authenticated to eDirectory, Application Launcher uses the information in the NAL cache. This means that after a user installs an application to his or her workstation, the application is available to the user regardless of whether or not he or she is authenticated to eDirectory.

You can force cache an application to the user's workstation to ensure that the application is available to the user when disconnected from eDirectory, even if it has not yet been installed. In this situation, the cache contains all the information and files required to install the application.

For more information about the cache, see [Chapter 24, “Novell Application Launcher: Managing the Cache,”](#) on page 303.

23.2 File System Access Overview

During distribution, launching, caching, or uninstalling of an application, Application Launcher requires access to the local file system and might require access to network file systems such as NetWare, Linux, or Windows servers.

Local File System Access

Local file system access is provided through the logged-in Windows user account and through the Windows System user account.

For Application Launcher to work properly, the logged-in user requires the following rights:

- ◆ At least Read access to the NAL cache directory (typically, `c:\nalcache`). For more information, see [Section 24.2, “File System Rights to the NAL Cache,”](#) on page 306.
- ◆ Full Control access to the user's temp directory (typically, `c:\documents and settings\username\local settings\temp`).
- ◆ Full Control access to the user's data encryption directory (typically, `c:\documents and settings\username\application data\microsoft\crypto`). This is required only if the user is using the Desktop Management Agent without a network client.
- ◆ Read\Write rights to the `HKEY_CURRENT_USER\Software\NetWare\NAL\1.0` registry key.
- ◆ Read rights to the `HKEY_LOCAL_MACHINE\Software\NetWare\NAL\1.0` registry key
- ◆ Read rights to the `HKEY_LOCAL_MACHINE\Software\Novell\ZENworks` registry key.

In addition, the System user requires full access to all areas of the workstation. By default, this access is granted to the System user as a member of the Administrators group. Do not limit the default rights given to the Administrators group or the System user account.

Network File System Access

File system access to NetWare, Linux, and Windows servers is provided through a network client, such as the Novell Client or the Client for Microsoft Networks, that supports the file system protocols (CIFS, Samba, NCP, etc.) used by the server. In general, network clients work well in a local area network environment.

If Application Launcher must access a file system through a firewall, or if you don't want to use a network client, you can use the ZENworks Middle Tier Server. The Middle Tier Server provides limited file system access to network servers. For information about setting up a Middle Tier server, see [“Installing the ZENworks Middle Tier Server”](#) in the *Novell ZENworks 7 Desktop Management Installation Guide*.

File System Access Limitations in a Clientless Environment

The ZENworks Middle Tier Server does not include the same file system access capabilities as a full network client such as the Novell Client and Microsoft Client. Although the Middle Tier Server can copy files from a network server location to a user's workstation, it does not provide the ability to

open or execute files located on a network server. Because of this limitation, the following distribution and launch issues exist when running Application Launcher in a clientless environment or an environment where the network client cannot establish the appropriate network server connection (for example, through a firewall):

- ♦ Application Launcher is not able to launch applications that reside on a network server (in other words, when the *Path to file* field on the *Run Options > Application* page points to a network location).
- ♦ Application Launcher requires MSI applications to be marked Force Cache in order to distribute them. Application Launcher copies the force cached applications to the workstation's NAL cache directory. Microsoft Windows Installer then installs the applications to the workstation from the cache.

File Path Syntax

When specifying a network file path for distributing or launching an application, you can use either mapped drives or UNC paths. The following path syntaxes are valid:

```
server\volume:path  
\\server_name\volume\path  
\\IP_address\volume\path  
volume_object_name:path  
directory_map_object_name:path  
driveletter:\path
```

If you use a mapped drive, the user's workstation must have the same drive mapping as the server from which you are running ConsoleOne.

If you use a UNC path, applications running on Windows 2000 servers might launch slowly because of the way Windows resolves UNC paths. For more information, see [Microsoft Knowledge Base Article Q150807](http://support.microsoft.com/support/kb/articles/Q150/8/07.asp) (<http://support.microsoft.com/support/kb/articles/Q150/8/07.asp>).

23.3 Authentication and File System Access for User-Associated Applications

The Application Launcher components, authentication methods, and file system access used when managing user-associated applications differ from a Windows 98 operating system to a Windows 2000/XP operating system, as explained in the following sections.

- ♦ [Section 23.3.1, “Windows 98 \(User-Associated Applications\),” on page 295](#)
- ♦ [Section 23.3.2, “Windows 2000/XP \(User-Associated Applications\),” on page 297](#)

23.3.1 Windows 98 (User-Associated Applications)

The following table lists the components, authentication method, and file system access used by Application Launcher when managing a user-associated application on a Windows 98 workstation.

Table 23-1 *Windows 98 (User-Associated Applications)*

Event	Responsible Component	eDirectory Authentication	Workstation File System Access	Network Server File System Access
Distribution	Application Launcher	eDirectory user (User object)	Windows user ¹	NetWare: Folder and file rights assigned to eDirectory user ² Windows: Permissions assigned to Active Directory user ³ Linux: Rights assigned to Samba user ⁴
Launch (normal)	Same as Distribution			
Launch (force run ⁵)	Same as Distribution			
Caching	Same as Distribution			
Uninstall	Application Launcher	eDirectory user (User object)	Windows user	Not applicable

¹ The Windows 98 operating system, unlike Windows 2000/XP, does not provide file system security for individual users. Each Windows 98 user account has full access to the local file system, which means that Application Launcher has all the file system access it requires.

² NetWare server file rights can be assigned through the Application object (*Common* tab > *File Rights* page). Any object that is associated with the Application object receives these rights. You can also directly assign rights to users through their User objects (*Rights to Files and Folder* tab > *Trustee File System Rights* page) or some other method, such as adding them to a group that has been assigned the appropriate rights.

³ Windows server file permissions must be assigned through the user's Active Directory account, which must have the same username and password as the user's eDirectory account. The user, workstation, Middle Tier Server (if used), and Windows server must be members of the same Windows domain. For information about using Novell DirXML to synchronize user account information between eDirectory and Active Directory, see “[Installing Nsure Identity Manager 2.02 Bundle Edition](#)” in “[Installing ZENworks 7 Desktop Management in a Windows Network Environment](#)” in the *Novell ZENworks 7 Desktop Management Installation Guide*.

⁴ Linux server file rights are assigned through Samba. The procedures for doing this depend on the Linux distribution (Open Enterprise Server Linux, SUSE Linux Enterprise Server, etc.) being used. For OES Linux and SUSE Linux Enterprise Server, see “[Configuring a Linux Server for ZENworks File Access](#)” in the *Novell ZENworks 7 Desktop Management Installation Guide*. For additional information, refer to the OES Linux and SUSE Linux Enterprise Server documentation available on the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation) or refer to the documentation for your Linux distribution.

⁵ The Force Run setting causes the application to automatically distribute after it becomes available. For information about configuring an application as Force Run, see “[Associations Page](#)” on [page 534](#).

23.3.2 Windows 2000/XP (User-Associated Applications)

The following table lists the components, authentication method, and file system access used by Application Launcher when managing a user-associated application on a Windows 2000/XP workstation.

Table 23-2 *Windows 2000/XP (User-Associated Applications)*

Event	Responsible Component	eDirectory Authentication	Workstation File System Access	Network Server File System Access
Distribution	NAL Service	eDirectory user	Windows System user ¹	NetWare: Folder and file rights assigned to eDirectory user ² Windows: Permissions assigned to Active Directory user ³ Linux: Rights assigned to Samba user ⁴
Launch (normal)	Application Launcher	eDirectory user	Windows user	NetWare: Folder and file rights assigned to eDirectory user ² Windows: Permissions assigned to Active Directory user ³ Linux: Rights assigned to Samba user ⁴
Launch from Server (secure/unsecure5)	NAL Service (when run as secure/unsecure System user ⁵)	eDirectory workstation	Windows System user (when run as secure/unsecure System user)	NetWare: Folder and file rights assigned to eDirectory workstation. (Novell Client) Windows: If the workstation is not a member of Active Directory on the Windows server where it accesses files, then "anonymous logon" permissions should be granted for files to be read there. Otherwise, folder and file rights assigned to everyone (guest). Linux: Rights assigned to guest/anonymous user
Launch (force run ⁶)	Same as Launch (normal)			
Caching	NAL Service	eDirectory user	Windows System user	NetWare: Folder and file rights assigned to eDirectory user Windows: Permissions assigned to Active Directory user Linux: Rights assigned to Samba user
Uninstall	NAL Service	eDirectory user	Windows System user	Not applicable

¹ For Application Launcher and its associated programs (NAL Service and Workstation Helper) to work properly, the Windows System user account must have full rights to all areas of the workstation. By default, this access is granted to the System user as a member of the Administrators group. Do not limit the default rights given to the Administrators group or the System user account.

In addition, Application Launcher requires that the user's Windows account provide the following rights:

- ◆ At least Read access to the NAL cache directory (typically, `c:\nalcache`). For more information, see [Section 24.2, “File System Rights to the NAL Cache,” on page 306](#).
- ◆ Full Control access to the user's temp directory (typically, `c:\documents and settings\username\local settings\temp`).
- ◆ Full Control access to the user's data encryption directory (typically, `c:\documents and settings\username\application data\microsoft\crypto`). This is required only if the user is using the Desktop Management Agent without a network client.
- ◆ Read\Write rights to the `HKEY_CURRENT_USER\Software\NetWare\NAL\1.0` registry key.
- ◆ Read rights to the `HKEY_LOCAL_MACHINE\Software\NetWare\NAL\1.0` registry key
- ◆ Read rights to the `HKEY_LOCAL_MACHINE\Software\Novell\ZENworks` registry key.

² NetWare file system rights can be assigned through the Application object (*Common tab > File Rights* page). Any user who is associated with the Application object receives these rights. You can also directly assign rights to users through their User objects (*Rights to Files and Folder tab > Trustee File System Rights* page) or some other method, such as adding them to a group that has been assigned the appropriate rights.

³ Windows server file permissions must be assigned through the user's Active Directory account, which must have the same username and password as the user's eDirectory account. The user, workstation, Middle Tier Server (if used), and Windows server must be members of the same Windows domain. For information about using Novell DirXML to synchronize user account information between eDirectory and Active Directory, see “[Installing Nsure Identity Manager 2.02 Bundle Edition](#)” in “[Installing ZENworks 7 Desktop Management in a Windows Network Environment](#)” in the *Novell ZENworks 7 Desktop Management Installation Guide*.

If the workstation is not a member of the Active Directory of the Windows server where it gets files, you need to grant “anonymous logon” permissions for files to be read from the Windows server.

⁴ Linux server file rights are assigned through Samba. The procedures for doing this depend on the Linux distribution (OES Linux, SUSE Linux Enterprise Server, etc.) being used. For OES Linux and SUSE Linux Enterprise Server, see “[Configuring a Linux Server for ZENworks File Access](#)” in the *Novell ZENworks 7 Desktop Management Installation Guide*. For additional information, refer to the OES Linux and SUSE Linux Enterprise Server documentation available on the [Novell Documentation Web site \(http://www.novell.com/documentation\)](#) or refer to the documentation for your Linux distribution.

⁵ The Secure System User and Unsecure System User settings apply to applications running on Windows 2000/XP only. These settings cause the application to run in the “system” space as the Windows System user rather than in the “user” space as the logged-in user. These settings are intended to ensure that users can run the application even if they have limited access rights to the workstation's file system. Only server files found in public locations at launch time are accessible to a user-associated application run with Unsecure System User. The user rights are never used for

remote files at launch time when running in Unsecure System User mode. If all files are local, they are accessible at launch time. For more information, see “[Environment Page](#)” on page 523.

⁶ The Force Run setting causes the application to automatically distribute after it becomes available. For information about configuring an application as Force Run, see “[Associations Page](#)” on page 534.

23.4 Authentication and File System Access for Workstation-Associated Applications

The Application Launcher components, authentication methods, and file system access used when managing workstation-associated applications differ from a Windows 98 operating system to a Windows 2000/XP operating system, as explained in the following sections:

- ◆ [Section 23.4.1, “Windows 98 \(Workstation-Associated Applications\),”](#) on page 299
- ◆ [Section 23.4.2, “Windows 2000/XP \(Workstation-Associated Applications\),”](#) on page 300

23.4.1 Windows 98 (Workstation-Associated Applications)

The following table lists the components, authentication method, and file system access used by Application Launcher when managing a workstation-associated application on a Windows 98 workstation.

Table 23-3 *Windows 98 (Workstation-Associated Applications)*

Event	Responsible Component	eDirectory Authentication	Workstation File System Access	Network Server File System Access
Distribution	Workstation Helper	eDirectory workstation (Workstation object)	Windows user ¹	NetWare: Folder and file rights assigned to eDirectory workstation ² Windows: Permissions assigned to Active Directory user ³ Linux: Rights assigned to Samba user ⁴
Launch (normal)	Same as Distribution.			
Launch (force run ⁵)	Workstation Helper	eDirectory workstation (Workstation object)	Windows user	NetWare: Folder and file rights assigned to eDirectory workstation Windows: Permissions assigned to Active Directory user Linux: Rights assigned to Samba user
Caching	Same as Launch (force run).			

Event	Responsible Component	eDirectory Authentication	Workstation File System Access	Network Server File System Access
Uninstall	Workstation Helper	eDirectory workstation (Workstation object)	Windows user	Not applicable

¹ The Windows 98 operating system, unlike Windows 2000/XP, does not provide file system security for individual users. Each Windows 98 user account has full access to the local file system, which means that Application Launcher and Workstation Helper have all the file system access they require.

² NetWare file system rights can be assigned through the Application object (*Common tab > File Rights* page). Any workstation that is associated with the Application object receives these rights. You can also directly assign rights to workstations through their Workstation objects (*Rights to Files and Folder tab > Trustee File System Rights* page) or some other method, such as adding them to a workstation group that has been assigned the appropriate rights.

³ Windows server file permissions must be assigned through the user's Active Directory account, which must have the same username and password as the user's eDirectory account. The user, Middle Tier Server (if used), and Windows server must be members of the same Windows domain. For information about using Novell DirXML to synchronize user account information between eDirectory and Active Directory, see “[Installing Nsure Identity Manager 2.02 Bundle Edition](#)” in “[Installing in a Windows Network Environment](#)” in the *Novell ZENworks 7 Desktop Management Installation Guide*.

Note that this is different than for a workstation-associated application distributed to a Windows 2000/XP workstation (see “[Windows 2000/XP \(Workstation-Associated Applications\)](#)” on [page 300](#)). Windows 98 does not differentiate between the user and the workstation (system), so the user credentials must be used for login, not the workstation credentials. If access to the Windows server is being provided through the ZENworks Middle Tier Server, the Domain User account used by the Middle Tier Server must be assigned the appropriate permissions.

⁴ Linux server file rights are assigned through Samba. The procedures for doing this depend on the Linux distribution (OES Linux, SUSE Linux Enterprise Server, etc.) being used. For OES Linux and SUSE Linux Enterprise Server, see “[Configuring a Linux Server for ZENworks File Access](#)” in the *Novell ZENworks 7 Desktop Management Installation Guide*. For additional information, refer to the OES Linux and SUSE Linux Enterprise Server documentation available on the [Novell Documentation Web site \(http://www.novell.com/documentation\)](#) or refer to the documentation for your Linux distribution.

⁵ The Force Run setting causes the application to automatically distribute after it becomes available. For information about configuring an application as Force Run, see “[Associations Page](#)” on [page 534](#).

23.4.2 Windows 2000/XP (Workstation-Associated Applications)

The following table lists the components, authentication method, and file system access used by Application Launcher when managing a workstation-associated application on a Windows 2000/XP workstation.

Table 23-4 *Windows 2000/XP (Workstation-Associated Applications)*

Event	Responsible Component	eDirectory Authentication	Workstation File System Access	Network Server File System Access
Distribution	NAL Service	eDirectory workstation (Workstation object)	Windows System user ¹	NetWare: Folder and file rights assigned to eDirectory workstation ² Windows: Permissions assigned to Active Directory workstation ³ Linux: Rights assigned to workstation through Samba ⁴
Launch (normal)	Application Launcher or NAL Service (when run as secure/unsecure System user ⁵)	eDirectory workstation (Workstation object)	Windows user or Windows System user (when run as secure/unsecure System user)	NetWare: Folder and file rights assigned to eDirectory user or folder and file rights assigned to eDirectory workstation (when run as secure/unsecure System user) Windows: Permissions assigned to Active Directory user or permissions assigned to Active Directory workstation (when run as secure/unsecure System user) Linux: Rights assigned to Samba user or rights assigned to workstation (when run as secure/unsecure System user)
Launch (force run ⁶)	NAL Service	eDirectory workstation (Workstation object)	Windows System user	NetWare: Folder and file rights assigned to eDirectory workstation Windows: Permissions assigned to Active Directory workstation Linux: Rights assigned to workstation through Samba
Caching	NAL Service	eDirectory workstation (Workstation object)	Windows System user	NetWare: Folder and file rights assigned to eDirectory workstation Windows: Permissions assigned to Active Directory workstation Linux: Rights assigned to workstation through Samba
Uninstall	NAL Service	eDirectory workstation (Workstation object)	Windows System user	Not applicable

¹ For Application Launcher and its associated programs (NAL Service and Workstation Helper) to work properly, the Windows System user account must have full rights to all areas of the workstation. By default, this access is granted to the System user as a member of the Administrators group. Do not limit the default rights given to the Administrators group or the System user account.

In addition, Application Launcher requires that the Windows user account provide the following rights:

- ◆ At least Read access to the NAL cache directory (typically, `c:\nalcache`). For more information, see [Section 24.2, “File System Rights to the NAL Cache,” on page 306](#).
- ◆ Full Control access to the user's temp directory (typically, `c:\documents and settings\username\local settings\temp`).
- ◆ Full Control access to the user's data encryption directory (typically, `c:\documents and settings\username\application data\microsoft\crypto`). This is required only if the user is using the Desktop Management Agent without a network client.
- ◆ Read\Write rights to the `HKEY_CURRENT_USER\Software\NetWare\NAL\1.0` registry key.
- ◆ Read rights to the `HKEY_LOCAL_MACHINE\Software\NetWare\NAL\1.0` registry key
- ◆ Read rights to the `HKEY_LOCAL_MACHINE\Software\Novell\ZENworks` registry key.

² NetWare server file rights can be assigned through the Application object (*Common* tab > *File Rights* page). Any workstation that is associated with the Application object receives these rights. You can also directly assign rights to workstations through their Workstation objects (*Rights to Files and Folder* tab > *Trustee File System Rights* page) or some other method, such as adding them to a workstation group that has been assigned the appropriate rights.

³ Windows server file permissions must be assigned through the workstation's Active Directory account. The workstation, Middle Tier Server (if used), and Windows server must be members of the same Windows domain. Note that this is different than for a workstation-associated application distributed to a Windows 98 workstation (see [“Windows 98 \(Workstation-Associated Applications\)” on page 299](#)). Windows 98 does not differentiate between the user and the workstation (system), so the user credentials must be used for login, not the workstation credentials. If access to the Windows server is being provided through the ZENworks Middle Tier Server, the Domain User account used by the Middle Tier Server must be assigned the appropriate permissions.

⁴ Linux server file rights are assigned through Samba. The workstation (not the logged-in user) must have rights to read files from the Linux server's Samba share. The procedures for assigning rights depend on the Linux distribution (OES Linux, SUSE Linux Enterprise Server, etc.) being used. For OES Linux and SUSE Linux Enterprise Server, see [“Configuring a Linux Server for ZENworks File Access”](#) in the *Novell ZENworks 7 Desktop Management Installation Guide*. For additional information, refer to the OES Linux and SUSE Linux Enterprise Server documentation available on the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation) or refer to the documentation for your Linux distribution.

⁵ The Secure System User and Unsecure System User settings cause the application to run in the "system" space as the Windows System user rather than in the "user" space as the logged-in user. These settings are intended to ensure that users can run the application even if they have limited access rights to the workstation's file system. For more information, see [“Environment Page” on page 523](#).

⁶ The Force Run setting causes the application to automatically distribute after it becomes available. For information about configuring an application as Force Run, see [“Associations Page” on page 534](#).

Novell Application Launcher: Managing the Cache

24

Novell® Application Launcher™ creates a cache directory, referred to as the NAL cache, on the workstation's Windows system drive (the drive containing the `system32` directory). The NAL cache enables Application Launcher to do the following:

- ◆ Display, launch, install, and fix applications when the user or workstation is not authenticated to Novell eDirectory™.
- ◆ Uninstall applications it has installed to the workstation.
- ◆ Delay reading eDirectory during startup in order to minimize network traffic during peak user login times.
- ◆ Resume downloading of an application's files and information while in remote mode, rather than starting over on the download.
- ◆ Reduce network traffic by minimizing the number of times it accesses eDirectory for application information.

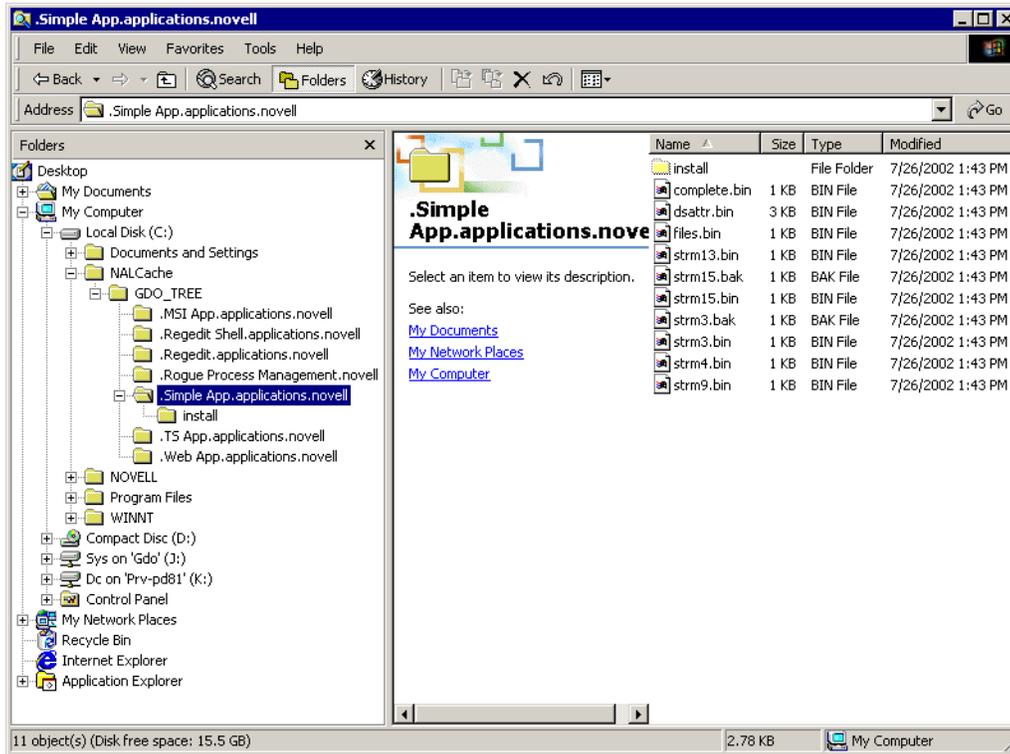
The following sections provide detailed information about the contents of the NAL cache and tasks to help you manage the NAL cache:

- ◆ [Section 24.1, “NAL Cache Structure and Content,” on page 303](#)
- ◆ [Section 24.2, “File System Rights to the NAL Cache,” on page 306](#)
- ◆ [Section 24.3, “Caching an Application,” on page 307](#)
- ◆ [Section 24.4, “Changing the Cache Location,” on page 308](#)
- ◆ [Section 24.5, “Disabling the Cache,” on page 310](#)

24.1 NAL Cache Structure and Content

By default, the NAL cache directory is created as a hidden `\nalcache` directory at the root of the drive, as shown in the following example.

Figure 24-1 NAL Cache Directory on a Local Drive



When an application is associated with the user or workstation, Application Launcher creates a directory for the application under the `\nalcache` directory. The application's directory is named the same as the Application object's eDirectory distinguished name (for example, `Simple App.applications.novell`) and contains the files that make up the launch cache and install cache. These two caches are described in the following sections:

- ◆ [Section 24.1.1, “Launch Cache,” on page 304](#)
- ◆ [Section 24.1.2, “Install Cache,” on page 306](#)

24.1.1 Launch Cache

The files in an application's launch cache contain the information Application Launcher needs to launch, run, and uninstall the application. Application Launcher automatically creates the application's launch cache when the application is distributed to the workstation. If Application Launcher has not yet distributed the application to the workstation, the launch cache contains only the information needed to display the application's icon on the workstation's desktop.

The launch cache files are located directly under the application directory (`\nalcache\edirectory_tree\app_object` directory). The following table describes the launch cache files. Depending on the Application object's settings, some of the files might not be necessary, in which case they are not included in the launch cache.

Table 24-1 *Launch Cache*

File	Description
complete.bin	Contains the Application object's complete eDirectory name. If this file does not exist, Application Launcher assumes that the cache is bad and recreates the cache.
dsattr.bin	Contains all of the Application object's eDirectory attributes.
delta.bin	Flags the cache as a delta cache. This means the cache contains only the information required by Application Launcher to display the application's icon on the desktop, and exists only if Application Launcher has not yet distributed the application to the workstation. After Application Launcher distributes the application, it removes this file.
files.bin	Contains the file vitals for the cache files.
folders.bin	Contains the list of Application Launcher window folders and Start menu folders where the Application object displays.
strm1.bin	Contains the Application object's icon.
strm2.bin	Contains all text file modifications.
strm3.bin ¹	Contains all files to be copied to the workstation.
strm4.bin ¹	Contains registry modifications that are made during initial distribution.
strm5.bin ¹	Contains INI modifications that are made during initial distribution.
strm6.bin ¹	Contains shortcut modifications that are made during initial distribution.
strm7.bin	Contains macro information.
strm8.bin	Contains schedule information.
strm9.bin	Contains system requirements information.
strm10.bin	Contains Administrator notes.
strm11.bin	Contains the shutdown script.
strm12.bin	Contains the startup script.
strm13.bin ¹	Contains registry modifications to be made each time the application is launched.
strm14.bin ¹	Contains INI modifications to be made each time the application is launched.
strm15.bin ¹	Contains files to be copied to the workstation each time the application is launched.
strm16.bin	Contains text file modifications to be made each time the application is launched.
strm17.bin ¹	Contains icons to be modified each time the application is launched.
strm18.bin	Contains environment variable settings to be made.
strm19.bin	Contains the pre-distribution script.
strm20.bin	Contains the post-distribution script.

File	Description
strm21.bin	Contains the pre-install schedule information.

¹ All files except those containing application files, INI settings, shortcut icons, and registry settings are refreshed each time Application Launcher performs a timed or manual refresh. The non-refreshed files remain the same until the version number (Application object > *Distribution Options* tab > *Options* page) is changed. This allows Application Launcher to successfully uninstall the application (including the files, INI settings, shortcuts, and registry settings) if necessary.

24.1.2 Install Cache

The install cache contains the application source files needed to install or verify (fix) the application. Application Launcher creates an install cache for an application in the following two situations:

- The application is marked as Force Cache. Application Launcher retains the install cache for a Force Cache application as long as the application is associated with the user or workstation.
- The application is not marked as Force Cache, but Application Launcher is running in Remote Mode and the application is launched. In this case, Application Launcher first copies the application to the install cache and then distributes it (which populates the launch cache). After the application is launched, Application Launcher removes its install cache.

The install cache is created in the `nalcache\edirectory_tree\app_object\install` directory.

Although the files in the install cache are compressed, they can still require a large amount of disk space on the workstation's local drive, so an application should only be cached if users need to install or verify the application while disconnected from eDirectory.

An application can still run in disconnected mode without being cached to the install cache. To run in disconnected mode, an application needs only to have been distributed (installed) on the workstation. Caching an application to the install cache only enables the application to be installed or verified (fixed) while in disconnected mode.

24.2 File System Rights to the NAL Cache

Prior to ZENworks 6.5 Support Pack 1 (SP1), the ZENworks Desktop Management Agent installation program created the NAL cache directory (typically, `c:\nalcache`) and automatically assigned the user Full Control access.

As a result, if you installed the Desktop Management Agent on a workstation, even a locked-down workstation, the installation program created the NAL cache directory and assigned Full Control access, which allowed Novell Application Launcher to access the NAL cache without further modifications to file system rights.

Beginning with ZENworks 6.5 SP1, the ZENworks Desktop Management Agent installation program no longer assigns the user any file system rights to the NAL cache directory (typically, `c:\nalcache`). Instead, the NAL cache directory inherits whatever rights are assigned to the root of the drive, assuming you use the default location (`c:\nalcache`). In order for users to be able to access the NAL cache, they need at least Read rights to the NAL cache directory.

As a result of this change, consider the following:

- ◆ When you install the ZENworks 7 Desktop Management Agent on workstations that had the ZENworks 6.5 (or earlier) Desktop Management Agent installed at one time, you should not experience any NAL cache file system access problems. The original installation created the NAL cache directory with Full Control rights, and the upgrade continues using these same rights.
- ◆ When you install the ZENworks 7 Desktop Management Agent on a workstation that has not had the ZENworks 6.5 (or earlier) Desktop Management Agent installed at one time, the installation program does not assign any file system rights to the NAL cache directory; it inherits whatever rights are assigned to the root of the drive.

As a result, users might experience problems accessing the NAL cache on locked-down workstations. For example, if users do not have at least Read rights to the root of the drive, they cannot access the NAL cache directory and Novell Application Launcher cannot use the cache.

- ◆ If you manually created the NAL cache directory and assigned rights before installing the ZENworks 7 Desktop Management Agent or if you changed rights to an existing directory, these rights remain in effect after installation. Be aware that the user needs at least Read rights to the NAL cache, or Novell Application Launcher cannot use the cache.

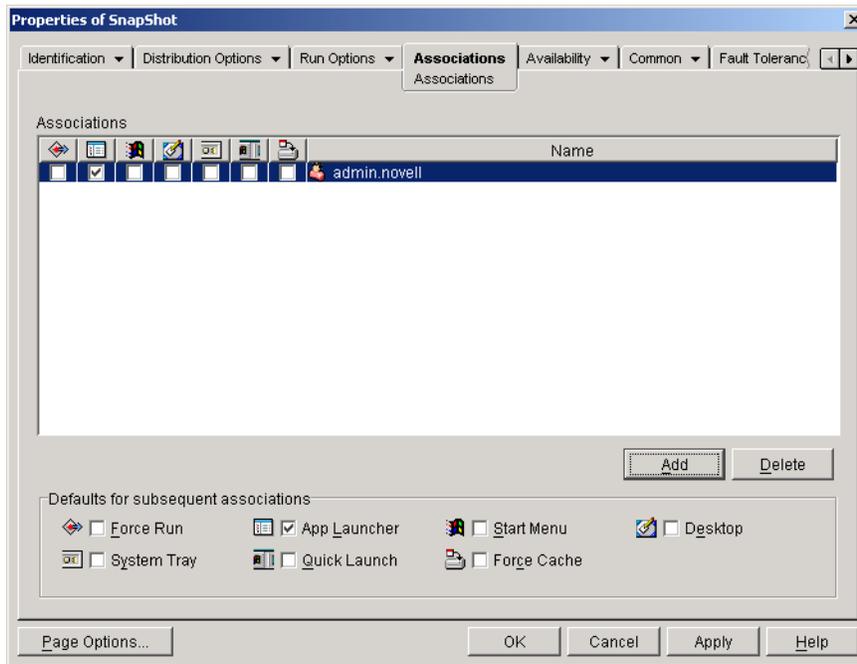
24.3 Caching an Application

Applications are not cached by default. If you want to enable a user to install or verify an application while disconnected from eDirectory, you must configure the application to be cached.

- 1 In ConsoleOne[®], right-click the Application object, then click *Properties*.
- 2 Click the *Identification* tab to display the Icon page.
- 3 Make sure the *Disconnectable* box is selected.

In order to cache an application, it must be marked as Disconnectable. If the application is not marked as Disconnectable, the Force Cache option is not available.

- 4 Click the *Associations* tab to display the Associations page.



- 5 In the Associations list, select the *Force cache* box for the user whose workstation you want to cache the application to.

NOTE: If the application is marked as Distribute Always (Distribution Options tab > Options page) and Application Launcher is running in remote mode, Application Launcher behaves as if the Force Cache and Check Point Restart options are enabled regardless of their actual settings.

- 6 Click *OK* to save the changes.

The application is cached to the user's local drive the next time Application Launcher restarts or refreshes.

The same task can be accomplished through the *ZENworks* tab > *Applications* page on a User, Workstation, Group, Workstation Group, or container object.

24.4 Changing the Cache Location

As mentioned in [Section 24.1, "NAL Cache Structure and Content," on page 303](#), the default cache is created as a hidden directory (`\nalcache`) on the root of the workstation's Windows drive.

You can change the cache location by modify the workstation's registry. The next time Application Launcher refreshes (either a timed refresh, a manual refresh, or a restart), the cache is created in the new location. The old cache directory is not removed automatically.

To change the location of the cache directory:

- 1 Use `regedit.exe` to open the Windows registry.
- 2 Locate the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\NetWare\NAL\1.0
```

- 3 Add a new String value to the key using the following information:

Value type: String

Value name: MasterCache

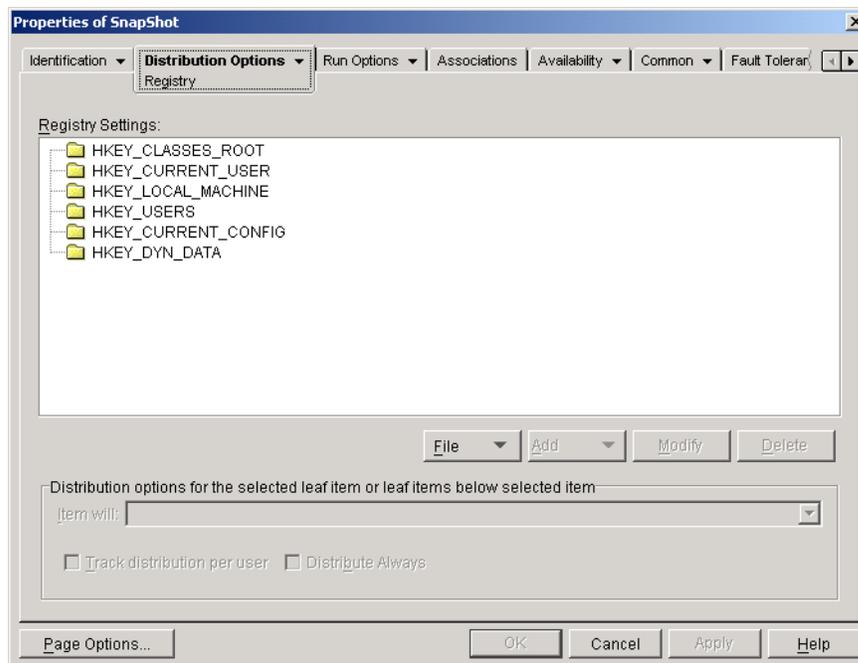
Value data: Specify the directory you want to use for the cache. To avoid having the cache files and subdirectories mixed in with other files and directories, you might want to specify an empty directory. For example, if you want the cache located under the `c:\novell` directory, you might want to specify `c:\novell\nalcache` rather than just `c:\novell`. Do not use macros in this field; macros are not supported.

- 4 Save the registry changes.
- 5 Refresh or restart Application Launcher.

Using Application Launcher to Change the Cache Location

You can also use Application Launcher to change the cache location. To do so:

- 1 In ConsoleOne, create a simple Application object. For instructions, see [Chapter 28, “Distribution: Simple Applications,”](#) on page 325.
- 2 Right-click the Application object, then click *Properties* to display the object's property pages.
- 3 Click *Distributions Options > Registry* to display the Registry page.



- 4 Add the following registry key:

`HKEY_LOCAL_MACHINE\SOFTWARE\NetWare\NAL\1.0`

- 5 Add a new String value to the key using the following information:

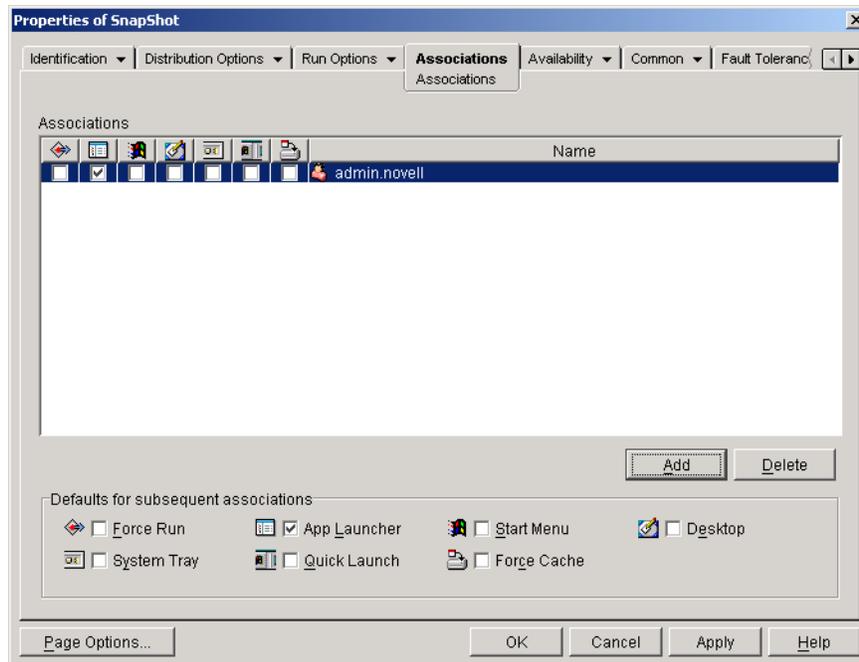
Value type: String

Value name: MasterCache

Value data: Specify the directory you want to use for the cache. To avoid having the cache files and subdirectories mixed in with other files and directories, you might want to specify an

empty directory. For example, if you want the cache located under the `c:\novell` directory, you might want to specify `c:\novell\nalcache` rather than just `c:\novell`. Do not use macros in this field; macros are not supported.

- 6 Click *Associations* to display the Associations page.



- 7 Add the users who you want the changes distributed to.
- 8 For each user association, select the *Force run* option to have the change made without requiring user intervention.
- 9 Click *OK* to save the information.

24.5 Disabling the Cache

You can disable the cache by turning off Application Launcher's ability to write application information to it. If you choose to disable the cache, please be aware of the following:

- ♦ The primary purpose of the cache is to enable users to work offline, disconnected from eDirectory. Disabling the cache forces users to be connected to eDirectory or a removable cache in order to access applications. To ensure that users don't manually go offline, the Work Offline feature in Application Launcher is disabled when the cache is disabled.
- ♦ The uninstall, random refresh, and checkpoint restart features do not work. These features rely on Application Launcher having access to the cache.

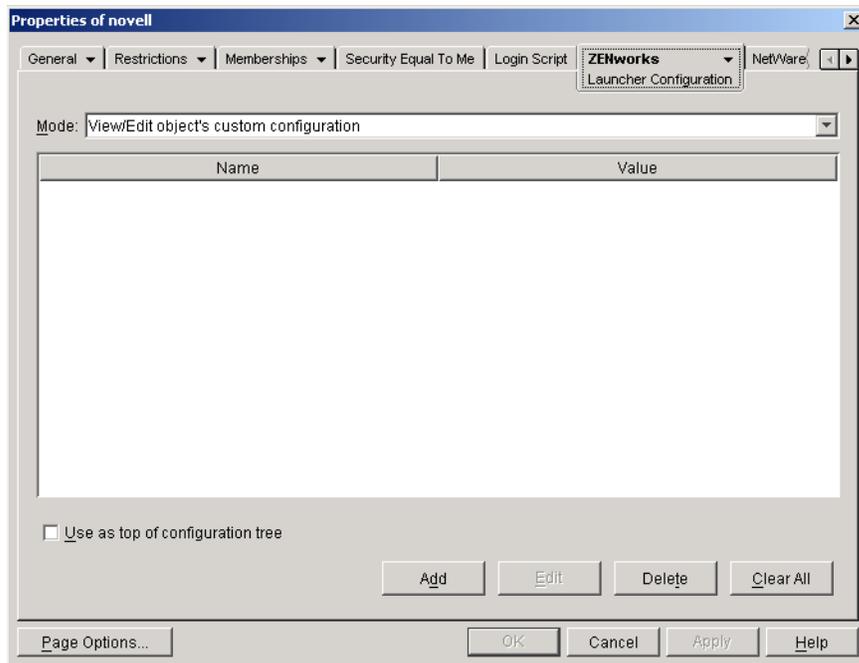
To disable the cache:

- 1 In ConsoleOne, select a container object if you want to disable the cache for all users in the container.

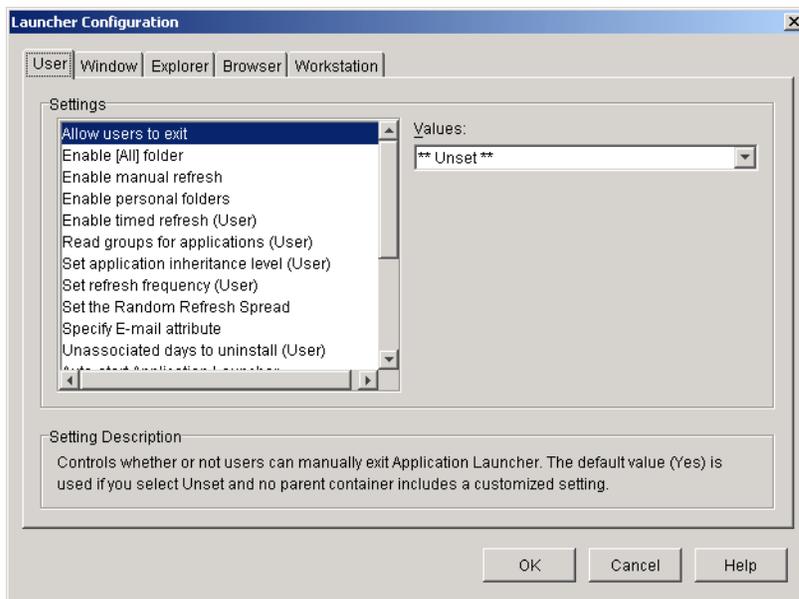
or

Select a User object if you want to disable the cache for an individual user.

- 2 Right-click the object, then click *Properties* to display the object's property pages.
- 3 Click the *ZENworks* tab, then click *Launcher Configuration* to display the Launcher Configuration page.



- 4 Click *Add* to display the Launcher Configuration Settings page.



- 5 In the Settings list on the User or Workstation tab, select the *Enable writing to the cache* option.

NOTE: The *Enable writing to the cache* option is available on both the User and Workstation tabs. You should use the appropriate tab, depending on whether you want to disable writing to the cache for user-associated applications or for workstation-associated applications. If you

want to disable the cache for both user-associated and workstation-associated applications, you should perform the steps in this section on both the User and Workstation tabs.

6 In the *Values* field, select *No*.

7 Click *OK* to save the change.

The change is applied the next time Application Launcher restarts or refreshes.

Novell Application Launcher: Organizing Applications

25

Novell® ZENworks® Desktop Management enables you to organize Application objects into folders. Folders show up in the Novell Application Launcher™ windows (Application Window, Application Explorer window, and Application Browser window) and on the Windows Start menu.

The following sections provide information about the different types of folders available for organizing applications, instructions for creating an Application Folder object, and instructions for adding Application objects to application folders or custom folders:

- ♦ [Section 25.1, “Types of Folders,” on page 313](#)
- ♦ [Section 25.2, “Creating and Deleting Application Folders,” on page 314](#)
- ♦ [Section 25.3, “Adding Applications to Application Folders,” on page 315](#)
- ♦ [Section 25.4, “Adding Applications to Custom Folders,” on page 317](#)

25.1 Types of Folders

There are four types of folders used to organize applications:

- ♦ **Application folder (administrator created):** An application folder is created as a Novell eDirectory™ object. A single Application Folder object can consist of one folder or multiple folders (a multi-level folder structure). After you create an Application Folder object, you can add any number of Application objects to the folder (or folders). Application Folder objects are the recommended folder type if you want to maintain consistent folders from user to user. For information about creating Application Folder objects, see [Section 25.2, “Creating and Deleting Application Folders,” on page 314](#). For information about adding applications to an application folder, see [Section 25.3, “Adding Applications to Application Folders,” on page 315](#).
- ♦ **Custom folder (administrator created):** A custom folder is created specifically for a single Application object in eDirectory. No other Application objects can be included in the folder. Custom folders support folders within folders, which means that you can create a custom folder structure. For example, although you could not have Calculator and Notepad in the same custom folder, you could create two subfolders within the same custom folder and place the two programs in the two subfolders (in other words, `winapps\calculator\calc.exe` and `winapps\notepad\notepad.exe`). For information about creating custom folders for an application, see [Section 25.4, “Adding Applications to Custom Folders,” on page 317](#).
- ♦ **Personal folder (user created):** A personal folder is created in Application Launcher by individual users. Users can move Application objects from other folders into the personal folder. By default, the ability to create personal folders is disabled. For information about enabling this feature, see [“Novell Application Launcher: Configuring Settings” on page 265](#).
- ♦ **System folder (system created):** A system folder is automatically created by Application Launcher to store an Application object if the object has not been added to an application folder or a custom folder.

If the Application object is associated with a user, the Application object's system folder is given the name specified in the User object's Full Name field (for example, John Smith). If the

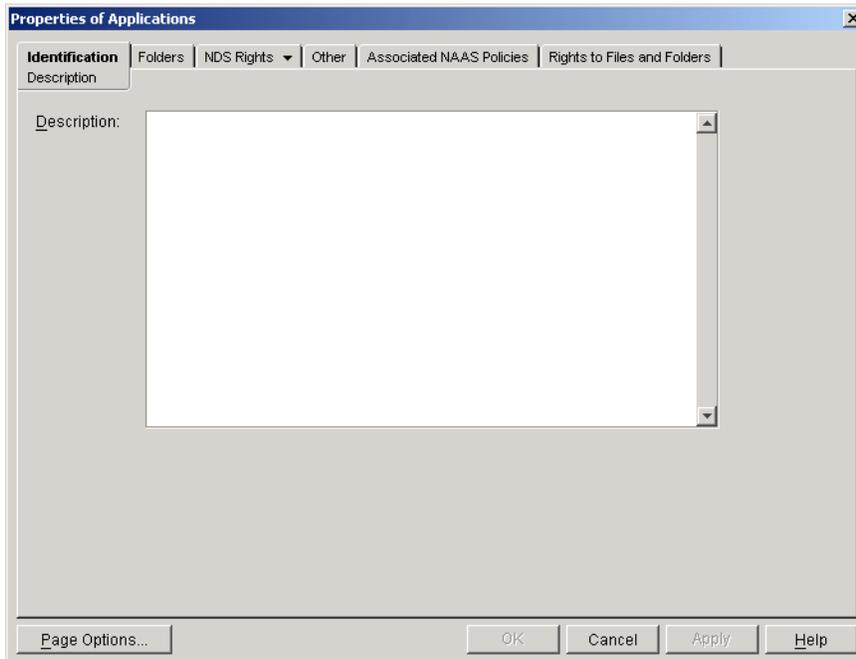
Application object is associated with another eDirectory object (Group, Workstation, container, and so forth), the system folder is given the name specified in the object's Description field (for example, System Test Group). If nothing is specified in the Full Name or Description field, the system folder is given the object's fully distinguished name. For example, an Application object associated with a user through his or her User object would be placed in a system folder that has the user's fully distinguished name, but an Application object associated with a user through a group would be placed in a folder that has the group's fully distinguished name.

25.2 Creating and Deleting Application Folders

- 1 In ConsoleOne[®], right-click the container where you want to create the Application Folder object, click *New*, then click *Object* to display the New Object dialog box.
- 2 Select *Application folder*, then click *OK* to display the New App:Folders dialog box.

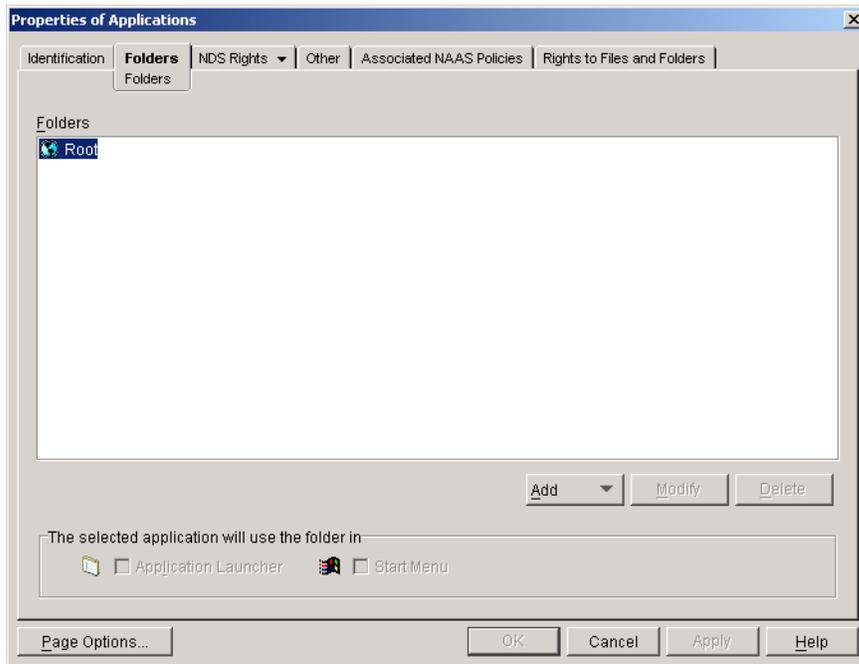


- 3 In the *Name* field, specify a name for the object. The name is displayed only in eDirectory and does not need to be the same name as the one you want to appear in the Application Launcher window and on the Start menu on the users' workstations.
- 4 Select the *Define additional properties* check box, then click *OK* to display the Application Folder object's Description page.



- 5 If desired, enter a description in the Description box.

6 Click the *Folders* tab to display the Folders page.



The Folders tree contains one entry called Root. Root represents the Windows Start menu and the Application Launcher window.

7 Use the *Add*, *Modify*, and *Delete* options to create the folder structure you want.

Add: To add a folder, select the folder (or Root) where you want to create a new folder, click *Add*, click *Folder*, type the name for the new folder, then press Enter.

The following characters are invalid when used in Windows folder and file names:

\ / : * ? " < > |

If you use any of these characters in the folder name, they are replaced by an underscore (_).

Modify: To change the name of a folder, select the folder in the Folders tree, click *Modify*, type the new folder name, then press Enter.

Delete: To remove a folder, select the folder in the Folders tree, then click *Delete*. The change is applied the next time Application Launcher restarts.

8 Click *OK* when you are finished creating the folder structure.

9 To add applications to your folders, continue with [Section 25.3, “Adding Applications to Application Folders,”](#) on page 315.

25.3 Adding Applications to Application Folders

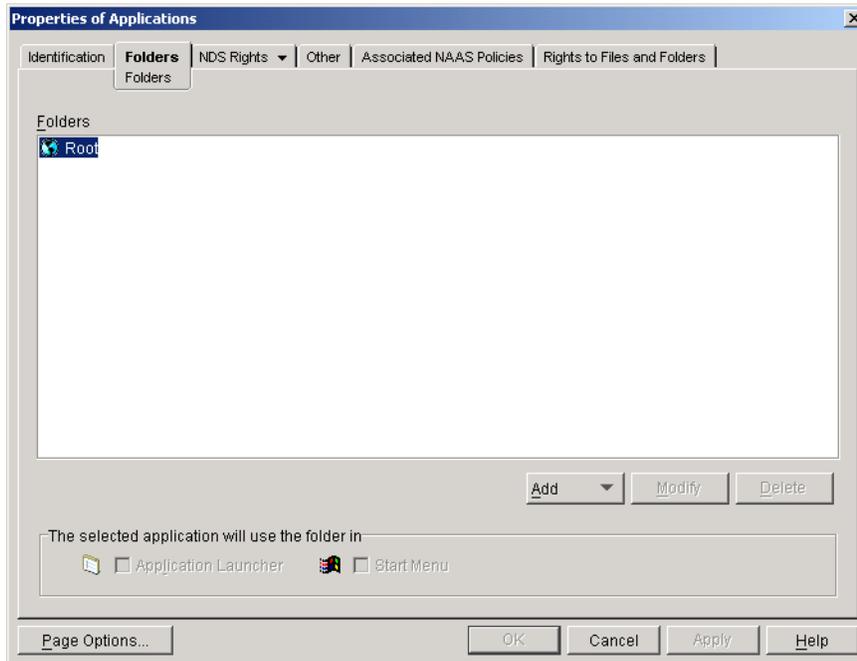
There are two ways you can add Application objects to application folders. You can add Application objects through the Folders page on the Application Folder object, or you can link individual Application objects through the Folders page on the Application object. Using the Application Folder object enables you to add multiple Application objects to the application folder at one time. Using the Application object enables you to add the single Application object only.

- ◆ [Section 25.3.1, “Using the Application Folder Object to Add Applications,”](#) on page 316

- ♦ [Section 25.3.2, “Using the Application Object to Add the Application to a Folder,”](#) on page 316

25.3.1 Using the Application Folder Object to Add Applications

- 1 In ConsoleOne, right-click the Application Folder object you want to add the Application object to, then click *Properties*.
- 2 Click the *Folders* tab to display the Folders page.

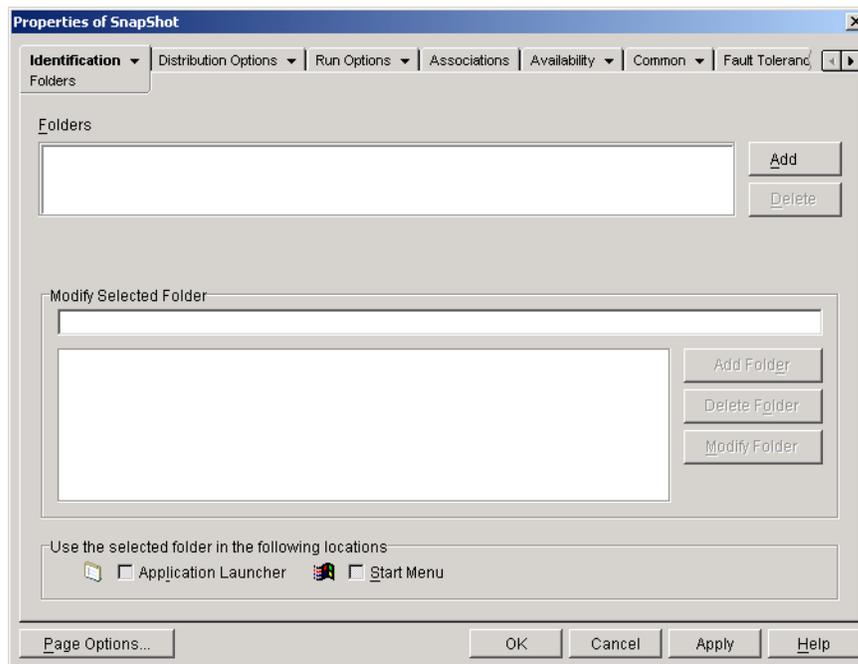


- 3 In the Folders tree, select the folder where you want to add the Application object.
- 4 Click *Add*, click *Application*, browse for and select the desired Application object, then click *OK* to add it to the folder in the folders list.
- 5 When finished adding Application objects, click *OK* to save your changes.

25.3.2 Using the Application Object to Add the Application to a Folder

- 1 In ConsoleOne, right-click the Application object you want to add to a folder, then click *Properties*.

- 2 Click the *Identification* tab > *Folders* to display the Folders page.



- 3 Click the *Add* button > *Linked folder* to display the Select Object dialog box.
- 4 Browse for and select the Application Folder object you want to add the Application object to, then click *OK* to display the Folder Object Structure dialog box.
- 5 Select the folder you want to add the Application object to, then click *OK* to add it to the Folders list.

If you select the Root folder, the Application object is displayed directly on the Start menu and at the root of the Application Launcher window.

By default, the Application object is displayed in the selected folder on the Start menu and the Application Launcher window.

- 6 In the Use the Selected Folder in the Following Locations box, deselect Application Launcher if you don't want the Application object included in that folder in the Application Launcher window and deselect Start Menu if you don't want the Application object included in that folder on the Windows Start menu.

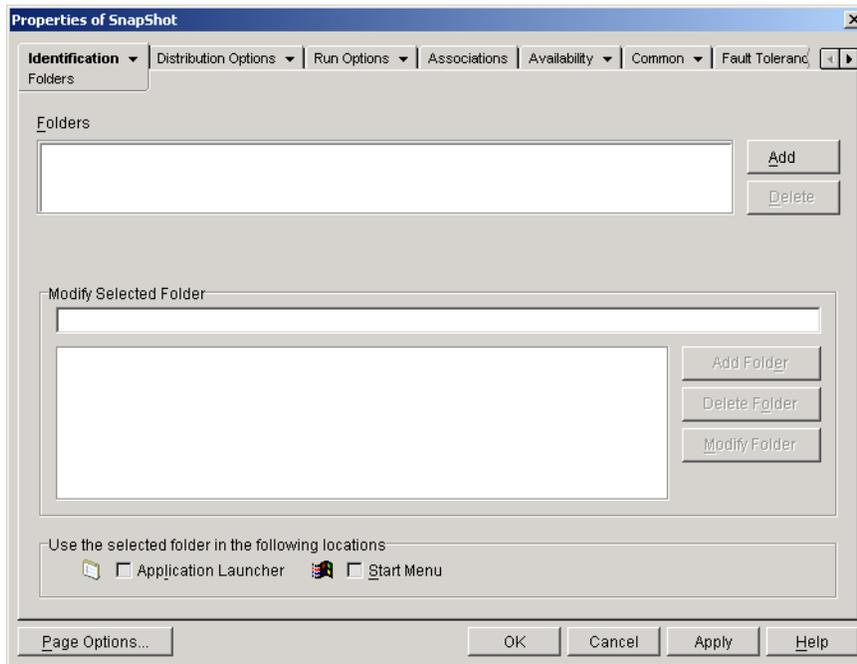
A folder displays in the Application Launcher window and on the Start menu only if it contains an Application object. If no Application objects have been linked to the folder, it is not displayed.

- 7 Click *OK* to save the changes.

25.4 Adding Applications to Custom Folders

- 1 In ConsoleOne, right-click the Application object you want to add to a folder, then click *Properties*.

- 2 Click the *Identification* tab > *Folders* to display the Folders page.



- 3 Click the *Add* button > *Custom folder* to add a New Folder entry to the Folders list.
- 4 In the Modify Selected Folder list, select the New Folder entry, click *Modify Folder*, type a name for the folder, then press Enter.
- 5 If you want to add subfolders, select the folder, click *Add Folder*, type a name for the subfolder, then press Enter.
When you add subfolders, the Application object is displayed in the lowest-level folder.
- 6 Click *OK* to save the changes.

ZENworks Launch Gadget: Configuring Settings

26

Novell® ZENworks® Desktop Management includes the ZENworks Launch gadget that can be used to launch user-associated applications from within a Novell exteNd Director™ 4.1 SE portal. If you have not installed the Launch gadget but are interested in using it, see “[Installing the ZENworks Launch Gadget](#)” in the *Novell ZENworks 7 Desktop Management Installation Guide*.

The Launch gadget includes several settings you can use to configure how the gadget works. To configure the settings:

- ♦ [Section 26.1, “Modifying Gadget Configuration Settings,” on page 319](#)
- ♦ [Section 26.2, “Adding Citrix MetaFrame Servers to the Portal,” on page 321](#)

26.1 Modifying Gadget Configuration Settings

Within the exteNd Director portal administration utility, there are two places that you can modify the Launch gadget configuration settings: the ZENworks Applications page and the ZENworks Launch gadget object.

- ♦ **ZENworks Applications page:** The Launch gadget installation creates a ZENworks Applications page and assigns the Launch gadget (OD_ZenLaunchGadget) to the page. You can modify the gadget configuration settings for the ZENworks Application page, the modifications apply only to the gadget on that page. If you assign the gadget to another page, the modifications are not applied to that page. For instructions, see “[Configuring Gadget Settings on the ZENworks Application Page](#)” on page 319.
- ♦ **Launch gadget object:** You can modify configuration settings on the Launch gadget object if you want the settings applied to all pages the gadget is assigned to. For instructions, see “[Configuring Gadget Settings on the Launch Gadget Object](#)” on page 320.

26.1.1 Configuring Gadget Settings on the ZENworks Application Page

- 1 Log in to the portal as a portal administrator.
- 2 Click *Administer the portal*.
- 3 Click *Pages*.
- 4 Select the ZENworks Application page, then click *Edit*.
- 5 In the gadget assignments list, select *ZENworks applications*, then click *Edit* to display the Gadget Assignment Configuration page.
- 6 Modify the configuration settings. For a description of each setting, see “[ZENworks Launch Gadget \(OD_ZENLaunchGadget\) Settings](#)” on page 320.

26.1.2 Configuring Gadget Settings on the Launch Gadget Object

- 1 Log in to the portal as a portal administrator.
- 2 Click *Administer the portal*.
- 3 Click *Gadgets*.
- 4 Select *OD_ZENLaunchGadget*, then click *Edit*.
- 5 Click *Configuration* to display the Gadget Configuration page.
- 6 Modify the configuration settings. For a description of each setting, see “ZENworks Launch Gadget (OD_ZENLaunchGadget) Settings” on page 320

26.1.3 ZENworks Launch Gadget (OD_ZENLaunchGadget) Settings

The Launch gadget has the following configuration settings:

Display name: Specifies the name used when displaying the gadget on the portal page. This field is available only if you are editing a gadget assignment rather than the gadget object.

Window height: Specifies the height of the window that lists the applications. If the application list is too long to fit in the window, the window provides a vertical scroll bar. The default setting is 198 pixels.

Novell Client authentication host server: Specifies the IP address or DNS host name of the server to be used when authenticating to eDirectory through the Novell Client™. This setting overrides (and changes) the user's current Novell Client configuration setting.

XTier authentication host server: Specifies the IP address or DNS host name of the server where the ZENworks Middle Tier server is installed. The Middle Tier servers enables access to Novell eDirectory without the Novell Client. This setting overrides (and changes) the user's current Middle Tier server configuration setting.

XTier port: Specifies the number of the Middle Tier server. This setting overrides (and changes) the user's current Middle Tier server configuration setting.

Display thin-client applications: Specifies whether or not to display terminal server applications in the list. The default is *TRUE*.

Display workstation installed applications: Specifies whether or not to display desktop applications in the list. The default is *TRUE*.

Display large icons: Specifies whether application icons are displayed in the Windows small icon format or in large icon format. The default is *FALSE*, which means small icon format is used.

Number of columns: This setting applies only if *Display large icons* is set to *TRUE*. It specifies the number of columns to use when displaying the icons. The default is 4.

ICA template file: Specifies the file that contains default initialization information for the Citrix ICA Client program. The settings in the file are overwritten with information from IMA database, *NFuse.conf*, and gadget settings. The default file is *OnDemandTemplate.ica*, located in the *tomcat\webapps\nps\portal\gadgets\com.novell.ondemand.gadgets.ZenLaunchGadget* directory.

ICA window type: Specifies the type of window used by the ICA client when launching applications. The default is seamless. Other possible values are: percent, pixels, fullscreen. If you specify percent, you must use the **ICA Percent** setting to specify the size. If you specify pixels, you must use the **ICA Width** and **ICA Height** settings to specify the size.

ICA width: This setting applies only when ICA Window Type is set to pixels. Specify the width of the ICA client window in pixels. The default is 640 pixels.

ICA height: This setting applies only when ICA Window Type is set to pixels. Specify the height of the ICA client window in pixels. The default is 480 pixels.

ICA percent: This setting applies only when ICA Window Type is set to percent. Specify the percentage of the user's screen to be used by the ICA client window. The default is 25 percent.

ICA color depth: Specifies the color settings used by the ICA client when launching applications. Possible values are:

- ◆ 1 — 16 colors
- ◆ 2 — 256 colors
- ◆ 4 — high color (16-bit)
- ◆ 8 — true color (24 -bit)

The default is 2.

ICA audio: Specifies whether ICA client support for audio is turned on or off. The default is off.

Production context attribute: This setting applies only in an environment with multiple eDirectory trees. Typically, in this type of environment, one tree is used for authentication and is synchronized with the production tree via Nsure™ Identity Manager.

In order for the Launch gadget to authenticate to the production tree, you must use the ConsoleOne® Schema Manager to create a new attribute (for example ProductionContext) and assign it to the User object (class). Create the attribute as a single valued string. After you create and assign it to the User object, set the attribute's value to the base context of the users in the production tree, formatted as Typeless Dot notation. For example: users.department.company.

After setting up the attribute in ConsoleOne, use this setting to specify the attribute name (for example, ProductionContext).

26.2 Adding Citrix MetaFrame Servers to the Portal

During installation of the ZENworks Launch gadget to a portal, you specified the DNS name or IP address of the server where the Citrix IMA database resides. If you have multiple Citrix farms, you should have specified addresses of each server where an IMA database resides.

If you need to define additional servers, you can do so through the portal configuration:

- 1 Log in to the portal as a portal administrator.
- 2 Click *Administer the portal*.
- 3 Click *Portal* to display the Edit Portal page.
- 4 Under the Configuration heading, click *Configuration* to display the Portal Configuration page.

- 5** In the *XML_SERVER_PORT* field, add the DNS names or IP addresses of the new servers, using a semicolon (;) to separate each address. For example:

123.456.79.90;123.456.78.91;123.456.78.92

- 6** Click *Save* to save your changes.

Novell® ZENworks® Desktop Management lets you distribute applications to users on Windows 98 and Windows 2000/XP workstations.

The distribution process for an application might be as simple as creating a shortcut to an already installed network application, Web application, or terminal server application, or it might be as complex as installing the application files on the workstation, modifying the workstation's registry and configuration settings, and mapping drive paths. The following sections provide basic information you should understand before configuring applications for distribution:

- ♦ [Section 27.1, “Application Types,” on page 323](#)
- ♦ [Section 27.2, “User Associated vs. Workstation Associated,” on page 323](#)

After you've familiarized yourself with the information in these sections, you should refer to the following for detailed instructions about configuring applications for distribution:

- ♦ [Chapter 28, “Distribution: Simple Applications,” on page 325](#)
- ♦ [Chapter 29, “Distribution: Complex Applications,” on page 331](#)
- ♦ [Chapter 30, “Distribution: Terminal Server Applications,” on page 341](#)
- ♦ [Chapter 31, “Distribution: Web Applications,” on page 347](#)

27.1 Application Types

Applications you distribute are categorized into four types:

- ♦ **Simple application:** Distribution requires Novell Application Launcher™ to copy no (or few) files to the workstation and make no (or few) changes to the workstation's registry, INI files, environment variables, and so forth. An example is Windows Notepad.
- ♦ **Complex application:** Distribution requires Application Launcher to copy many files and make many configuration changes to the workstation. This type of application also includes (or uses) .msi files. An example is Novell GroupWise® or Microsoft Office.
- ♦ **Web application:** Distribution requires Application Launcher to display an application icon that, when launched, opens the workstation's Web browser and displays the Web-based application (or other Web content). No files need to be copied to the workstation and no configuration settings are modified.
- ♦ **Terminal server application:** Distribution requires Application Launcher to display an application icon that, when launched, opens a client session to the terminal server and starts the application (or displays the desktop). An example is Novell GroupWise or Microsoft Office running on a terminal server rather than the user's local workstation.

27.2 User Associated vs. Workstation Associated

When configuring an application for distribution, you can associate the application with both users and workstations.

With a user-associated application, the application is available to the user regardless of the workstation from which the user logs in to eDirectory.

With a workstation-associated application, the application is available on that workstation only. If the user moves to another workstation that is not associated with the application, the application is not available to the user.

If you choose to associate applications with workstations, you should be aware of the following:

- ◆ Each workstation associated with applications must be imported into eDirectory™ as a Workstation object. For details about importing workstations, see [Part III, “Automatic Workstation Import and Removal,” on page 125](#).
- ◆ If you associate an application with a Windows 98 workstation, any users of that workstation must be added as trustees of the Application object. Unlike Windows 2000/XP, Windows 98 does not differentiate between the user and workstation, so Application Launcher always uses the user's eDirectory credentials when distributing or launching workstation-associated applications on a Windows 98 workstation. Trustee rights are not granted automatically; you must use ConsoleOne® to manually give default trustee rights to each user who runs the application on the workstation. If you don't assign trustee rights to the user, the distribution or launch fails.
- ◆ The Application Launcher Workstation Helper, running under Workstation Manager, is responsible for authenticating the workstation to eDirectory and retrieving information about workstation-associated applications. However, the Workstation Helper does not have a user interface. For users to see workstation-associated applications, they must run Application Launcher, but they do not need to be logged in to eDirectory. The Workstation Helper passes the list of workstation-associated applications to Application Launcher, which then displays the applications.
- ◆ Application Launcher displays applications that are associated with the workstation and applications that are associated with the user logged in to eDirectory. This enables you to configure applications you want on the workstation regardless of the user who logs in, while still providing individual users with access to their specific applications.

Distribution: Simple Applications

28

Novell® ZENworks® Desktop Management lets you distribute simple applications to users on Windows 98 and Windows 2000/XP workstations.

A simple application requires no (or few) files to be copied to the workstation and no (or few) changes to the workstation's registry, INI files, environment variables, and so forth.

The following table gives some examples of simple application distributions.

Table 28-1 *Simple Application Distributions*

Example	Distribution Requirement
New application run from user's workstation	Copy an application, consisting of three files, to the user's workstation; no workstation configuration settings need to be changed.
Existing application run from user's workstation	Make an existing application (for example, Windows Notepad) accessible through Application Launcher.
Software patch	Copy a software patch to the user's workstation and automatically apply it.
Database application run from the network	Make a network database application available on the user's workstation. No files need to be copied to the workstation; it simply requires a shortcut that points to the database application's executable.

The following sections help you distribute a simple applications:

- ◆ [Section 28.1, “Preparing a Simple Application,” on page 325](#)
- ◆ [Section 28.2, “Configuring the Application in eDirectory,” on page 326](#)
- ◆ [Section 28.3, “Establishing File System Access,” on page 329](#)
- ◆ [Section 28.4, “Establishing eDirectory Rights,” on page 329](#)
- ◆ [Section 28.5, “What's Next,” on page 329](#)

28.1 Preparing a Simple Application

In general, the following rules apply when preparing a simple application for distribution:

- ◆ If the distribution requires Application Launcher to copy files to the workstation, place the files on a NetWare®, Linux, or Windows server that is available to Application Launcher. For more information about what is required for Application Launcher to access a network resource, see [“Novell Application Launcher: Managing Authentication and File System Access” on page 293](#).
- ◆ If the application runs from the network, make sure the application is installed on a NetWare, Linux, or Windows server that is available to Application Launcher. For more information about what is required for Application Launcher to access a network resource, see [“Novell Application Launcher: Managing Authentication and File System Access” on page 293](#).

After creating the eDirectory object for the simple application (see [Section 28.2, “Configuring the Application in eDirectory,” on page 326](#)), you can configure the Application object so that Application Launcher copies the files to the workstation, creates the shortcut to the application executable, or does whatever else is needed to properly distribute the application. If necessary, you can also configure the Application object to modify any workstation settings (registry settings, INI settings, and so forth) required by the application.

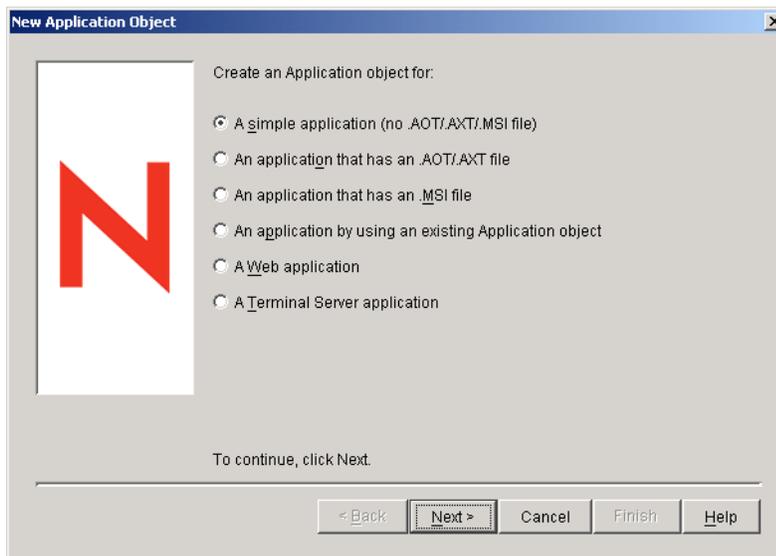
28.2 Configuring the Application in eDirectory

After you have prepared the application for distribution (see [Section 28.1, “Preparing a Simple Application,” on page 325](#)), you are ready to create the application as an Application object in eDirectory™, define its distribution rules, and associate it with users and workstations.

You can create the Application object in any container you want. Because Application Launcher accesses the object on behalf of the logged-in user or workstation, you should place it in a container whose partition (or a replica of the partition) is available to the user or workstation over a local area connection rather than a wide area connection. For more information, see [Chapter 51, “Reference: Application Object Location,” on page 617](#).

To create the Application object in eDirectory and configure it:

- 1 In ConsoleOne®, right-click the container where you want to create the Application object, click *New*, then click *Application* to display the New Application Object dialog box.



- 2 In the New Application Object dialog box, select the *Simple application (No .AOT/.AXT/.MSI file)* option, then click *Next*.
- 3 In the Object Name field, type a name for the Application object.

The Application object’s name must conform to the following rules:

- ♦ The name must be unique in the container.
- ♦ Special characters are allowed. However, plus (+), equals (=), and period (.) must be preceded by a backslash (\) if used.

- ◆ Uppercase and lowercase letters, as well as underscores and spaces, are displayed as you first entered them, but they aren't distinguished. For example, ZENworks_Desktop_Management and ZENWORKS DESKTOP MANAGEMENT are considered identical.

The Application object's name is visible in eDirectory. By default, the name is also used as the Application object's icon title when displayed by Application Launcher on a user's workstation. You can, if necessary, change the icon title after the Application object has been created (Application object > Identification tab > Icon page).

- 4 Click *Next* to display the Path to File field, then enter the path to the location from which the application's executable file runs.

As you specify the path, be aware of the following:

- ◆ Include the executable file in the path.
 - ◆ If, as part of the distribution process, Application Launcher copies the executable file to a target directory on the workstation, specify that location as the path.
 - ◆ For Application Launcher to launch an application from a network server, the workstation must have the appropriate network client (Novell Client™ or Client for Microsoft Networks) installed. This is because the ZENworks Desktop Management Agent and Middle Tier Server do not support opening or executing of files from a network server; only copying of files is supported. In a clientless environment, the alternative to launching the application from a network server is to have Application Launcher first copy the application files to the local workstation and then launch the application locally. For more information about file system access in a clientless environment, see [Section 23.2, "File System Access Overview," on page 294](#).
 - ◆ When specifying a path to a network server, you can use a mapped drive or UNC path (with server name or IP address). If you use a UNC path and your workstations have multiple network clients, applications might launch more slowly than with a mapped drive because the Windows Multiple UNC Provider (MUP) must resolve the UNC paths to the appropriate network client. For more information, see [Microsoft Knowledge Base Article Q150807 \(http://support.microsoft.com/support/kb/articles/Q150/8/07.asp\)](http://support.microsoft.com/support/kb/articles/Q150/8/07.asp).
- 5 Click *Next*, then define the rules used by Application Launcher to determine if a workstation meets the requirements for the application.

The distribution rules ensure that Application Launcher does not distribute the application to workstations that cannot support the application. For example, if the application runs on Windows 2000/XP only, you can create an operating system rule that prohibits distribution to Windows 98 workstations.

NOTE: The requirement for an operating system to be defined before an application is available has been removed.

In previous ZENworks versions, an OS platform had to be defined in the System Requirements before an application was available for distribution and launching. This requirement has been removed.

The new behavior uses the following logic: If an application runs only on a specific operating system, define an operating system distribution rule. If an application does not require a specific operating system, there is no need to define a distribution rule. By default, applications without a defined operating system distribution rule are available on all supported platforms (Windows 98, Windows 2000, and Windows XP).

To add a distribution rule:

5a Click *Add*, then select the type of rule you want to define.

5b Fill in the information for the requirement (click *Help* for information about the requirement or refer to [“Distribution Rules Page” on page 537](#)), then click *OK* to add the requirement to the list.

If you want to create additional distribution rules for the application at a later time, you can use the Distribution Rules page on the Application object. For information, see [“Distribution Rules Page” on page 537](#).

6 Click *Next*, then associate the Application object with the users or workstations that you want to distribute the application to. To do so:

6a Click *Add*, then browse for and select User or Workstation objects.

Each workstation that you want to associate with applications must first be imported into eDirectory as a Workstation object. If a workstation with which you want to associate the application has not been imported as a Workstation object, see [Part III, “Automatic Workstation Import and Removal,” on page 125](#).

You can also select Group objects, Workstation Group objects, and container objects (Organizational Unit, Organization, or Country). If you select a container object, you are given the choice of associating all the container’s User and/or Workstation objects with the application.

Associating an Application object with a Group, Workstation Group, or other container object is the preferred method of associating the Application object in eDirectory. Associating the application to a large number of User or Workstation objects (for example, more than 250) might cause increased server utilization.

IMPORTANT: Do not associate the Application object with Alias objects. Alias objects are not supported.

6b After you add the user or workstation to the list, select the appropriate check boxes for the user or workstation to set the characteristics (*Force run*, *App Launcher*, *Start menu*, *Desktop*, *System tray*, *Quick launch*, and *Force cache*) you want applied to the application. Click *Help* for a description of each of these characteristics, or refer to [“Associations Page” on page 534](#).

If you want to associate the application with additional users or workstations at a later time, you can use the Associations page on the Application object. For information, see [“Associations Page” on page 534](#).

7 Click *Next*, review the Application object settings, then click *Finish* to create the Application object.

8 If, as part of the distribution process for the application, Application Launcher needs to copy files to the workstation, right-click the Application object, click *Properties*, then complete the remaining steps.

or

If the application runs from the network or is already located on the workstation, skip the remain steps and continue with [Section 28.3, “Establishing File System Access,” on page 329](#).

9 Click the *Distribution Options* tab, then click *Application files*.

You need to add the files to the Application Files list in order for Application Launcher to copy them to the workstation.

10 Click *Add*, click *File*, then fill in the following fields.

Source file: Select the file you want copied to the workstation (for example, \\server1\sys\public\notepad\notepad.exe).

Target file: Enter the full path, including the file name, of where you want the file copied (for example, c:\notepad\notepad.exe).

- 11 Click *OK* to add the file to the list.
- 12 Repeat **Step 10** and **Step 11** for each file to be copied.
- 13 When finished adding files, click *OK* to save the Application object information.
- 14 Continue with then next section, **Establishing File System Access**.

28.3 Establishing File System Access

For Application Launcher to be able to distribute or launch an application from a NetWare, Linux, or Windows server, it must have the appropriate rights to the server's file system. If you have not already established file system access for users and/or workstations you've associated the application to, see **Chapter 23, "Novell Application Launcher: Managing Authentication and File System Access,"** on page 293.

28.4 Establishing eDirectory Rights

By default, when a user or workstation is associated with an Application object, the User object or Workstation object is assigned trustee rights to the object. This provides Application Launcher with sufficient eDirectory rights to distribute and launch the application on the user's workstation.

However, for workstation-associated applications that are installed on Windows 98 workstations, you need to also manually assign trustee rights to any users who distribute or launch the application. Windows 98 does not differentiate between the user and workstation, so Application Launcher always uses the user's eDirectory credentials when distributing or launching workstation-associated applications on a Windows 98 workstation. If you don't assign trustee rights to the user, the distribution or launch fails.

To assign trustee rights to a user:

- 1 In ConsoleOne, right-click the Application object, then click *Properties*.
- 2 Click the *NDS Rights* tab, then click *Trustees of this object*.
- 3 Click *Add trustee*, select the user you want to add as a trustee, then click *OK* to add the user to the trustee list. Repeat this step for all users you need to add.
- 4 Click *OK* to save the changes.

28.5 What's Next

To better manage the application you've distributed, you can configure the Application object to:

- ♦ Automatically cache the application's files to the local workstation (if the distribution requires installation of the application). This enables users to install or fix problems with the application when they are disconnected from eDirectory and your network. See **Chapter 24, "Novell Application Launcher: Managing the Cache,"** on page 303.
- ♦ Add the Application object to a folder in the Application Window, Application Explorer window, Application Browser window, or on the Windows Start menu. See **Chapter 25, "Novell Application Launcher: Organizing Applications,"** on page 313.

- ◆ Automatically distribute and run other applications on which it is dependent. See [Chapter 37, “Advanced Distribution: Configuring Application Dependencies and Chains,”](#) on page 399
- ◆ Report on the success or failure of events such as distributing, caching, and uninstalling the application on workstations. See [Chapter 44, “Reporting Application Events,”](#) on page 441.
- ◆ Track the number of licenses being used for the application. See [Chapter 45, “Metering Software Licenses,”](#) on page 469.
- ◆ Determine the schedule of when the application is available to users. See [“Schedule Page”](#) on page 552.
- ◆ Set up backup copies of the application's file package to enable fault tolerance. See [“Setting Up Fault Tolerance for Simple or AOT/AXT Applications”](#) on page 388.
- ◆ Define multiple Application objects for the same application and enable load balancing. See [Section 36.2, “Setting Up Load Balancing,”](#) on page 391.
- ◆ Set up site lists to enable users who travel from site to site to continue to have access to their applications. See [Section 36.3, “Setting Up Site Lists,”](#) on page 395.
- ◆ Define many additional Application object properties to control the behavior of the application. For information about all Application object properties, see [Chapter 48, “Reference: Application Object Settings,”](#) on page 481.

Distribution: Complex Applications

29

Novell® ZENworks® Desktop Management lets you distribute complex applications to users on Windows 98 and Windows 2000/XP workstations.

A complex application is one that requires Application Launcher to copy many files and make many configuration changes to the workstation during distribution of the application. An example is Novell GroupWise® or Microsoft Office.

The following sections help you set up a complex application for distribution:

- ♦ [Section 29.1, “Understanding Software Packages,” on page 331](#)
- ♦ [Section 29.2, “Preparing the Software Package,” on page 333](#)
- ♦ [Section 29.3, “Configuring the Application in eDirectory,” on page 333](#)
- ♦ [Section 29.4, “Establishing File System Access,” on page 339](#)
- ♦ [Section 29.5, “Establishing eDirectory Rights,” on page 339](#)
- ♦ [Section 29.6, “What's Next,” on page 339](#)

29.1 Understanding Software Packages

A complex application such as Novell GroupWise or Microsoft Office requires many files to be installed and many configuration changes to be made to the workstation. To ensure that complex application are distributed and installed correctly, ZENworks Desktop Management supports the use of software packages.

Software packages contain the files, configuration settings, and instructions required to install the application. The following sections describe the two types of software packages that ZENworks Desktop Management supports:

- ♦ [Section 29.1.1, “Microsoft Windows Installer Packages,” on page 331](#)
- ♦ [Section 29.1.2, “ZENworks snAppShot Packages,” on page 332](#)

In addition to the information provided in the sections listed above, the following articles contain excellent information about software packages and the ZENworks utilities provided to create them:

- ♦ [Novell snAppShot versus FLEXnet AdminStudio's Repackager \(http://www.macrovision.com/company/news/newsletter/tips/novell_vs_adminstudio.shtml\)](http://www.macrovision.com/company/news/newsletter/tips/novell_vs_adminstudio.shtml)
- ♦ [Consistently Consistent \(http://www.novell.com/connectionmagazine/2004/11/tech_talk_2.html\)](http://www.novell.com/connectionmagazine/2004/11/tech_talk_2.html)

29.1.1 Microsoft Windows Installer Packages

The Microsoft Windows Installer is an installation and configuration service that ships as part of the Windows 2000 and Windows XP operating systems, and is provided in a service pack to Windows 98.

The software packages used by Windows Installer are called Windows Installer packages or MSI packages (because of their `.msi` file extension). Windows Installer packages are the common and accepted standard for Windows applications. Because of the benefits associated with using the native Windows Installer, Novell recommends that you use Windows Installer packages whenever possible.

Vendor-Supplied MSI Packages

Many software vendors provide Windows Installer packages that include the files and configuration settings required by the Windows Installer to install the application to a workstation. In this case, unless you want to customize the package, the package is ready to distribute.

Customized MSI Packages

Different groups within an organization often use the same application, but that doesn't mean they require the same feature set. One of the benefits of Windows Installer is that if you have 10 groups needing 10 different feature sets or other alterations for the same application, you can deploy the same MSI package to all 10 user groups, but with a different transform file (MST) applied for each group.

A transform file is a collection of changes applied to an MSI installation. It contains all modification information, such as whether features are installed; how they are installed; which files, shortcuts, and registry entries are included; and Add/Remove Programs applet information.

If you have vendor-supplied MSI packages, you can use FLEXnet AdminStudio 7.5 ZENworks Edition to create and manage transforms. For information about using AdminStudio ZENworks Edition, see [Chapter 46, "Reference: AdminStudio ZENworks Edition," on page 473](#).

Non-Windows Installer Installations

Not all Windows applications are designed to be installed by Windows Installer. Many applications use other software installers such as MacroVision InstallShield.

AdminStudio ZENworks Edition includes a Repackaging Wizard that lets you create MSI packages for non-Windows Installer applications. For information about using AdminStudio ZENworks Edition, see [Chapter 46, "Reference: AdminStudio ZENworks Edition," on page 473](#).

29.1.2 ZENworks snAppShot Packages

Windows Installer packages are the recommended package format for distributing applications. If you cannot create a Windows Installer package for an application, you can use the ZENworks snAppShot utility to create a software package.

ZENworks snAppShot was created prior to the existence of Windows Installer. It lets you create a proprietary software package called a snAppShot package or AOT package/AXT package. AOT stands for Application Object Template and AXT stands for Application Object Text Template. Both packages contain the same information. However, an AOT package is an uneditable binary format, while an AXT package is an editable text-based format.

For information about using snAppShot, see [Chapter 47, "Reference: ZENworks SnAppShot," on page 475](#).

29.2 Preparing the Software Package

- 1 To create or customize a Windows Installer package, see [Chapter 46, “Reference: AdminStudio ZENworks Edition,”](#) on page 473.

or

To create a ZENworks snAppShot package, see [Chapter 47, “Reference: ZENworks SnAppShot,”](#) on page 475.

- 2 Make sure the NetWare, Linux, or Windows server where you save the Windows Installer package or the ZENworks snAppShot package is available to Application Launcher.

For more information about what is required for Application Launcher to access a network resource, see [Chapter 23, “Novell Application Launcher: Managing Authentication and File System Access,”](#) on page 293.

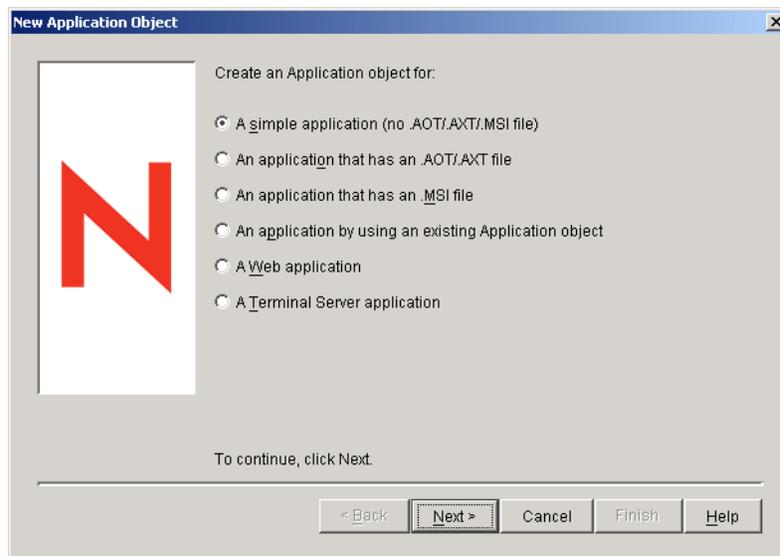
29.3 Configuring the Application in eDirectory

After you have prepared the application for distribution (see [Section 29.1, “Understanding Software Packages,”](#) on page 331), you are ready to create the application as an Application object in eDirectory, define its distribution rules, and associate it with users and workstations.

You can create the Application object in any container you want. Because Application Launcher accesses the object on behalf of the logged-in user or workstation, you should place it in a container whose partition (or a replica of the partition) is available to the user or workstation over a local area connection rather than a wide area connection. For more information, see [Chapter 51, “Reference: Application Object Location,”](#) on page 617.

To create the Application object in eDirectory and configure it:

- 1 In ConsoleOne[®], right-click the container where you want to create the Application object, click *New*, then click *Application* to display the New Application Object dialog box.



- 2 Select from the following options to create the appropriate type of object for the application:

An application that has an .aot/.axt file: Lets you specify a .aot or .axt file you've created with snAppShot or exported from another Application object. The .aot or .axt file is used to populate the Application object's property fields. Skip to [Section 29.3.1, “Creating an Application Object from a ZENworks snAppshot File,”](#) on page 334.

An application that has a .msi file: Lets you specify a Microsoft Windows Installer (.msi) file. The .msi file is used to populate the Application object's property fields. Skip to [Section 29.3.2, “Creating the Application Object from a Windows Installer \(.MSI\) File,”](#) on page 336.

29.3.1 Creating an Application Object from a ZENworks snAppshot File

- 1 (Conditional) If the New Application Object dialog box is not open, see [Section 29.3, “Configuring the Application in eDirectory,”](#) on page 333.
- 2 In the New Application Object dialog box, select the *Application that has an .aot/.axt file* option, then click *Next*.
- 3 Specify the path to the .aot or .axt file.

or

Click the browse button to browse for and select the file.

The file should be in the network location where you saved it when creating it with snAppShot. If you browse for the file, the Open dialog box defaults to * .axt for its file type display. If you created an .aot file, you must change the file type display to * .aot or All Files in order to select the .aot file.

- 4 Click *Next*, then modify the following fields to customize the Application object.

Object name: This field defaults to the Application object name that was specified when running snAppShot. You can change the name if you want. The name must conform to the following rules:

- ♦ The name must be unique in the container.
- ♦ Special characters are allowed. However, plus (+), equals (=), and period (.) must be preceded by a backslash (\) if used.
- ♦ The following characters are valid in Application object names but are invalid when used in Windows folder and file names:

\ / : * ? " < > |

If you use these characters in the Application object name, they are replaced by an underscore (_) when displayed in locations controlled by Windows and not Novell Application Launcher™ (for example, on the Windows desktop).

- ♦ Uppercase and lowercase letters, as well as underscores and spaces, are displayed as you first entered them, but they aren't distinguished. For example, ZENworks_Desktop_Management and ZENWORKS DESKTOP MANAGEMENT are considered identical.

The Application object's name is visible in eDirectory. By default, the name is also used as the Application object's icon title when displayed by Application Launcher on a user's workstation. You can, if necessary, change the icon title after the Application object has been created (Application object > *Identification* tab > *Icon* page).

SOURCE_PATH (location of installation files (.fil)): This field defaults to the location where the application's files (.fil) files were stored when running snAppShot. You should verify that the path is correct. If the path uses a drive mapping, you can either 1) ensure that all workstations have the same drive mapped to the source location or 2) change the drive mapping to another format such as UNC. For information about valid formats, see Filepath Syntax in [Section 23.2, “File System Access Overview,” on page 294](#).

The path you enter here is added as the SOURCE_PATH macro in the Macros list for the Application object (*Common* tab > *Macros* page) and used in any fields that require a path to the source location.

TARGET_PATH (client workstation directory path): This path specifies the workstation location where the application files should be installed. It defaults to the path defined in the .aot or .axt, which is the location where the application was installed when running snAppShot. You should verify that this is the workstation directory where you want the application installed.

The path you enter here is added as the TARGET_PATH macro in the Macros list for the Application object (*Common* tab > *Macros* page) and used in any fields that require a path to the target location.

- 5 Click *Next*, then define the rules used by Application Launcher to determine if a workstation meets the requirements for the application.

The distribution rules ensure that Application Launcher does not distribute the application to workstations that cannot support the application. For example, if the application runs on Windows 2000/XP only, you can create an operating system rule that prohibits distribution to Windows 98 workstations.

NOTE: The requirement for an operating system to be defined before an application is available has been removed.

In previous ZENworks versions, an OS platform had to be defined in the System Requirements before an application would be available for distribution and launching. This requirement has been removed.

The new behavior uses the following logic: If an application runs only on a specific operating system, define an operating system distribution rule. If an application does not require a specific operating system, there is no need to define a distribution rule. By default, applications without a defined operating system distribution rule are available on all supported platforms (Windows 98, Windows 2000, and Windows XP).

To add a distribution rule:

5a Click *Add*, then select the type of rule you want to define.

5b Fill in the information for the requirement (click *Help* for information about the requirement or refer to [“Distribution Rules Page” on page 537](#)), then click *OK* to add the requirement to the list.

If you want to create additional distribution rules for the application at a later time, you can use the Distribution Rules page on the Application object. For information, see [“Distribution Rules Page” on page 537](#).

- 6 Click *Next*, then associate the Application object with the users or workstations that you want to distribute the application to. To do so:

6a Click *Add*, then browse for and select User or Workstation objects. You can also select Group objects, Workstation Group objects, and container objects (Organizational Unit,

Organization, or Country). If you select a container object, you are given the choice of associating all the container's User and/or Workstation objects with the application.

Each workstation that you want to associate with applications must first be imported into eDirectory as a Workstation object. If a workstation with which you want to associate the application has not been imported as a Workstation object, see [Part III, “Automatic Workstation Import and Removal,” on page 125](#).

Associating an Application object with a Group, Workstation Group, or other container object is the preferred method of associating the Application object in eDirectory. Associating the application to a large number of User or Workstation objects (for example, more than 250) might cause increased server utilization.

IMPORTANT: Do not associate the Application object with Alias objects. Alias objects are not supported.

- 6b** After you add the user or workstation to the list, select the appropriate check boxes for the user or workstation to set the characteristics (*Force run, App Launcher, Start menu, Desktop, System tray, Quick launch, and Force cache*) you want applied to the application. Click *Help* for a description of each of these characteristics, or refer to [“Associations Page” on page 534](#).

If you want to associate the application with additional users or workstations at a later time, you can use the Associations page on the Application object. For information, see [“Associations Page” on page 534](#).

- 7** Click *Next*, review the Application object settings, then click *Finish* to create the Application object.
- 8** Continue with [Section 28.3, “Establishing File System Access,” on page 329](#).

29.3.2 Creating the Application Object from a Windows Installer (.MSI) File

- 1** (Conditional) If the New Application Object dialog box is not open, see [Section 29.3, “Configuring the Application in eDirectory,” on page 333](#).
- 2** In the New Application Object dialog box, select the *Application that has an .msi file* option, then click *Next*.
- 3** In the *Path to .msi file* field, specify the complete path to the `.msi` file to use as the source file during distribution to workstation.

You can use a mapped drive or UNC path. If you use a drive mapping, you must ensure that all workstations have the same drive mapped to the source location. The path you enter here is added to the *Package source* list for the Application object (*Common* tab > *Sources*).

NOTE: After you create the Application object, you cannot change the `.msi` filename; however, you can change the path to the `.msi` file. If you change the `.msi` filename, the installation fails.

- 4** Click *Next*, then modify the following fields to customize the Application object.
- Object name:** This field defaults to the Application object name defined in the `.msi` file. You can change the name if you want. The name must conform to the following rules:
- ♦ The name must be unique in the container.

- ◆ Special characters are allowed. However, plus (+), equals (=), and period (.) must be preceded by a backslash (\) if used.
- ◆ The following characters are valid in Application object names but are invalid when used in Windows folder and file names:

\ / : * ? " < > |

If you use these characters in the Application object name, they are replaced by an underscore (_) when displayed in locations controlled by Windows and not Novell Application Launcher (for example, on the Windows desktop).

- ◆ Uppercase and lowercase letters, as well as underscores and spaces, are displayed as you first entered them, but they aren't distinguished. For example, ZENworks_Desktop_Management and ZENWORKS DESKTOP MANAGEMENT are considered identical.

The Application object's name is visible in eDirectory. By default, the name is also used as the Application object's icon title when displayed by Application Launcher on a user's workstation. You can, if necessary, change the icon title after the Application object has been created (Application object > *Identification* tab > *Icon* page).

Administration package path: This path specifies the location of the MSI package you want to use for administrative purposes. ConsoleOne uses the .msi file at this location to populate information in the Application object. This field is used only by ConsoleOne for reading of the .msi package. It is not used by Novell Application Launcher for distribution of the application. For distribution, Application Launcher uses the path defined in the *Path to .msi file* field located on the previous page.

The path defaults to the path defined in *Path to .msi file* field on the previous page. Change it if necessary. You can use a mapped drive or UNC path. If you use a drive mapping, you must ensure that all ConsoleOne workstations have the same drive mapped to the location. You cannot use macros in this field.

The path you enter here is added to the *Administration package path* field for the Application object (*Identification* tab > *Package Information* page).

NOTE: Do not use macros in this field or creation of the Application object will fail. After ConsoleOne has created the Application object, you can define a macro for the source location (*Common* tab > *Macros*) and use it in other Application object fields (such as the *Package source* list) if desired.

- 5 Click *Next*, then define the rules used by Application Launcher to determine if a workstation meets the requirements for the application.

The distribution rules ensure that Application Launcher does not distribute the application to workstations that cannot support the application. For example, if the application runs on Windows 2000/XP only, you can create an operating system rule that prohibits distribution to Windows 98 workstations.

To add a distribution rule:

- 5a Click *Add*, then select the type of rule you want to define.
- 5b Fill in the information for the requirement (click *Help* for information about the requirement or refer to [“Distribution Rules Page” on page 537](#)), then click *OK* to add the requirement to the list.

If you want to create additional distribution rules for the application at a later time, you can use the Distribution Rules page on the Application object. For information, see [“Distribution Rules Page” on page 537](#).

- 6 Click *Next*, then associate the Application object with the users or workstations that you want to distribute the application to. To do so:

- 6a Click *Add*, then browse for and select User or Workstation objects.

Each workstation that you want to associate with applications must first be imported into eDirectory as a Workstation object. If a workstation with which you want to associate the application has not been imported as a Workstation object, see [Part III, “Automatic Workstation Import and Removal,” on page 125](#).

You can also select Group objects, Workstation Group objects, and container objects (Organizational Unit, Organization, or Country). If you select a container object, you are given the choice of associating all the container's User and/or Workstation objects with the application.

Associating an Application object with a Group, Workstation Group, or other container object is the preferred method of associating the Application object in eDirectory. Associating the application to a large number of User or Workstation objects (for example, more than 250) might cause increased server utilization.

IMPORTANT: Do not associate the Application object with Alias objects. Alias objects are not supported.

- 6b After you add the user or workstation to the list, select the appropriate check boxes for the user or workstation to set the characteristics (*Force run*, *App Launcher*, *Start menu*, *Desktop*, *System tray*, *Quick launch*, and *Force cache*) you want applied to the application. Click *Help* for a description of each of these characteristics, or refer to [“Associations Page” on page 534](#).

With MSI applications, you must use the *Force cache* option if users or workstations do not have network client access to the source `.msi` files. The Microsoft Windows Installer requires file access that is provided by a network client but not by the ZENworks Desktop Management Agent. Consider the following examples:

- ♦ Users outside your firewall need an MSI application but have no network client access to the source `.msi` files on a server inside your firewall. They log in to the ZENworks Middle Tier Server and Application Launcher displays the MSI application. For successful distribution to occur, the MSI application must be marked as *Force cache* so that the source `.msi` files are copied to the user's cache directory (through the Middle Tier Server) and then distributed from the cache directory.
- ♦ Users inside your firewall need an MSI application. They don't have the Novell Client™ installed, so they log in to the ZENworks Middle Tier Server to authenticate to eDirectory. The users are part of an Active Directory domain, and the source `.msi` files are located on a Windows share that they have rights to. The distribution succeeds without force caching the application because the Microsoft network client provides file access to the source `.msi` files.

If you want to associate the application with additional users or workstations at a later time, you can use the Associations page on the Application object. For information, see [“Associations Page” on page 534](#).

- 7 Click *Next*, review the Application object settings, then click *Finish* to create the Application object.

If, after you create an Application object for an MSI application, you receive a new MSI package (`.msi` file) for the application, you must create a new Application object using the new `.msi` file. You cannot simply replace the old `.msi` file with the new one.

For example, the Desktop Management Agent is packaged as an `.msi` file (`zfdagent.msi`) that can be distributed through an Application object. Each time you receive a new `zfdagent.msi` file (through an upgrade or support pack), you must create a new Application object for it. This ensures that the GUID (global unique identifier) contained in the `.msi` file is synchronized with the one in the Application object and on the workstation, thus enabling the application to be installed and uninstalled correctly.

8 Continue with [Section 28.3, “Establishing File System Access,”](#) on page 329.

29.4 Establishing File System Access

For Application Launcher to be able to distribute or launch an application from a NetWare, Linux, or Windows server, it must have the appropriate rights to the server's file system. If you have not already established file system access for users and/or workstations you've associated the application to, see [Chapter 23, “Novell Application Launcher: Managing Authentication and File System Access,”](#) on page 293.

29.5 Establishing eDirectory Rights

By default, when a user or workstation is associated with an Application object, the User object or Workstation object is assigned trustee rights to the object. This provides Application Launcher with sufficient eDirectory rights to distribute and launch the application on the user's workstation.

However, for workstation-associated applications that are installed on Windows 98 workstations, you need to also manually assign trustee rights to any users who distribute or launch the application. Windows 98 does not differentiate between the user and workstation, so Application Launcher always uses the user's eDirectory credentials when distributing or launching workstation-associated applications on a Windows 98 workstation. If you don't assign trustee rights to the user, the distribution or launch fails.

To assign trustee rights to a user:

- 1 In ConsoleOne, right-click the Application object, then click *Properties*.
- 2 Click the *NDS Rights* tab, then click *Trustees of this object*.
- 3 Click *Add trustee*, select the user you want to add as a trustee, then click *OK* to add the user to the trustee list. Repeat this step for all users you need to add.
- 4 Click *OK* to save the changes.

29.6 What's Next

To better manage the application you've distributed, you can configure the Application object to:

- ♦ Automatically cache the application's files to the local workstation (if the distribution requires installation of the application). This enables users to install or fix problems with the application when they are disconnected from eDirectory and your network. See [“Novell Application Launcher: Managing the Cache”](#) on page 303.
- ♦ Add the Application object to a folder in the Application Window, Application Explorer window, Application Browser window, or on the Windows Start menu. See [Chapter 25, “Novell Application Launcher: Organizing Applications,”](#) on page 313.
- ♦ Ensure that all dependent applications are automatically distributed before the application. For example, if AppA has a dependency on AppB and AppC, you can ensure that AppB and AppC

are installed before AppA. See [Chapter 37, “Advanced Distribution: Configuring Application Dependencies and Chains,”](#) on page 399

- ◆ Report on the success or failure of events such as distributing, caching, and uninstalling the application on workstations. See [Chapter 44, “Reporting Application Events,”](#) on page 441.
- ◆ Track the number of licenses being used for the application. See [Chapter 45, “Metering Software Licenses,”](#) on page 469.
- ◆ Determine the schedule of when the application is available. See [“Schedule Page”](#) on page 552.
- ◆ Set up backup copies of the application's file package to enable fault tolerance. See [Section 36.1, “Setting Up Fault Tolerance,”](#) on page 387.
- ◆ Define multiple Application objects for the same application and enable load balancing. See [Section 36.2, “Setting Up Load Balancing,”](#) on page 391.
- ◆ Set up site lists to enable users who travel from site to site to install applications from the closest server. See [Section 36.3, “Setting Up Site Lists,”](#) on page 395.
- ◆ Define many additional Application object properties to control the behavior of the application. For information about all Application object properties, see [Chapter 48, “Reference: Application Object Settings,”](#) on page 481.

Distribution: Terminal Server Applications

30

Novell® ZENworks® Desktop Management lets you distribute terminal server applications to users on Windows 98 and Windows 2000/XP workstations.

Distribution of a terminal server application requires Application Launcher to display an application icon that, when launched, opens a client session to the terminal server and starts the application. An example is Novell GroupWise® or Microsoft Office running on a terminal server rather than the user's local workstation.

Complete the tasks in the following order to distribute a terminal server application:

- ◆ [Section 30.1, “Preparing a Terminal Server Application,” on page 341](#)
- ◆ [Section 30.2, “Configuring the Application in eDirectory,” on page 341](#)
- ◆ [Section 30.3, “Establishing eDirectory Rights,” on page 344](#)
- ◆ [Section 30.4, “What's Next,” on page 345](#)

For Application Launcher to successfully launch a terminal server application from a user's workstation, you need to ensure that the user's workstation is properly configured. For instructions, see [Chapter 38, “Users: Supporting Terminal Server Users,” on page 407](#).

30.1 Preparing a Terminal Server Application

A terminal server application is an application being hosted on a Microsoft Windows terminal server or a Citrix MetaFrame server. Users run the application through terminal server client sessions on their workstations.

To prepare a terminal server application for distribution to users:

- 1 Install the application on the terminal server. For information about distributing applications to terminal servers, see [“Distribution: Applications to Terminal Servers” on page 351](#).
- 2 Ensure that users have the ICA client (for Citrix MetaFrame servers) or RDP client (for Windows terminal servers) installed on their workstations. For more information, see [Section 38.1, “Installing the RDP and ICA Clients,” on page 407](#).
- 3 Ensure that users have terminal server accounts that provide the file system rights needed to run the application. If desired, you can set up one account for all users. For more information, see [Section 38.3, “Managing Terminal Server User Accounts,” on page 408](#).

30.2 Configuring the Application in eDirectory

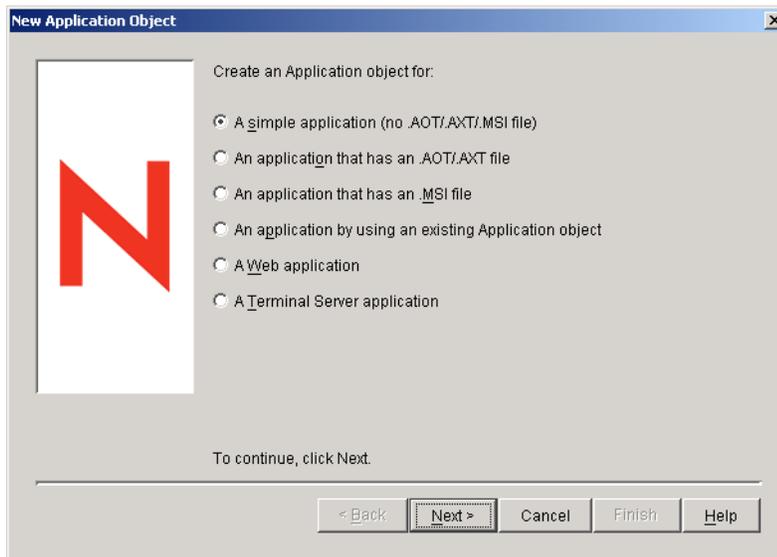
After you have prepared the application for distribution (see [Section 30.1, “Preparing a Terminal Server Application,” on page 341](#)), you are ready to create the application as an Application object in eDirectory, define its distribution rules, and associate it with users and workstations.

You can create the Application object in any container you want. Because Application Launcher accesses the object on behalf of the logged-in user or workstation, you should place it in a container

whose partition (or a replica of the partition) is available to the user or workstation over a local area connection rather than a wide area connection. For more information, see [Chapter 51, “Reference: Application Object Location,”](#) on page 617.

To create the Application object in eDirectory™ and configure it:

- 1 In ConsoleOne®, right-click the container where you want to create the Application object, click *New*, then click *Application* to display the New Application Object dialog box.



- 2 In the New Application Object dialog box, select the *Terminal server application* option, then click *Next*.
- 3 Specify a name for the Application object.

The Application object's name must conform to the following rules:

- ♦ The name must be unique in the container.
- ♦ Special characters are allowed. However, plus (+), equals (=), and period (.) must be preceded by a backslash (\) if used.
- ♦ The following characters are valid in Application object names but are invalid when used in Windows folder and file names:
\\ / : * ? " < > |

If you use these characters in the Application object name, they are replaced by an underscore (_) when displayed in locations controlled by Windows and not Novell Application Launcher™ (for example, on the Windows desktop).

- ♦ Uppercase and lowercase letters, as well as underscores and spaces, are displayed as you first entered them, but they aren't distinguished. For example, ZENworks_Desktop_Management and ZENWORKS DESKTOP MANAGEMENT are considered identical.

The Application object's name is visible in eDirectory. By default, the name is also used as the Application object's icon title when displayed by Application Launcher on a user's workstation. You can, if necessary, change the icon title after the Application object has been created (Application object > *Identification* tab > *Icon* page).

- 4 Click *Next*, then fill in the following information to configure the terminal server client session in which the application runs.

ICA session: Select this option if the terminal server requires the application to run in an ICA (Independent Computing Architecture) client session. Citrix MetaFrame requires ICA client sessions. Fill in the following fields:

- ♦ **Published application name:** By default, this field is populated with the Application object name you previously entered. If the name does not match the application name exactly as it is defined in Citrix, change the name to the Citrix published application name.
- ♦ **Servers hosting this application:** Add the Citrix servers that host the application. To add a server, click the *Add* button, specify the server's IP address or hostname, then click *OK*.
The server's you define here are used only when the application is launched from Application Launcher or the NAL plug-in. The ZENworks Launch gadget uses the servers defined in its configuration settings. For information about the ZENworks Launch gadget's configuration settings, see [Chapter 26, "ZENworks Launch Gadget: Configuring Settings,"](#) on page 319.

RDP session: Select this option if the terminal server requires the application to run in an RDP (Remote Desktop Protocol) client session. Microsoft Windows Terminal Server requires RDP client sessions.

- ♦ **Terminal server address and port:** Specify the terminal server's IP address or hostname. If the terminal server is not using default port 3389, edit the *Port* field to specify the correct port number.
- ♦ **Server domain:** If the terminal server is part of a Windows NT domain or an Active Directory domain, specify the domain name. If the user's eDirectory username and password match the domain username and password, the user is not prompted to log in to the server when the RDP session is initiated.
- ♦ **Application path:** Specify the path to the application's executable file. If you are running ConsoleOne on the terminal server, you can browse for and select the file.
- ♦ **Working directory:** Specify the path to the directory you want the application to use for its working files.

- 5 Click *Next*, then define the rules used by Application Launcher to determine if a workstation meets the requirements for the application.

The distribution rules ensure that Application Launcher does not distribute the application to workstations that cannot support the application. For example, if the application runs on Windows 2000/XP only, you can create an operating system rule that prohibits distribution to Windows 98 workstations.

NOTE: The requirement for an operating system to be defined before an application is available has been removed.

In previous ZENworks versions, an OS platform had to be defined in the System Requirements before an application would be available for distribution and launching. This requirement has been removed.

The new behavior uses the following logic: If an application runs only on a specific operating system, define an operating system distribution rule. If an application does not require a specific operating system, there is no need to define a distribution rule. By default, applications without a defined operating system distribution rule are available on all supported platforms (Windows 98, Windows 2000, and Windows XP).

To add a distribution rule:

5a Click *Add*, then select the type of rule you want to define.

5b Fill in the information for the requirement (click *Help* for information about the requirement or refer to “[Distribution Rules Page](#)” on page 537), then click *OK* to add the requirement to the list.

If you want to create additional distribution rules for the application at a later time, you can use the Distribution Rules page on the Application object. For information, see “[Distribution Rules Page](#)” on page 537.

6 Click *Next*, then associate the Application object with the users or workstations that you want to distribute the application to. To do so:

6a Click *Add*, then browse for and select User or Workstation objects.

Each workstation that you want to associate with applications must first be imported into eDirectory as a Workstation object. If a workstation with which you want to associate the application has not been imported as a Workstation object, see [Part III, “Automatic Workstation Import and Removal,”](#) on page 125.

You can also select Group objects, Workstation Group objects, and container objects (Organizational Unit, Organization, or Country). If you select a container object, you are given the choice of associating all the container's User and/or Workstation objects with the application.

Associating an Application object with a Group, Workstation Group, or other container object is the preferred method of associating the Application object in eDirectory.

Associating the application to a large number of User or Workstation objects (for example, more than 250) might cause increased server utilization.

IMPORTANT: Do not associate the Application object with Alias objects. Alias objects are not supported.

6b After you add the user or workstation to the list, select the appropriate check boxes for the user or workstation to set the characteristics (*Force run*, *App Launcher*, *Start menu*, *Desktop*, *System tray*, *Quick launch*, and *Force cache*) you want applied to the application. Click *Help* for a description of each of these characteristics, or refer to “[Associations Page](#)” on page 534.

If you want to associate the application with additional users or workstations at a later time, you can use the Associations page on the Application object. For information, see “[Associations Page](#)” on page 534.

7 Click *Next*, review the Application object settings, then click *Finish* to create the Application object.

30.3 Establishing eDirectory Rights

By default, when a user or workstation is associated with an Application object, the User object or Workstation object is assigned trustee rights to the object. This provides Application Launcher with sufficient eDirectory rights to distribute and launch the application on the user's workstation.

However, for workstation-associated applications that are installed on Windows 98 workstations, you need to also manually assign trustee rights to any users who distribute or launch the application. Windows 98 does not differentiate between the user and workstation, so Application Launcher always uses the user's eDirectory credentials when distributing or launching workstation-associated

applications on a Windows 98 workstation. If you don't assign trustee rights to the user, the distribution or launch fails.

To assign trustee rights to a user:

- 1 In ConsoleOne, right-click the Application object, then click *Properties*.
- 2 Click the *NDS Rights* tab, then click *Trustees of this object*.
- 3 Click *Add trustee*, select the user you want to add as a trustee, then click *OK* to add the user to the trustee list. Repeat this step for all users you need to add.
- 4 Click *OK* to save the changes.

30.4 What's Next

To better manage the application you've distributed, you can configure the Application object to:

- ♦ Automatically cache the application's files to the local workstation (if the distribution requires installation of the application). This enables users to install or fix problems with the application when they are disconnected from eDirectory and your network. See [“Novell Application Launcher: Managing the Cache” on page 303](#).
- ♦ Add the Application object to a folder in the Application Window, Application Explorer window, Application Browser window, or on the Windows Start menu. See [Chapter 25, “Novell Application Launcher: Organizing Applications,” on page 313](#).
- ♦ Automatically distribute and run other applications on which it is dependent. See [Chapter 37, “Advanced Distribution: Configuring Application Dependencies and Chains,” on page 399](#).
- ♦ Report on the success or failure of events such as distributing, caching, and uninstalling the application on workstations. See [Chapter 44, “Reporting Application Events,” on page 441](#).
- ♦ Determine the schedule of when the application is available to users. See [“Schedule Page” on page 552](#).
- ♦ Define many additional Application object properties to control the behavior of the application. For information about all Application object properties, see [Chapter 48, “Reference: Application Object Settings,” on page 481](#).

Novell® ZENworks® Desktop Management lets you distribute Web applications to users on Windows 98 and Windows 2000/XP workstations.

Distribution of a Web applications requires Application Launcher to display an application icon that, when launched, opens the workstation's Web browser and displays the Web-based application (or other Web content). No files need to be copied to the workstation and no configuration settings are modified.

Complete the tasks in the following order to distribute a Web application:

1. [Section 31.1, “Preparing a Web Application,” on page 347](#)
2. [Section 31.2, “Configuring the Application in eDirectory,” on page 347](#)
3. [Section 31.3, “Establishing eDirectory Rights,” on page 350](#)
4. [Section 31.4, “What's Next,” on page 350](#)

31.1 Preparing a Web Application

A Web application is any application (or Web content) that can be launched via a URL in a Web browser. You are simply making the Web application's URL available to users through a shortcut displayed by Application Launcher. When a user selects the shortcut, Application Launcher launches the user's Web browser, which then displays the Web application.

To prepare a Web application for distribution to users:

- 1 Install the Web application to the appropriate Web servers.
- 2 If you use a security product, such as Novell iChain®, to provide secure authentication and access to your Web content, make sure you've established access for the users to whom you distribute the Web application.

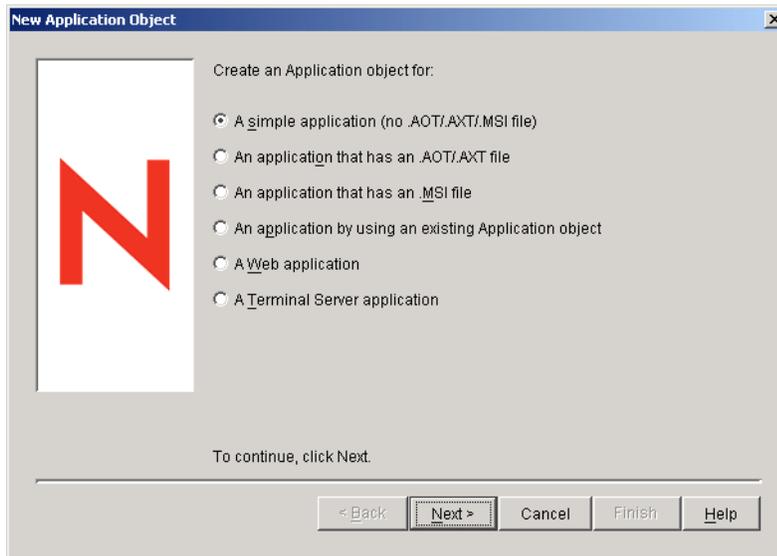
31.2 Configuring the Application in eDirectory

After you have prepared the application for distribution (see [Section 31.1, “Preparing a Web Application,” on page 347](#)), you are ready to create the application as an Application object in eDirectory, define its distribution rules, and associate it with users and workstations.

You can create the Application object in any container you want. Because Application Launcher accesses the object on behalf of the logged-in user or workstation, you should place it in a container whose partition (or a replica of the partition) is available to the user or workstation over a local area connection rather than a wide area connection. For more information, see [Chapter 51, “Reference: Application Object Location,” on page 617](#).

To create the Application object in eDirectory and configure it:

- 1 In ConsoleOne[®], right-click the container where you want to create the Application object, click *New*, then click *Application* to display the New Application Object dialog box.



- 2 In the New Application Object dialog box, select the *Web application* option, then click *Next*.
- 3 Enter a name for the Application object.

The Application object's name must conform to the following rules:

- ♦ The name must be unique in the container.
- ♦ Special characters are allowed. However, plus (+), equals (=), and period (.) must be preceded by a backslash (\) if used.
- ♦ The following characters are valid in Application object names but are invalid when used in Windows folder and file names:

\ / : * ? " < > |

If you use these characters in the Application object name, they are replaced by an underscore (_) when displayed in locations controlled by Windows and not Novell Application Launcher[™] (for example, on the Windows desktop).

- ♦ Uppercase and lowercase letters, as well as underscores and spaces, are displayed as you first entered them, but they aren't distinguished. For example, ZENworks_Desktop_Management and ZENWORKS DESKTOP MANAGEMENT are considered identical.

The Application object's name is visible in eDirectory[™]. By default, the name is also used as the Application object's icon title when displayed by Application Launcher on a user's workstation. You can, if necessary, change the icon title after the Application object has been created (Application object > *Identification* tab > *Icon* page).

- 4 Click *Next*, then enter the Web application's URL.

The URL can be to the Web application or to a Web page that provides access to the application.

- 5 Click *Next*, then define the rules used by Application Launcher to determine if a workstation meets the requirements for the application.

The distribution rules ensure that Application Launcher does not distribute the application to workstations that cannot support the application. For example, if the application runs on Windows 2000/XP only, you can create an operating system rule that prohibits distribution to Windows 98 workstations.

NOTE: The requirement for an operating system to be defined before an application is available has been removed.

In previous ZENworks versions, an OS platform had to be defined in the System Requirements before an application would be available for distribution and launching. This requirement has been removed.

The new behavior uses the following logic: If an application runs only on a specific operating system, define an operating system distribution rule. If an application does not require a specific operating system, there is no need to define a distribution rule. By default, applications without a defined operating system distribution rule are available on all supported platforms (Windows 98, Windows 2000, and Windows XP).

To add a distribution rule:

5a Click *Add*, then select the type of rule you want to define.

5b Fill in the information for the requirement (click *Help* for information about the requirement or refer to “[Distribution Rules Page](#)” on page 537), then click *OK* to add the requirement to the list.

If you want to create additional distribution rules for the application at a later time, you can use the Distribution Rules page on the Application object. For information, see “[Distribution Rules Page](#)” on page 537.

6 Click *Next*, then associate the Application object with the users or workstations that you want to distribute the application to. To do so:

6a Click *Add*, then browse for and select User or Workstation objects.

Each workstation that you want to associate with applications must first be imported into eDirectory as a Workstation object. If a workstation with which you want to associate the application has not been imported as a Workstation object, see [Part III, “Automatic Workstation Import and Removal,”](#) on page 125.

You can also select Group objects, Workstation Group objects, and container objects (Organizational Unit, Organization, or Country). If you select a container object, you are given the choice of associating all the container's User and/or Workstation objects with the application.

Associating an Application object with a Group, Workstation Group, or other container object is the preferred method of associating the Application object in eDirectory.

Associating the application to a large number of User or Workstation objects (for example, more than 250) might cause increased server utilization.

IMPORTANT: Do not associate the Application object with Alias objects. Alias objects are not supported.

6b After you add the user or workstation to the list, select the appropriate check boxes for the user or workstation to set the characteristics (*Force run*, *App Launcher*, *Start menu*, *Desktop*, *System tray*, *Quick launch*, and *Force cache*) you want applied to the application. Click *Help* for a description of each of these characteristics, or refer to “[Associations Page](#)” on page 534.

If you want to associate the application with additional users or workstations at a later time, you can use the Associations page on the Application object. For information, see [“Associations Page” on page 534](#).

- 7 Click *Next*, review the Application object settings, then click *Finish* to create the Application object.

31.3 Establishing eDirectory Rights

By default, when a user or workstation is associated with an Application object, the User object or Workstation object is assigned trustee rights to the object. This provides Application Launcher with sufficient eDirectory rights to distribute and launch the application on the user's workstation.

However, for workstation-associated applications that are installed on Windows 98 workstations, you need to also manually assign trustee rights to any users who distribute or launch the application. Windows 98 does not differentiate between the user and workstation, so Application Launcher always uses the user's eDirectory credentials when distributing or launching workstation-associated applications on a Windows 98 workstation. If you don't assign trustee rights to the user, the distribution or launch fails.

To assign trustee rights to a user:

- 1 In ConsoleOne, right-click the Application object, then click *Properties*.
- 2 Click the *NDS Rights* tab, then click *Trustees of this object*.
- 3 Click *Add trustee*, select the user you want to add as a trustee, then click *OK* to add the user to the trustee list. Repeat this step for all users you need to add.
- 4 Click *OK* to save the changes.

31.4 What's Next

To better manage the application you've distributed, you can configure the Application object to:

- ♦ Automatically cache the application's files to the local workstation (if the distribution requires installation of the application). This enables users to install or fix problems with the application when they are disconnected from eDirectory and your network. See [Chapter 24, “Novell Application Launcher: Managing the Cache,” on page 303](#).
- ♦ Add the Application object to a folder in the Application Window, Application Explorer window, Application Browser window, or on the Windows Start menu. See [Chapter 25, “Novell Application Launcher: Organizing Applications,” on page 313](#).
- ♦ Automatically distribute and run other applications on which it is dependent. See [Chapter 37, “Advanced Distribution: Configuring Application Dependencies and Chains,” on page 399](#).
- ♦ Report on the success or failure of events such as distributing, caching, and uninstalling the application on workstations. See [Chapter 44, “Reporting Application Events,” on page 441](#).
- ♦ Determine the schedule of when the application is available to users. See [“Schedule Page” on page 552](#).
- ♦ Define many additional Application object properties to control the behavior of the application. For information about all Application object properties, see [Chapter 48, “Reference: Application Object Settings,” on page 481](#).

Distribution: Applications to Terminal Servers

32

Novell® ZENworks® Desktop Management lets you manage applications that are being hosted in a terminal server environment. This includes the ability to both distribute applications to the host terminal servers and then make the hosted applications available to users through Novell Application Launcher™.

This section provides information about how to distribute applications to your host terminal servers. For information about distributing terminal server applications to users, see [Chapter 28, “Distribution: Simple Applications,” on page 325](#) and [Chapter 29, “Distribution: Complex Applications,” on page 331](#).

Distributing applications to a terminal server is much the same as distributing applications to a user or workstation, which means you can follow the processes described in the sections referenced above. However, you should also be aware of the following:

- ◆ You can distribute simple applications, AOT/AXT applications, or MSI applications to a terminal server.
- ◆ You can use user associations or workstation associations to distribute applications.

To use user associations, you must install Application Launcher on the terminal server, create a User object in Novell eDirectory™ for the terminal server, and associate the desired applications with the terminal server's User object.

To use workstation associations, you must install Application Launcher and Workstation Manager on the terminal server, import the terminal server into eDirectory as a Workstation object, and associate the desired applications with the terminal server's Workstation object.

- ◆ For both user-associated applications and workstation-associated applications, you must be logged on at the terminal server console through a Windows administrator account to ensure successful distribution to the terminal server.
- ◆ Application Launcher does not create a NAL cache directory on the terminal server. Because there is no NAL cache directory, the following applies:
 - ◆ Distribution of MSI applications to the terminal server requires the appropriate network client (Microsoft Client or Novell Client™) to be installed on the terminal server. Distribution cannot occur through the ZENworks Middle Tier Server because there is no NAL cache to copy the MSI source package to. A possible workaround to this issue is to create a simple Application object that distributes the `.msi` package to the terminal server and launches the Windows Installer to install the package.
 - ◆ Force cache does not work.
 - ◆ Uninstall does not work. You must manually uninstall the application files.
- ◆ For MSI applications, to ensure that the applications are installed for multisession access, use the `change user /install` command in the pre-distribution script and the `change user /execute` command in the post-distribution script. This causes the application to be installed to the systemroot rather than an individual user's home directory.

- ◆ You can use the Terminal Server distribution rule to ensure that an application is only distributed to a terminal server. For information about the Terminal Server distribution rule, see [“Distribution Rules Page” on page 537](#)
- ◆ If you want an application to be distributed when a user launches it (rather than you installing), the user needs to have appropriate file system and registry access. To install an MSI application on the terminal server, the user must be a member of the Administrators group. Microsoft Windows Installer does not allow non-administrator users to do installations from a terminal client session.

Advanced Distribution: Creating Distribution Rules

33

Novell® ZENworks® Desktop Management lets you create rules to control distribution of an application.

Novell Application Launcher™ or the Workstation Helper applies the rules to the workstation when the application is associated to either the user or the workstation. If the workstation does not meet the distribution rules, the application is not distributed.

You can create distribution rules based on a variety of criteria, such as operating system type and version, available hard disk space, and existence of certain files. Rules can be simple or quite complex. The following sections provide information to help you create rules that best meet your distribution needs:

- ◆ [Section 33.1, “Types of Distribution Rules,” on page 353](#)
- ◆ [Section 33.2, “AND/OR Boolean Operators and Groups,” on page 354](#)
- ◆ [Section 33.3, “Distribution Rules vs. Legacy System Requirements,” on page 355](#)
- ◆ [Section 33.4, “Distribution Rules Examples,” on page 355](#)
- ◆ [Section 33.5, “Creating Distribution Rules,” on page 356](#)
- ◆ [Section 33.6, “Creating Distribution Rules by Importing Legacy System Requirements,” on page 358](#)
- ◆ [Section 33.7, “Maintaining Legacy System Requirements,” on page 362](#)

33.1 Types of Distribution Rules

You can use a variety of rules to determine whether or not an application is distributed to a workstation. These rules can be used individually or joined together to create more complex rules. Below are brief descriptions of each rule:

- ◆ **Applications:** Lets you base distribution on the presence (or absence) of a specific application. The application must have been distributed through an Application object.
- ◆ **Client:** Lets you base distribution on whether the workstation is authenticated to eDirectory through the Novell Client™ or the ZENworks Middle Tier Server.
- ◆ **Connection speed:** Lets you base distribution on the speed of the workstation's network connection.
- ◆ **Disk space:** Lets you base distribution on the amount of free disk space on the workstation.
- ◆ **Environment variables:** Lets you base distribution on the presence (or absence) of specific environment variables.
- ◆ **File date:** Lets you base distribution on the date of a specific file.
- ◆ **File existence:** Lets you base distribution on the presence (or absence) of a specific file.
- ◆ **File size:** Lets you base distribution on the size of a specific file.
- ◆ **File version:** Lets you base distribution on the version of a specific file.

- ◆ **Memory:** Lets you base distribution on the amount of memory on the workstation.
- ◆ **Operating system:** Lets you base distribution on a specific operating system.
- ◆ **Processor:** Lets you base distribution on a specific processor type.
- ◆ **Registry:** Lets you base distribution on the presence (or absence) of specific registry values.
- ◆ **Remote access:** Lets you base distribution on Application Launcher's operation mode, either remote mode or local (LAN) mode.
- ◆ **Terminal server:** Lets you base distribution on the presence (or absence) of Microsoft Terminal Services.

33.2 AND/OR Boolean Operators and Groups

You can use AND/OR Boolean operators to join individual rules or groups of rules to create more flexible distribution conditions.

AND Operator

When using the AND operator, both joined rules must be met for the distribution to occur. For example:

```
Processor type(Processor) >= Pentium III AND System memory(RAM) >= 512
```

For the application to be distributed to the workstation in the above example, both rules must evaluate to true: the workstation must be a Pentium III or better with a minimum of 512 MB RAM.

OR Operator

When using the OR operator, either of the joined rules must be met for the distribution to occur. For example:

```
Processor type(Processor) >= Pentium III OR System memory(RAM) >= 512
```

For the application to be distributed to the workstation in the above example, only one of the rules must evaluate to true: the workstation must be at least a Pentium III, or it must have a minimum of 512 MB RAM.

Groups

You can use both the AND and OR operators to join groups of rules. For example:

```
( Processor type(Processor) >= Pentium III AND System memory(RAM) >=
1024)
OR
( Processor type(Processor) >= Pentium IV AND System memory(RAM) >=
512)
```

For the application to be distributed to the workstation in the above example, only one of the groups must evaluate to true: either the workstation must be a Pentium III with 1 GM of RAM, or it must be a Pentium 4 with 512 MB of RAM.

Nested Groups

You can nest groups two levels deep. In the following example, Group3 and Group4 are nested within Group2:

Group1
OR
(Group2 AND (Group3 OR Group4))

33.3 Distribution Rules vs. Legacy System Requirements

Prior to ZENworks 6.5, distribution rules were known as system requirements and did not include the ability to use both AND/OR Boolean operators and to create groupings of rules. Because of the new operators and groupings, Application Launcher versions prior to ZENworks 6.5 cannot understand distribution rules.

To ensure that pre-ZENworks 6.5 versions of Application Launcher can continue to work in a ZENworks 7 environment, the pre-ZENworks 6.5 system requirements are preserved as legacy system requirements. Therefore, in a mixed-version environment, you need to decide how you want to use distribution rules and legacy system requirements together. You have the following options:

- ◆ **Define distribution rules only:** The ZENworks 6.5 (or later) version of Application Launcher can read the distribution rules. Pre-ZENworks 6.5 versions cannot read the distribution rules and therefore are unable to distribute the application.
- ◆ **Define distribution rules and legacy system requirements:** ZENworks 6.5 (or later) versions use the distribution rules and pre-ZENworks 6.5 versions use the legacy system requirements. This method enables you to maintain the legacy system requirements for pre-ZENworks 6.5 Application Launcher while using the grouping, operator logic, and new rules available beginning with ZENworks 6.5. If you use this method, you can define the legacy system requirements first and then use them to populate the distribution rules. After populating the distribution rules, you can modify them to meet your needs.
- ◆ **Define legacy system requirements only:** Both ZENworks 6.5 (or later) and pre-ZENworks 6.5 versions use the legacy system requirements. This solution reduces administrative maintenance of distribution rules and system requirements, but excludes you from using the new grouping, operators, and rules. Be aware that if you define any distribution rules, ZENworks 6.5 (or later) versions of Application Launcher use the distribution rules, not the legacy system requirements. The ZENworks 7 Application Launcher checks first for the existence of distribution rules; only if no distribution rules exist does it then check for legacy system requirements.

If you have Windows NT 4 workstations in your environment that you are continuing to manage through the use of the ZENworks for Desktops 4.0.1 version of Novell Application Launcher, you need to retain your legacy system requirements. For information about Windows NT 4 support in ZENworks 7, see “[Interoperability with Windows NT 4 Workstations](#)” in “[Interoperability with ZENworks for Desktops 4.x](#)” in “[Interoperability](#)” in the *Novell ZENworks 7 Desktop Management Installation Guide*.

33.4 Distribution Rules Examples

You can use AND and OR Boolean operators when defining the rules that control the distribution of an application. You can also use groups of rules to further control the distribution. Examples of using AND/OR Boolean operators and groups are provided below.

33.4.1 Simple Rule Using an OR Operator

You want to require a workstation to have either a Pentium III processor (or better) or the Windows 2000 operating system. To do so, use the following rule:

```
Processor type(Processor) >= Pentium III OR OS version(Windows NT/2000/XP) >= 5.00.2195
```

To create this rule:

- 1 On the Distribution Rules page, click the *Add* button, then select *Processor*.
- 2 In the Processor Rule dialog box, select *Greater than or equal to (>=)*, select *Pentium III*, then click *OK* to add the rule to the *Add rules to control availability of this application* list.
- 3 On the Distribution Rules page, click the *Add* button, then select *Operating system*.
- 4 In the Operating System Rule dialog box, select *Windows NT/2000/XP*, select *Greater than or equal to (>=)*, enter *5.00.2195*, then click *OK* to add the rule to the *Add rules to control availability of this application* list.

The rule is added to the list with an AND operator.

- 5 In the list, change the AND operator, located in front of the OS Version rule, to an OR operator.

33.4.2 Simple Group

You want to require a workstation to have either a Pentium III processor (or better) or the Windows 2000 operating system with at least 512 MB of memory. To do so, use the following rule:

```
Processor type(Processor) >= Pentium III OR (OS version(Windows NT/2000/XP) >= 5.00.2195 AND System memory(RAM) >= 512)
```

To create this rule:

- 1 On the Distribution Rules page, click the *Add* button, then select *Processor*.
- 2 In the Processor Rule dialog box, select *Greater than or equal to (>=)*, select *Pentium III*, then click *OK* to add the rule to the *Add rules to control availability of this application* list.
- 3 On the Distribution Rules page, click the *Add* button, then select *New group* to add a new group to the *Add rules to control availability of this application* list.

The new group is labeled Group 1, unless you've previously added groups to the list. In addition, it is assigned the AND operator by default.

- 4 Locate Group 1 in the list and change the AND operator to an OR operator.
- 5 Next to Group 1 in the list, click the *Add to Group* button, then select *Operating System*.
- 6 In the Operating System Rule dialog box, select *Windows NT/2000/XP*, select *Greater than or equal to (>=)*, enter *5.00.2195*, then click *OK* to add the rule under Group 1 in the list.
- 7 Next to Group 1 in the list, click the *Add to Group* button, then select *Memory*.
- 8 In the Memory Rule dialog box, select *greater than or equal to (>=)*, enter *512*, then click *OK* to add the rule under Group 1 in the list.

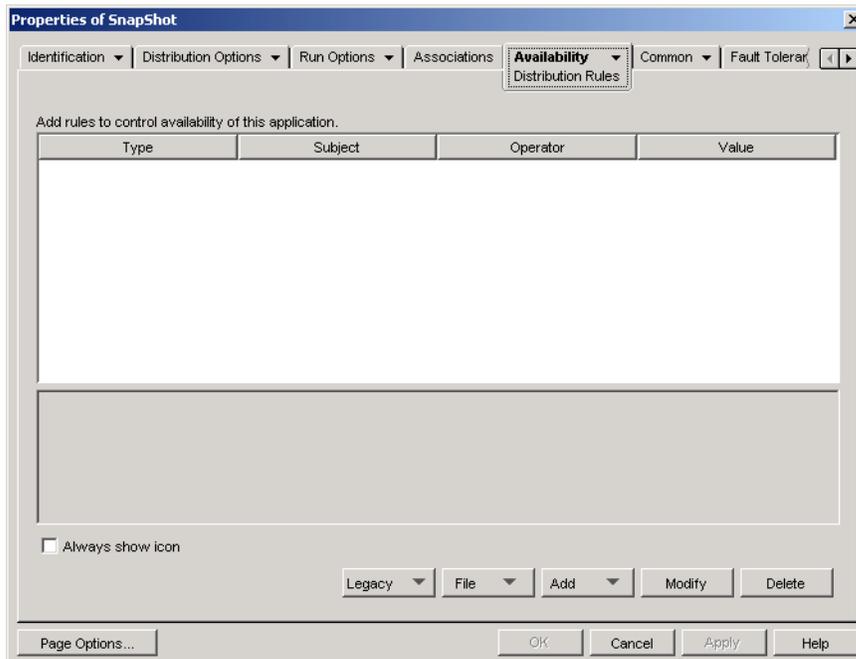
33.5 Creating Distribution Rules

The following steps explain how to create distribution rules that do not have existing legacy system requirements (see [Section 33.3, "Distribution Rules vs. Legacy System Requirements,"](#) on

page 355). If the application has legacy system requirements, you can import them to form the basis of the distribution rules. For instructions, see [Section 33.6, “Creating Distribution Rules by Importing Legacy System Requirements,”](#) on page 358.

- 1 In ConsoleOne[®], right-click the Application object for which you want to create distribution rules, then click *Properties* to display the Application object's property pages.
- 2 Click the *Availability* tab > *Distribution Rules* to display the Distribution Rules page.

If there are legacy system requirements and no distribution rules for the Application object, you are prompted to import the legacy system requirements into the distribution rules. If you want to use the legacy system requirements as the basis for your distribution rules, click *Yes*. Otherwise, click *No*. If you change your mind later, you can use the Import Legacy Settings option on the Distribution Rules page to import the requirements.



- 3 Use the *Add* button to add rules and new groups:
 - ♦ **To add a rule:** Click *Add*, then select the rule to display a dialog box you can use to define the rule. For a brief description of the rules, see [Section 33.1, “Types of Distribution Rules,”](#) on page 353. For more detailed information about a rule, click the *Help* button in the rule's dialog box or see [“Distribution Rules Page”](#) on page 537.
 - ♦ **To add a group:** Click *Add*, then select *New group*.
 - ♦ **To add a rule to a group:** Locate the group in the list, click *Add to group*, then select the rule.
 - ♦ **To add a group to a group:** Locate the group in the list, click *Add to group*, then select *New group*.

For examples of how to use AND/OR operators and groups, see [Section 33.2, “AND/OR Boolean Operators and Groups,”](#) on page 354 and [Section 33.4, “Distribution Rules Examples,”](#) on page 355.

- 4 (Conditional) If you want Application Launcher to display the application's icon even if the workstation does not satisfy the distribution rules, select the *Always show icon* option.

If you select this option and one or more rules is not met, the icon is displayed in a disabled state. Users can right-click the disabled icon, select *Details*, then select *Requirements* to view a list of all of the distribution requirements that were defined for the application.

5 When finished, click *OK* to save your changes.

33.6 Creating Distribution Rules by Importing Legacy System Requirements

You can import an application's legacy system requirements to form the basis of its distribution rules.

If an application has legacy system requirements and no distribution rules, you are prompted to import the legacy system requirements into the distribution rules when you open the Distribution Rules page on the Application object.

If an application has distribution rules already defined, you are not prompted to import the legacy system requirements when you open the Distribution Rule page on the Application object. You can, however, use the Import Legacy Settings option on the Distribution Rules page to import the requirements.

The following sections provide instructions for handling both scenarios:

- ♦ [Section 33.6.1, “Importing the Requirements When Opening the Distribution Rules Page,” on page 358](#)
- ♦ [Section 33.6.2, “Importing the Requirements After Opening the Distribution Rules Page,” on page 360](#)

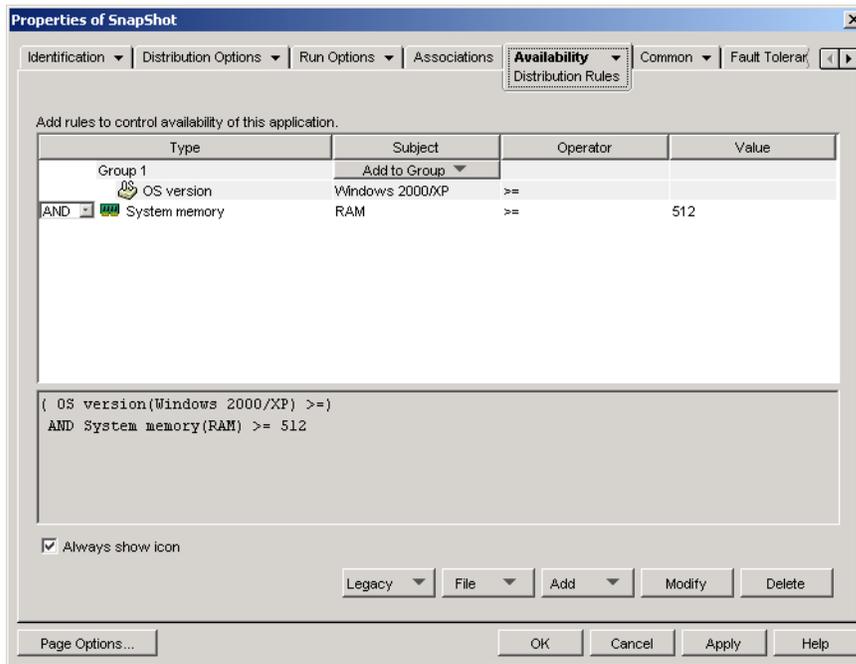
33.6.1 Importing the Requirements When Opening the Distribution Rules Page

- 1 In ConsoleOne, right-click the Application object for which you want to create distribution rules by importing legacy system requirements, then click *Properties* to display the Application object's property pages.
- 2 Click the *Availability* tab > *Distribution Rules* to display the Distribution Rules page.

If there are legacy system requirements and no distribution rules for the Application object, you are prompted to import the legacy system requirements into the distribution rules.



3 Click *Yes* to import the requirements.



4 Modify the distribution rules as desired:

- ♦ **To add a rule:** Click *Add*, then select the rule to display a dialog box you can use to define the rule. For a brief description of the rules, see [Section 33.1, “Types of Distribution Rules,”](#) on page 353. For more detailed information about a rule, click the *Help* button in the rule's dialog box or see [“Distribution Rules Page”](#) on page 537.
- ♦ **To add a group:** Click *Add*, then select *New group*.
- ♦ **To add a rule to a group:** Locate the group in the list, click *Add to group*, then select the rule.
- ♦ **To add a group to a group:** Locate the group in the list, click *Add to group*, then select *New group*.

For examples of how to use AND/OR operators and groups, see [Section 33.2, “AND/OR Boolean Operators and Groups,”](#) on page 354 and [Section 33.4, “Distribution Rules Examples,”](#) on page 355.

5 If you don't want Application Launcher to display the application's icon on workstations that don't satisfy the distribution rules, deselect the *Always show icon* option.

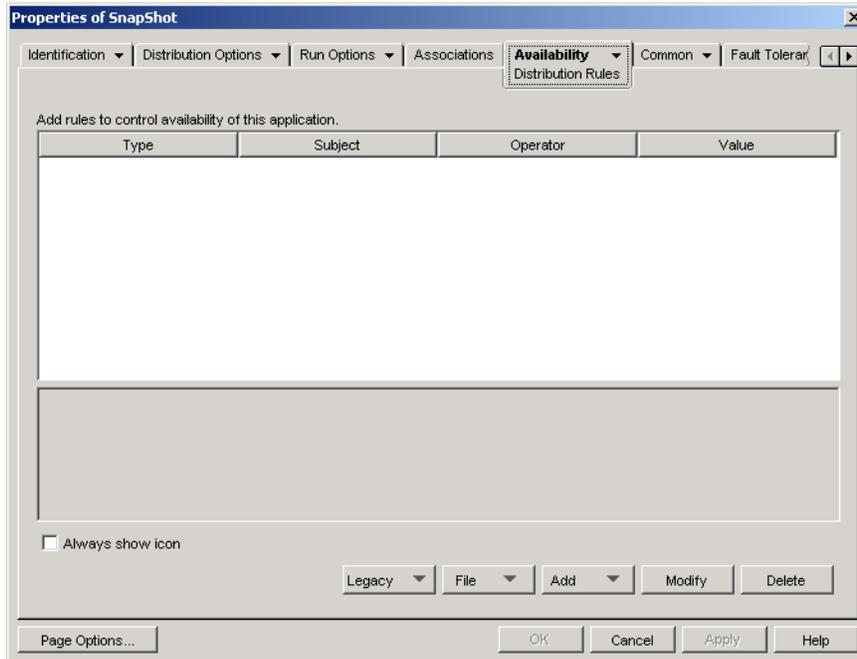
When you imported the legacy system requirements ([Step 3](#)), the *Always show icon* option was automatically turned on, even if it was previously turned off for all legacy system requirements. You might need to deselect the option to maintain the previously-configured behavior.

If you select this option and one or more rules is not met, the icon is displayed in a disabled state. Users can right-click the disabled icon, select *Details*, then select *Requirements* to view a list of all of the distribution requirements that were defined for the application.

6 When finished, click *OK* to save your changes.

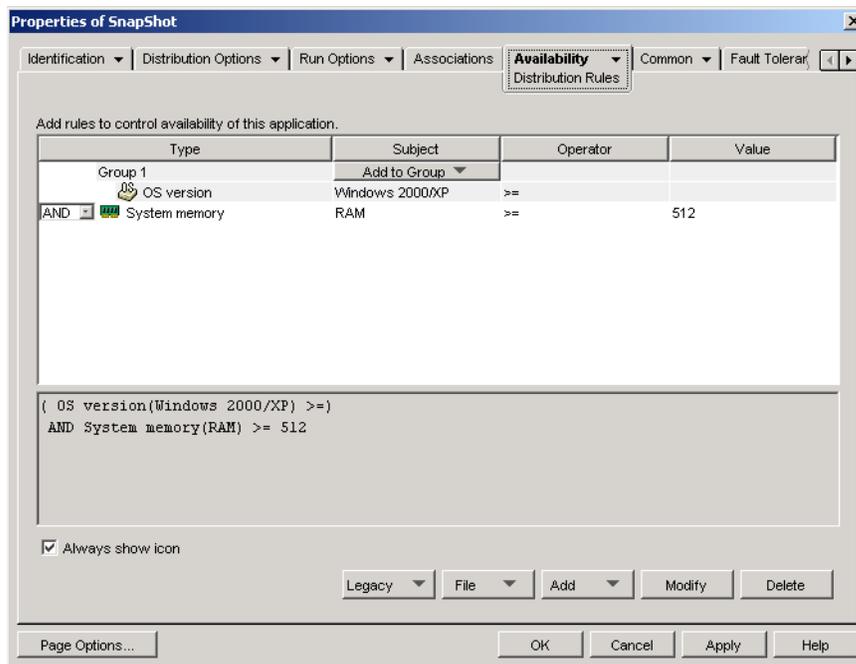
33.6.2 Importing the Requirements After Opening the Distribution Rules Page

- 1 In ConsoleOne, right-click the Application object for which you want to create distribution rules by importing legacy system requirements, then click *Properties* to display the Application object's property pages.
- 2 Click the *Availability* tab > *Distribution Rules* to display the Distribution Rules page.



- 3 Click the *Legacy* button, then click *Import legacy settings*.

If you have already defined distribution rules, you are prompted that they will be overwritten by the legacy system rules. Otherwise, the system requirements are imported without further prompting.



4 Modify the new distribution rules as desired:

- ♦ **To add a rule:** Click *Add*, then select the rule to display a dialog box you can use to define the rule. For a brief description of the rules, see [Section 33.1, “Types of Distribution Rules,” on page 353](#). For more detailed information about a rule, click the *Help* button in the rule's dialog box or see [“Distribution Rules Page” on page 537](#).
- ♦ **To add a group:** Click *Add*, then select *New group*.
- ♦ **To add a rule to a group:** Locate the group in the list, click *Add to group*, then select the rule.
- ♦ **To add a group to a group:** Locate the group in the list, click *Add to group*, then select *New group*.

For examples of how to use AND/OR operators and groups, see [Section 33.2, “AND/OR Boolean Operators and Groups,” on page 354](#) and [Section 33.4, “Distribution Rules Examples,” on page 355](#).

5 If you don't want Application Launcher to display the application's icon on workstations that don't satisfy the distribution rules, deselect the *Always show icon* option.

When you imported the legacy system requirements ([Step 3](#)), the *Always show icon* option was automatically turned on, even if it was previously turned off for all legacy system requirements. You might need to deselect the option to maintain the previously-configured behavior.

If you select this option and one or more rules is not met, the icon is displayed in a disabled state. Users can right-click the disabled icon, select *Details*, then select *Requirements* to view a list of all of the distribution requirements that were defined for the application.

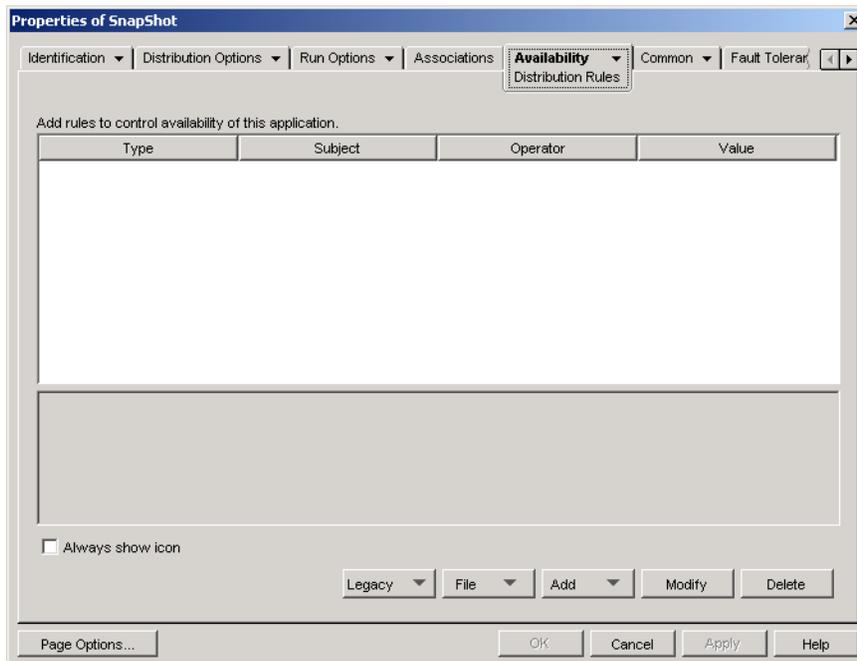
6 When finished, click *OK* to save your changes.

33.7 Maintaining Legacy System Requirements

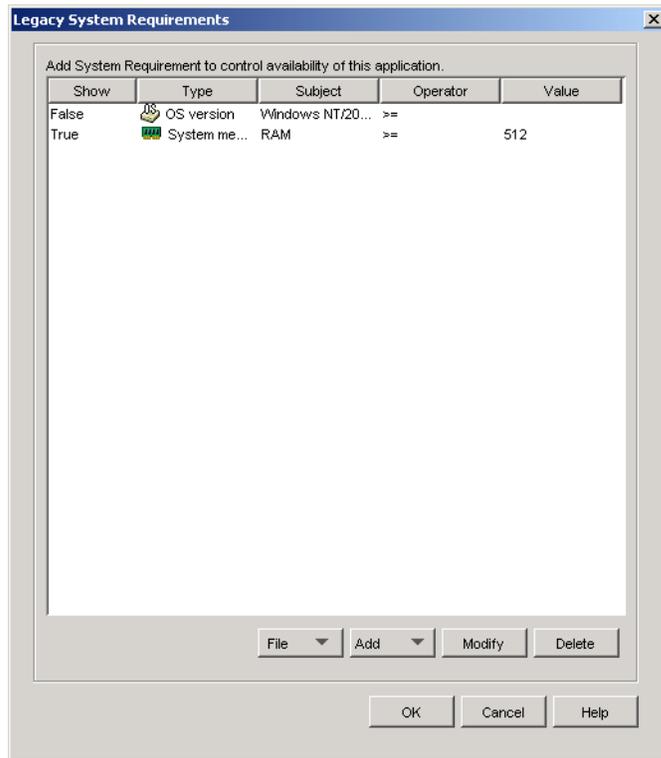
If your environment includes pre-ZENworks 7 versions of Application Launcher, you need to continue to maintain your applications' legacy system requirements in order to support the pre-7 Application Launcher versions. Application Launcher versions prior to ZENworks 7 do not support distribution rules.

To add, remove, or modify an application's legacy system requirements:

- 1 In ConsoleOne, right-click the Application object whose legacy system requirements you to add, remove, or modify, then click *Properties* to display the Application object's property pages.
- 2 Click the *Availability* tab > *Distribution Rules* to display the Distribution Rules page.



- 3 Click the *Legacy* button, then click *Edit legacy settings* to display the Legacy System Requirements dialog box.



- 4 Modify the system requirements as desired:
- ♦ **To add a requirement:** Click *Add*, then select the requirement to display a dialog box you can use to define the requirement. For information about the requirement, click the *Help* button in the requirement's dialog box.
 - ♦ **To remove a requirement:** Select the requirement in the list, then click *Delete*.
 - ♦ **To modify a requirement:** Select the requirement in the list, then click *Modify*. For information about the requirement, click the *Help* button in the requirement's dialog box.
- 5 When finished, click *OK* to save your changes.

Advanced Distribution: Transferring Applications Using BITS

34

This functionality is available beginning with Support Pack 1 of Novell® ZENworks® 7. The Support Pack 1 Desktop Management Agent must be installed on each workstation that will use the Microsoft* Background Intelligent Transfer Service (BITS) to transfer applications to the workstation. Versions of the Desktop Management Agent prior to Support Pack 1 do not support BITS.

Novell Application Launcher™ can use BITS to transfer applications to a workstation. Unlike Application Launcher which competes with other applications for available network bandwidth, BITS uses idle network bandwidth to transfer files, increasing or decreasing the rate at which files are transferred based on the amount of idle network bandwidth available. For example, if a network application begins to consume more bandwidth, BITS decreases its transfer rate.

The following sections provide background information and setup instructions for using BITS. You should complete the sections in the order listed.

1. [Section 34.1, “Understanding How Application Launcher Uses BITS,” on page 365](#)
2. [Section 34.2, “Setting Up Workstations,” on page 367](#)
3. [Section 34.3, “Enabling Application Launcher to Use BITS,” on page 368](#)
4. [Section 34.4, “Configuring an Application to be Transferred by BITS,” on page 370](#)
5. [Section 34.5, “Changing the Default Settings for a BITS Job,” on page 371](#)
6. [Section 34.6, “Changing the BITS Maximum Network Bandwidth Setting,” on page 372](#)

34.1 Understanding How Application Launcher Uses BITS

The following sections provide information to help you understand the interaction between Application Launcher and BITS:

- ◆ [Section 34.1.1, “Application Launcher and BITS Process,” on page 366](#)
- ◆ [Section 34.1.2, “Application Launcher Transfer vs. BITS Transfer,” on page 366](#)
- ◆ [Section 34.1.3, “BITS Job Settings,” on page 366](#)
- ◆ [Section 34.1.4, “BITS Maximum Bandwidth Setting,” on page 367](#)
- ◆ [Section 34.1.5, “Scalability,” on page 367](#)

For additional information about general BITS concepts and functionality, see [Background Intelligent Transfer Service \(http://msdn.microsoft.com/library/default.asp?url=/library/en-us/bits/bits/bits_start_page.asp\)](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/bits/bits/bits_start_page.asp) in the MSDN Library.

34.1.1 Application Launcher and BITS Process

For Application Launcher to use BITS to transfer an application, Application Launcher must be enabled to use BITS as explained in [Section 34.3, “Enabling Application Launcher to Use BITS,” on page 368](#), and the application must be configured properly as explained in [Section 34.4, “Configuring an Application to be Transferred by BITS,” on page 370](#).

After Application Launcher has been BITS enabled and an application configured for BITS, Application Launcher creates a BITS job that contains the application content (files, Application Object settings, etc.) and job settings (Minimum Retry Delay, No Progress Timeout, and Priority) and then queues the job for BITS. BITS transfers the application content to a temporary directory on the local drive (`%systemroot%\temp\%jobid%`, where `%systemroot%` is the Windows system directory and `%jobid%` is the ID of the BITS job). As soon as BITS completes the transfer, Application Launcher copies the application content to the NAL cache and then distributes the application when it is installed or launched.

For specific information about how BITS processes jobs, see [Life Cycle of a BITS Job \(http://msdn.microsoft.com/library/en-us/bits/bits/life_cycle_of_a_bits_job.asp\)](http://msdn.microsoft.com/library/en-us/bits/bits/life_cycle_of_a_bits_job.asp).

34.1.2 Application Launcher Transfer vs. BITS Transfer

Any of the following circumstances will cause Application Launcher to transfer the application itself rather than use BITS:

- ♦ The workstation does not have BITS v2.0 installed.
- ♦ BITS encounters a fatal error during transfer of the application content.
- ♦ BITS encounters a transient error during transfer of the application content and cannot make any progress in the allotted time (**No Progress Timeout** setting).
- ♦ The user launches the application before BITS has completed the transfer. This is the default Application Launcher behavior. However, you can configure Application Launcher to not resume control of the transfer, in which case the user cannot launch the application until BITS finishes the transfer. Or, you can have Application Launcher prompt the user as to whether or not to interrupt the BITS transfer and have Application Launcher immediately download the application. See [“Enabling Application Launcher to Use BITS” on page 368](#).

34.1.3 BITS Job Settings

When Application Launcher creates a job and passes it to BITS, it not only passes the application content (files, Application Object settings, etc.) but also passes three configuration settings:

- ♦ **Minimum Retry Delay:** If BITS encounters an error during transfer of the application, BITS classifies it as a fatal error or a transient error. BITS cannot recover from fatal errors; fatal errors require administrator intervention to resolve the problem. BITS can possibly recover from transient errors. This setting lets you specify the minimum amount of time you want BITS to wait after a transient error occurs before trying to transfer the application again. The default is 600 seconds (10 minutes).
- ♦ **No Progress Timeout:** This setting lets you specify how many days you want BITS to continue to attempt to transfer the application after a transient error has occurred if no progress is made. For example, the default setting (14 days) causes BITS to continue to attempt the transfer for 14 days after the error. If any transfer progress is made during the 14 days, the

counter is reset. If BITS times out because no progress is being made, control of the transfer is returned to Application Launcher, which then transfers the application itself.

- ♦ **Priority:** This setting lets you assign a priority level to the BITS transfer. You can choose from one foreground priority and three background priorities (high, normal, low). Foreground is the highest priority, followed by the three background priorities. The default is Background Normal.

These settings are configurable for each application (Application object > Distribution Options tab > BITS Settings page). You can use the default settings, or you can change the settings based upon application or user needs. These settings are explained in more detail in [Section 34.5, “Changing the Default Settings for a BITS Job,” on page 371](#).

34.1.4 BITS Maximum Bandwidth Setting

Windows XP includes a Group Policy setting for the maximum amount of the workstation’s idle network bandwidth that BITS uses. By default, this setting is not configured, which enables BITS to use all of the workstation’s available idle bandwidth.

If you have BITS running on a large number of workstations, this might impact your network. To reduce this impact, during periods of high utilization you can limit the amount of bandwidth available to BITS. Instructions for using ZENworks to manage the BITS maximum bandwidth setting through the use of a Windows Group Policy are provided in [Section 34.6, “Changing the BITS Maximum Network Bandwidth Setting,” on page 372](#)

IMPORTANT: Windows 2000 does not include a Group Policy setting for BITS maximum bandwidth.

34.1.5 Scalability

If you plan to deliver a large number of applications at one time or deliver applications with a large number of files, you should first review the [BITS scalability guidelines \(http://msdn.microsoft.com/library/en-us/bits/bits/best_practices_when_using_bits.asp\)](http://msdn.microsoft.com/library/en-us/bits/bits/best_practices_when_using_bits.asp) provided by Microsoft.

34.2 Setting Up Workstations

- 1 Make sure the Support Pack 1 Desktop Management Agent is installed on any workstations where you want to use BITS.

The Support Pack 1 Desktop Management Agent must be installed on each workstation that will use BITS to transfer applications to the workstation. Versions of the Desktop Management Agent prior to Support Pack 1 do not support BITS. If necessary, see “[Installing and Configuring the Desktop Management Agent](#)” in the *Novell ZENworks 7 Desktop Management Installation Guide* for instructions about installing the Support Pack 1 Desktop Management Agent.

- 2 Install BITS v2.0 (minimum requirement) on any workstations where you want to use BITS. BITS v2.0 is already included in Microsoft Windows XP SP2. You do not need to install it on Windows XP workstation with SP2 or newer installed.

For workstations with Windows XP, Windows XP SP1, or Windows 2000, you can download BITS v2.0 from [Microsoft Support \(http://support.microsoft.com/default.aspx?kbid=842773\)](http://support.microsoft.com/default.aspx?kbid=842773).

- 3 Make sure BITS is running on the workstations.

BITS is a Windows Service. You can verify that it is running by viewing Services under the Administrative Tools in the Windows Control Panel.

- 4 Make sure that BITS has the required network file system rights to access the application files it will transfer.

BITS requires the same access as Application Launcher to transfer files from a network server. If Application Launcher can transfer files, BITS will be able to. For more information about the required file system access, see [Chapter 23, “Novell Application Launcher: Managing Authentication and File System Access,”](#) on page 293.

- 5 Continue with the next section, [Enabling Application Launcher to Use BITS](#).

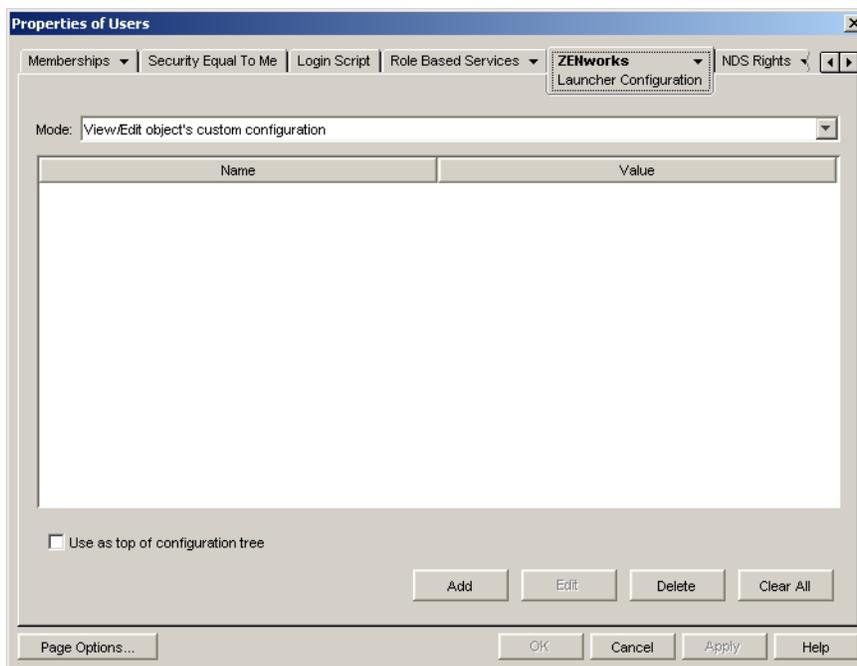
34.3 Enabling Application Launcher to Use BITS

By default, Application Launcher is not configured to use BITS. You must modify the Application Launcher configuration settings in ConsoleOne to enable BITS.

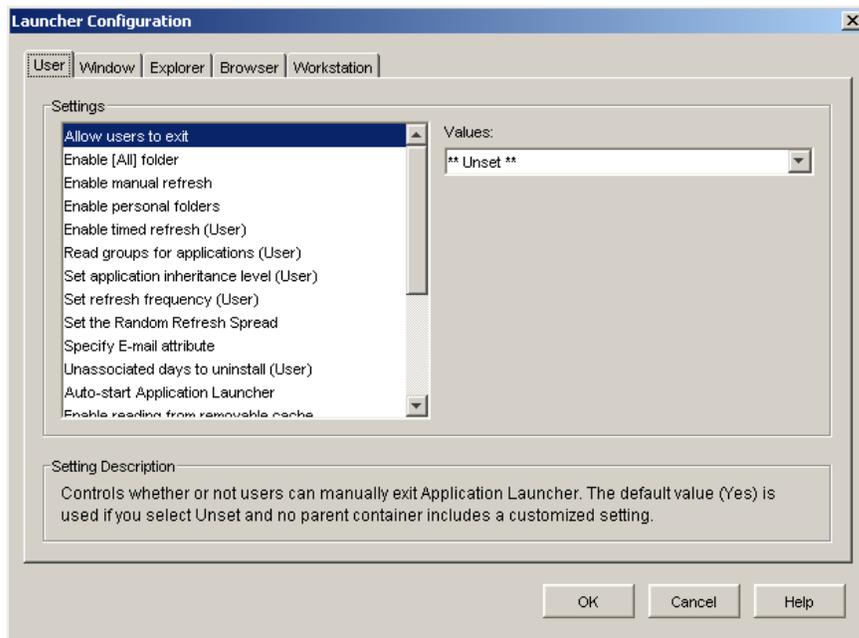
As with all Application Launcher configuration settings, you can modify the BITS settings for a User object, Workstation object, or container (Organization Unit, Organization, Group, etc.) object. For example, if you want to enable BITS for a specific user, you can modify the configuration settings for that User object only. However, if you want to enable BITS for all users in a specific container, you can modify the configuration settings for that container object, in which case all users in the container inherit the container setting.

To enable Application Launcher:

- 1 In ConsoleOne[®], right-click the User, Workstation, or container object where you want to enable BITS, then click *Properties*.
- 2 Click the *ZENworks* tab > Launcher Configuration to display the Launcher Configuration page.



- 3 Click Add to display the Launcher Configuration dialog box.



Depending on the type of object you're configuring, the User tab or Workstation tab might not be available. The User tab is not displayed on Workstation or Workstation Group objects, and the Workstation tab is not displayed on User or User Group objects.

- 4 (Conditional) If you are configuring Application Launcher for a user, user group, or a container with users, open the User tab and configure the following two settings:

Enable BITS: Set this option to Yes to enabled Application Launcher to use BITS to transfer user-associated applications.

Allow User to Override BITS Transfer: Because BITS uses idle network bandwidth to transfer applications, an application might not be available when a user attempts to launch it. Specify the action you want to occur in this case:

- ♦ Yes (default): Causes Application Launcher to cancel the BITS job and immediately transfer the application itself. Rather than starting the transfer over, Application Launcher resumes transfer of the application where BITS left off.
- ♦ No: Causes BITS to maintain control of the transfer; the user is unable to launch the application until the transfer is complete.
- ♦ Prompt: Prompts the user as to whether or not to interrupt the BITS transfer and have Application Launcher immediately download the application.

- 5 (Conditional) If you are configuring Application Launcher for a workstation, workstation group, or a container with workstations, open the Workstation tab and configure the following setting:

Enable BITS (Workstation): Set this option to Yes to enabled Application Launcher to use BITS to transfer workstation-associated applications.

- 6 Click OK to save your changes, then click OK to close the object's properties pages.

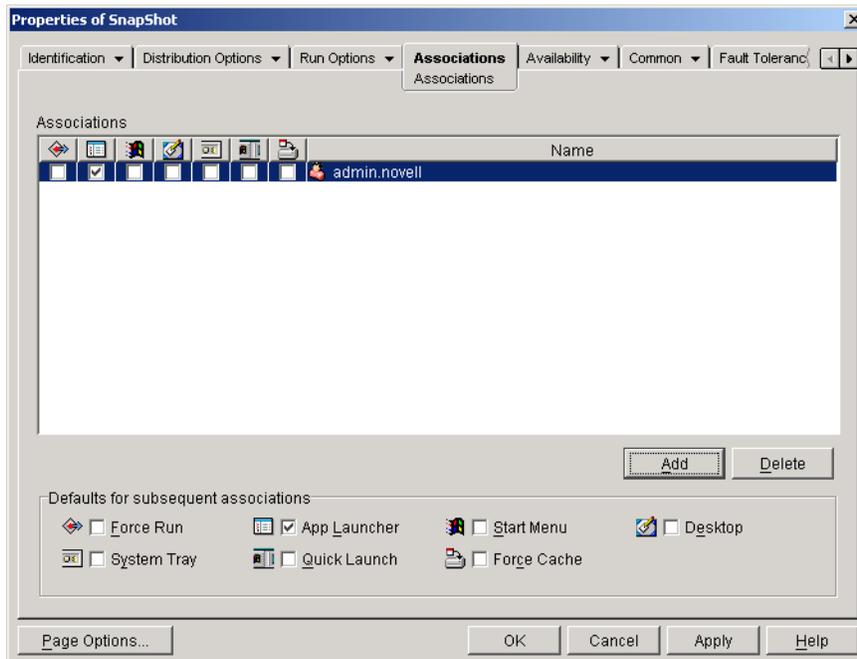
- 7 Continue with the next section, [Configuring an Application to be Transferred by BITS](#).

34.4 Configuring an Application to be Transferred by BITS

Application Launcher uses BITS to transfer an application to a user or workstation only if that user's or workstation's association with the application is marked Force Cache. Otherwise, Application Launcher transfers the application.

To mark an application association as Force Cache:

- 1 In ConsoleOne, right-click the Application object for which you want to configure BITS support, then click *Properties* to display the Application object's property pages.
- 2 Click the *Associations* tab to display the Associations page



- 3 For each association (user or workstation) for which you want to use BITS to transfer the application content, select the association's Force Cache box.

For example, in the above screen shot, if you want to use BITS to transfer the application to the admin.novell user's workstation, you must select the Force Cache  box.

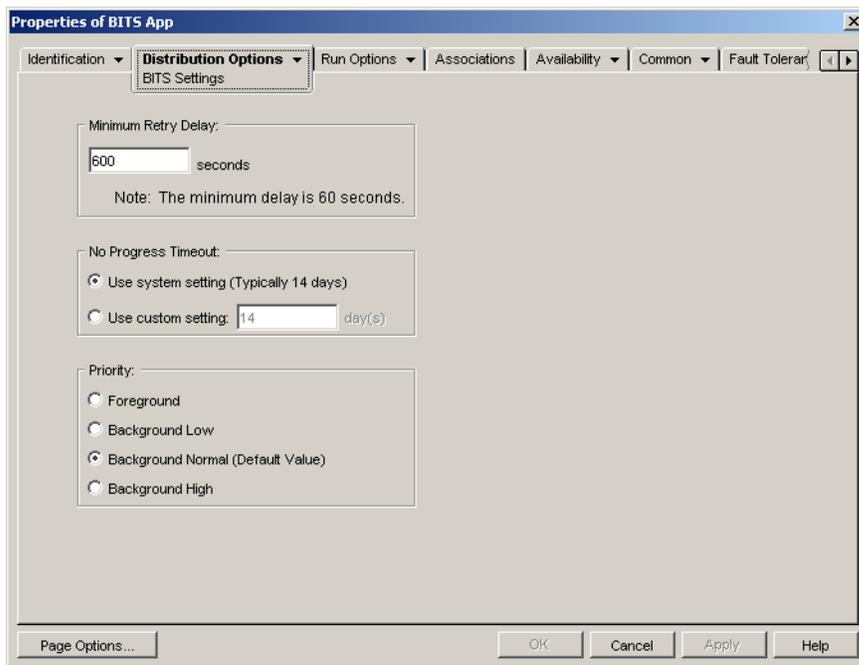
Application Launcher uses BITS only for associations that are marked Force Cache. Application Launcher transfers the application for any associations that are not marked Force Cache.

- 4 If you want to change the settings (Minimum Retry Delay, No Progress Timeout, and Priority) that Application Launcher uses when creating the application's BITS job, skip to the next section, [Changing the Default Settings for a BITS Job](#). Otherwise, click OK to save your changes and close the Application object's property pages.

34.5 Changing the Default Settings for a BITS Job

When using BITS to transfer an application, Application Launcher creates a BITS job that contains the application content (files, Application Object settings, etc.) and job settings (Minimum Retry Delay, No Progress Timeout, and Priority) and then queues the job for BITS. If necessary, you can change the job settings for an application.

- 1 In ConsoleOne, right-click the Application object for which you want to modify the BITS job settings, then click *Properties* to display the Application object's property pages.
- 2 Click the *Distribution Options* tab > *BITS Settings* to display the BITS Settings page.



- 3 Configure the following options as desired:

Minimum Retry Delay: If BITS encounters an error during transfer of the application, BITS classifies it as a fatal error or a transient error. BITS cannot recover from fatal errors; fatal errors require administrator intervention to resolve the problem. BITS can possibly recover from transient errors.

Use this option to specify the minimum amount of time you want BITS to wait after a transient error occurs before trying to transfer the application again. The default is 600 seconds, or 10 minutes. The minimum is 60 seconds. The maximum setting is 2,147,483,647 seconds.

No Progress Timeout: Use this option to specify how many days you want BITS to continue to attempt to transfer the application after a transient error has occurred if no progress is made. For example, a setting of 14 days causes BITS to continue to attempt the transfer for 14 days after the error. If any transfer progress is made during the 14 days, the counter is reset.

- ♦ *Use System Setting (Typically 14 Days):* Select this option to use the Windows system setting. The Windows system setting comes from either 1) the BITS default setting, which is 14 days, or 2) the Timeout (Days) for Inactive Jobs setting in the Windows Group Policy, which is not set by default. If you select this option, the BITS default setting (14

days) is used unless the Windows Group Policy setting has been assigned a value. You can use the Windows Group Policy Editor (gpedit.msc) to view and change the Windows Group Policy setting.

- ♦ *Use Custom Setting:* Select this option to manually enter a timeout period.
 - ♦ The minimum setting is 0 days. Enter 0 only if you do not want BITS to attempt to transfer the application again after it encounters a transient error; in this case, BITS immediately returns control of the transfer to Application Launcher.
 - ♦ The maximum setting is 24,855 days. However, BITS compares this number to the number in the Timeout (Days) for Inactive Jobs setting in the Windows Group Policy. If the Timeout (Days) for Inactive Jobs setting is less than this number, BITS uses the policy setting. For example, if you enter 45 days for this setting, but the policy is set to 30 days, BITS uses 30 days. If the Timeout (Days) for Inactive Jobs setting is undefined (the default state), the policy setting defaults to 90 days. In this case, for example, if you enter 91 days in this setting, BITS uses the policy setting (90 days).

If BITS times out because no progress is being made, control of the transfer is returned to Application Launcher, which then transfers the application itself.

Priority: Use this option to specify the priority assigned to the BITS transfer. You can choose from one foreground priority and three background priorities (high, normal, low).

The foreground priority causes BITS to transfer the application in the foreground. Foreground transfers are the highest priority. Foreground transfers compete for network bandwidth with other applications, which can impede the user's network experience. Unless the timing of the transfer is critical or the user is actively waiting, you should use a background priority.

For the three background priorities, the priority level determines when the transfer is processed relative to other transfers in the queue. Higher priority transfers preempt lower priority transfers. Transfers with the same priority level share transfer time, which prevents a large transfer from blocking the transfer queue. Lower priority transfers do not receive transfer time until all higher priority transfers are completed or in an error state.

All foreground jobs run concurrently with one background job. If there are multiple background jobs with the same priority level (high, for example), BITS splits the background transfer time among the jobs.

- 4 When finished, click OK to save your changes.

34.6 Changing the BITS Maximum Network Bandwidth Setting

Windows XP includes a Group Policy setting for the maximum amount of the workstation's idle network bandwidth that BITS uses. By default, this setting is not configured, which enables BITS to use all of the workstation's available idle bandwidth.

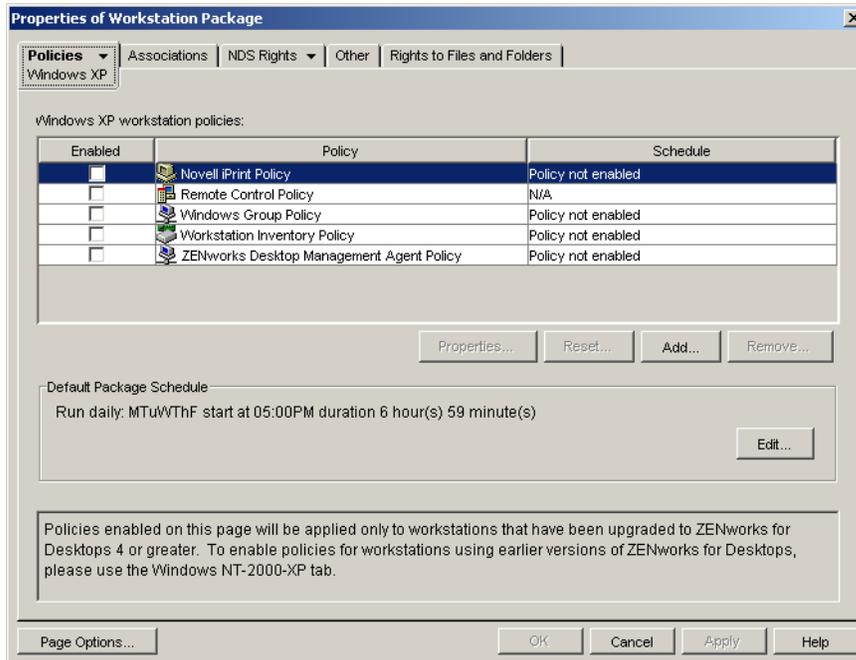
IMPORTANT: The BITS maximum network bandwidth setting is available as a Group Policy on Windows XP only. Windows 2000 does not include a Group Policy setting for BITS maximum bandwidth.

If you have BITS running on a large number of workstations, this might impact your network. To reduce this impact, during periods of high utilization you can limit the amount of bandwidth available to BITS.

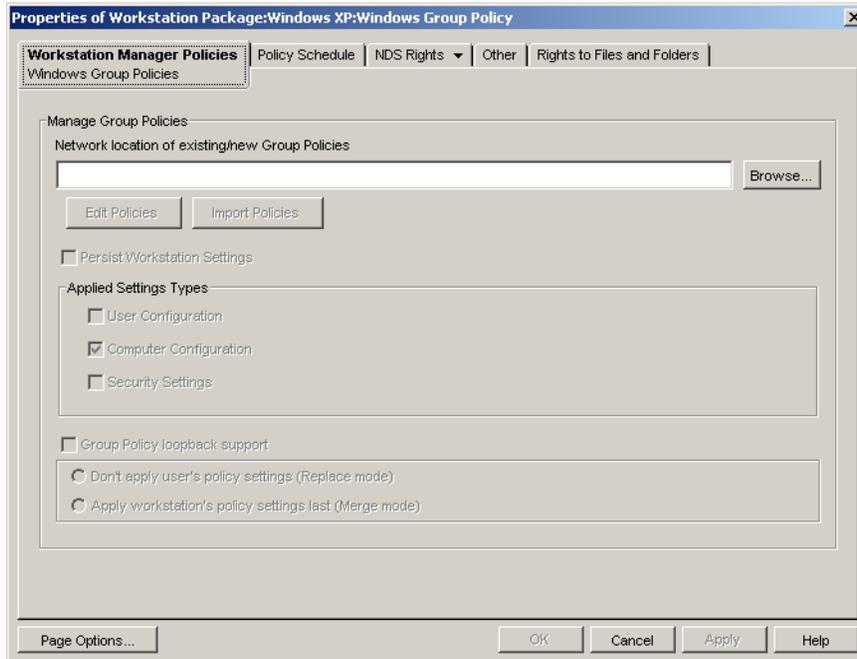
The following steps provide instructions for using a Windows Group Policy to configure the BITS maximum bandwidth setting. The instructions assume that you are familiar with how to use ZENworks policies. If you need additional instructions, see [Chapter 15, “Setting Up User and Workstation Package Policies,”](#) on page 177.

- 1 In ConsoleOne on a Windows XP workstation, right-click the *Workstation Package*, click *Properties*, click the *Policies* tab > *Windows XP*.

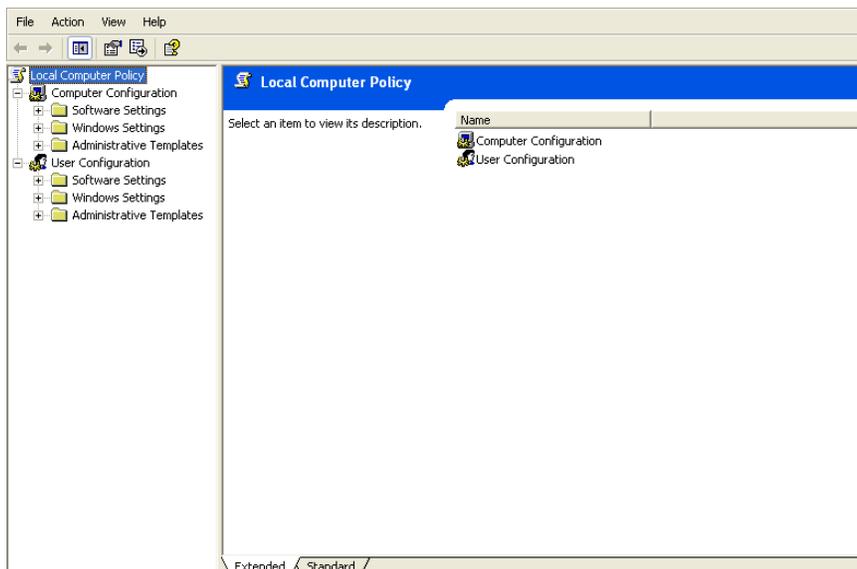
If you do not have a Workstation Package already created, see [Chapter 11, “Creating Policy Packages,”](#) on page 151



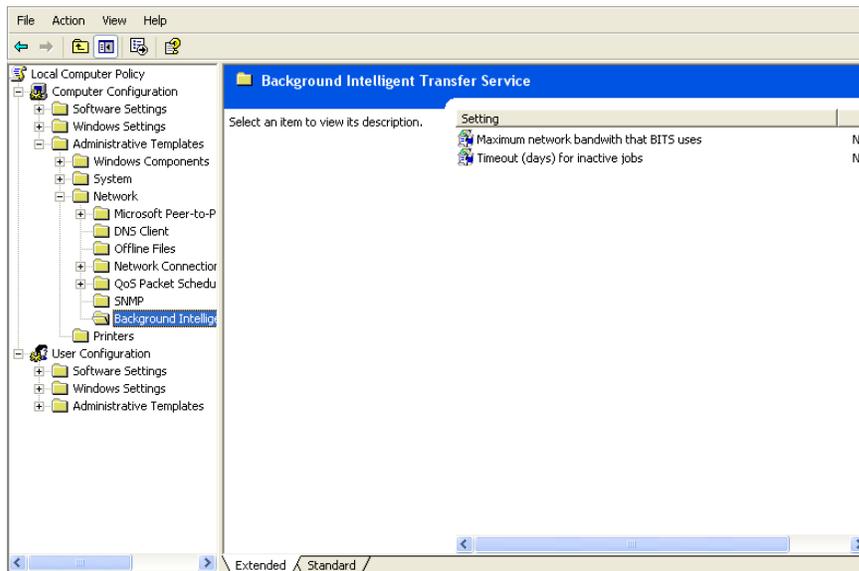
- 2 Enable the *Windows Group Policy*, then click *Properties* to display the Windows Group Policy page.



- 3 Enter a location where you want the new Group Policy file created, or select an existing Group Policy file.
- 4 Click *Edit Policies* to launch the Windows Group Policy Editor.



- Expand the console tree to display the BITS setting (Local Computer Policy > Computer Configuration > Administrative Templates > Network > Background Intelligent Transfer Service).



- Double-click the *Maximum network bandwidth that BITS uses* setting, then configure the settings as desired. Refer to the Help for details about the setting.

Notice that you can also configure the *Timeout (days) for inactive jobs* setting. This setting corresponds to the *No Progress Timeout* setting (configured on the Application object's *BITS Settings* page). It is used if you select the *Use System Setting* option for the *No Progress Timeout* setting (see [Section 34.5, "Changing the Default Settings for a BITS Job,"](#) on [page 371](#)).

- When finished, save your changes and exit the Group Policy Editor.
- Make any additional changes, schedule the policy, and make associations as necessary. For information, see [Chapter 15, "Setting Up User and Workstation Package Policies,"](#) on [page 177](#).

Advanced Distribution: Pre-Installing Applications

35

Novell® ZENworks™ Desktop Management lets you distribute portions of an application to a workstation before the user launches the application the first time. Because you can schedule the distribution, you can perform an off-line, or lights-out, distribution of the application and save the user some of the wait typically associated with a distribution. For example, you could pre-install the application after work hours so the application is ready to use the next day.

With a pre-install, all workstation-related distribution processes (file copying, modifying text files, .ini files, and workstation registry settings) are performed prior to launching of the application. When the user launches the application, the user-specific distribution processes (modifying user registry keys and so forth) are completed.

The following sections provide information for pre-installing applications:

- ♦ [Section 35.1, “User-Associated Applications vs. Workstation-Associated Applications,” on page 377](#)
- ♦ [Section 35.2, “Pre-Installing an Application Immediately,” on page 377](#)
- ♦ [Section 35.3, “Scheduling an Application Pre-Install,” on page 380](#)

35.1 User-Associated Applications vs. Workstation-Associated Applications

You can pre-install an application that is associated with either users or workstations:

- ♦ For user-associated applications, the user must be logged in and Application Launcher must be running. Application Launcher uses the logged-in user’s credentials (authentication and file system access) to distribute the application.
- ♦ For workstation-associated applications, the workstation must be running but Application Launcher does not need to be running.

If the application is a non-MSI application (for example, an AOT application), NAL Workstation Helper uses the workstation's credentials to distribute the application.

If the application is an MSI application, NAL Workstation Helper uses the logged-in user's credentials. If you want it to use the workstation's credentials rather than require a user to be logged in (for example, to perform a lights-out distribution of the MSI application), you must enable the *Distribute in workstation security space if workstation associated* option (*Distribution Options* tab > *Options* page).

35.2 Pre-Installing an Application Immediately

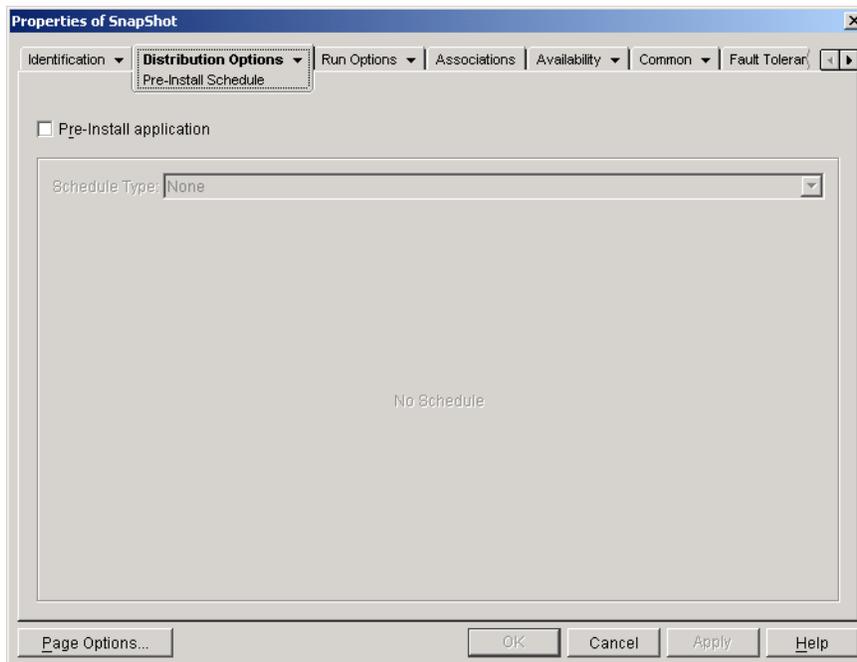
Rather than schedule a pre-install, you can have the pre-install occur as soon as possible.

For users and workstations that are already associated with the application but have not installed it yet, the pre-install occurs the next time Application Launcher refreshes (user-associated application) or NAL Workstation Helper refreshes (workstation-associated application).

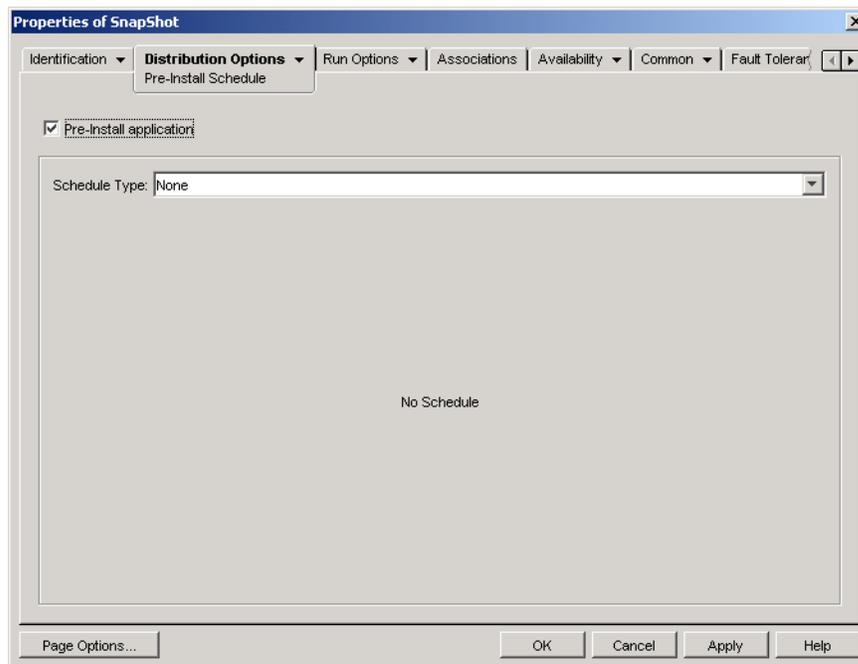
For users and workstations that you associate with the application after enabling pre-install, the pre-install occurs as soon as the association takes place and Application Launcher starts or refreshes (user-associated application) or NAL Workstation Helper starts or refreshes (workstation-associated application).

To enable an application to be pre-installed as soon as possible:

- 1 In ConsoleOne™, right-click the Application object that you want to pre-install, then click *Properties* to display the Application object's property pages.
- 2 Click the *Distribution Options* tab > *Pre-Install Schedule* to display the Pre-Install Schedule page.



- 3 Select the *Pre-install application* check box to enable the option.



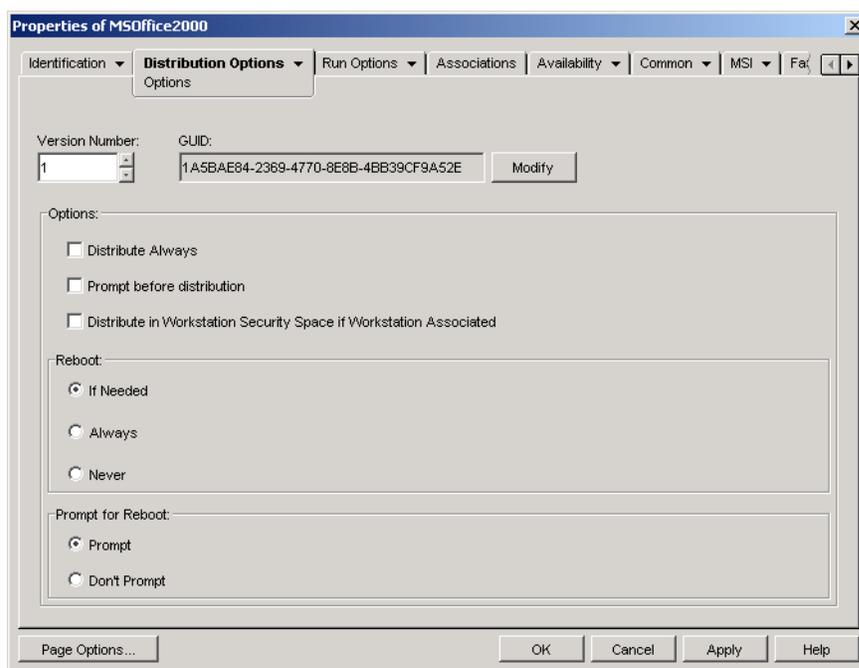
- 4 Leave the *Schedule type* set to *None*.
- 5 (Conditional) If you are pre-installing a workstation-associated MSI application and you want to use the workstation's credentials used rather than the logged-in users, you must enable the *Distribute in workstation security space if workstation associated* option.

By default, workstation-associated MSI applications are distributed in the user security space, meaning that Application Launcher uses the user's credentials and file system access. If you want to perform a lights-out distribution without requiring the user to remain logged in at the workstation, you can have Application Launcher turn over distribution to the NAL Workstation Helper, which runs in the system space and uses the workstation's credentials.

Not all MSI applications can be installed using the workstation's credentials. Some MSI applications have dependencies on a logged-in user (for example, to read and write to the HKCU hive in the Windows registry). In this situation, you must deselect this option in order to have the distribution occur in the user security space and not the workstation security space.

To use enable pre-install using the workstation's credentials:

- 5a** Click the *Distribution Options* tab > *Options* to display the Distribution Options page.



- 5b** Select the *Distribute in workstation security space if workstation associated* check box to enable the option.

It is important to remember that NAL Workstation Helper uses the workstation's credentials, not the user's credentials, to distribute the application. This means that you must assign the workstation the appropriate file system rights to access the network location where the source .msi files reside.

- 5c** (Conditional) If an application requires a reboot during installation, you must select *Reboot if needed* or *Reboot always* in the Reboot group box and *Don't prompt in the prompt* for Reboot group box.

- 6** When finished, click *OK* to save your changes.

35.3 Scheduling an Application Pre-Install

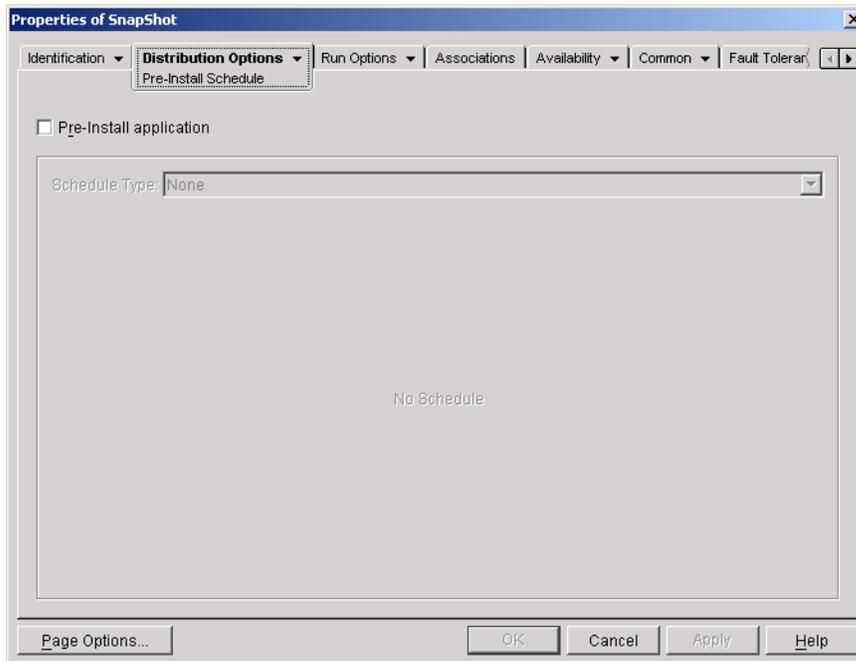
Scheduling a pre-install enables you to perform an off-line, or lights-out, distribution of the application and save the user some of the wait typically associated with a distribution. For example, you could pre-install an application after work hours so the application is ready to use the next day.

For user-associated applications to be pre-installed at the schedule time, Application Launcher must be running on the user's workstation

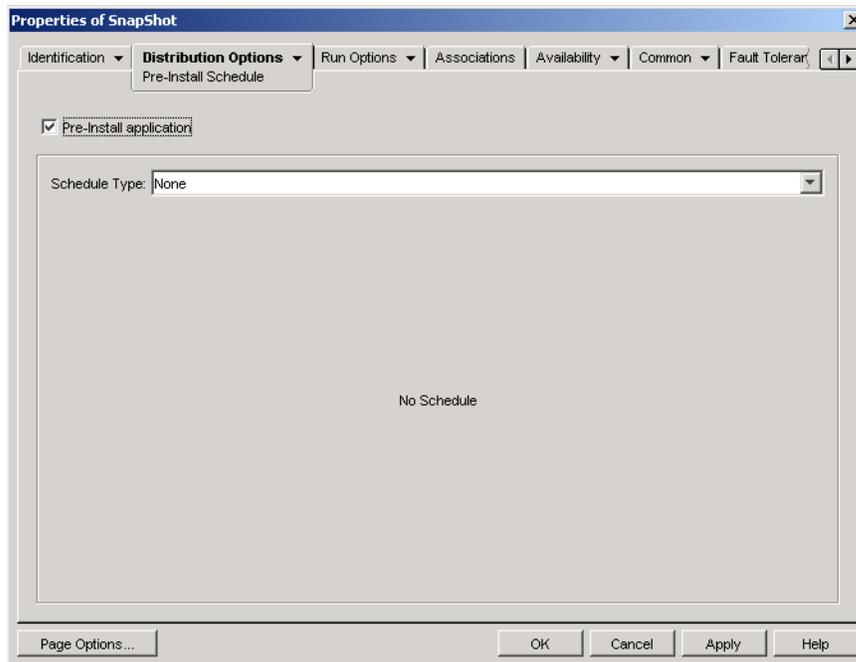
For workstation-associated applications to be pre-installed at the scheduled time, Application Launcher does not need to be running. Pre-install tasks are handled by the NAL Workstation Helper, running under the Workstation Manager.

- 1** In ConsoleOne, right-click the Application object that you want to pre-install, then click *Properties* to display the Application object's property pages.

- 2 Click the *Distribution Options* tab > *Pre-Install Schedule* to display the Pre-Install Schedule page.



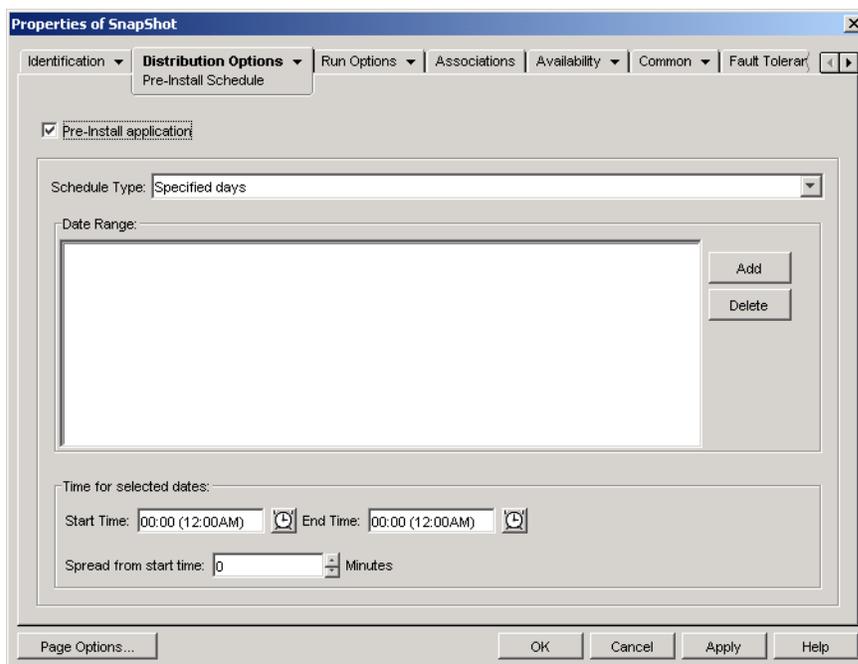
- 3 Select the *Pre-install application* check box to enable the option.



There are two schedule types for you to choose from: *Specified days* and *Range of days*.

- ♦ **Specified days:** Use this schedule type to designate specific days and times when the application can be pre-installed. You can specify up to 350 different days and times.

- ♦ **Range of days:** Use this schedule type to designate a range of days each week that the application can be pre-installed.
- 4 (Conditional) To use the specified days schedule, complete **Step 4a** through **Step 4c**. Otherwise, to use the range of days schedule, skip to **Step 5**.
- 4a In the *Schedule type* list, select *Specified days* to display the Specified Days options.



4b Fill in the following fields:

- ♦ **Date range:** The Date Range list displays all dates when the application can be pre-installed. To add a date, click Add, select the date you want, then click OK to display it in the list.
- ♦ **Time for selected dates:** Select the availability start time and end time. The times apply to all dates in the *Date range* list.

NOTE: The time increments in 5 minute intervals, with the earliest available start time being 00:00 (12:00 a.m.) and the latest end time being 23:55 (11:55 p.m.). This means there is always a 5-minute time period from 11:55 p.m. to 12:00 midnight when the application is unavailable. If you want the application to be available the entire day, you need to use the *Range of days* schedule type. For more information, see [Step 5b on page 384](#).

- ♦ **Spread from start time (in minutes):** The *Spread from start time* option spreads out user access times over the number of minutes specified so the application doesn't become available to all users at the same time. If you anticipate all users launching the application as soon as it becomes available and the application is being distributed or run from the network, you can use this option to avoid possible network overload.

For example, if you have a moderate number of users to whom the application is to be distributed (say about 100), you might specify a one-hour (60 minute) block of time (starting at the scheduled start time) to randomly distribute the application: thus

all users will gain access to the application some time during the first sixty minutes after the scheduled start time.

If you want to substantially ease the load on your servers caused by the application distribution, or if you have bandwidth concerns, you might want to make the application distribute randomly throughout the time of availability. To spread out access times across the entire time (*Specified days* and *Time for selected dates*) that the application is available, use the total availability time specified for that application in terms of minutes. This will require that you make the maximum time available for each day you specify. For example, if an application is configured for a typical business day in the United States (9 hours per day: 8:00 a.m. to 5:00 p.m.), you calculate the total time of availability for that application like this:

Number of specified hours x 60 minutes per hour = Total availability time per day

Using this equation, the example above would be calculated like this:

9 x 60 (minutes per hour) = 540 minutes of availability per day

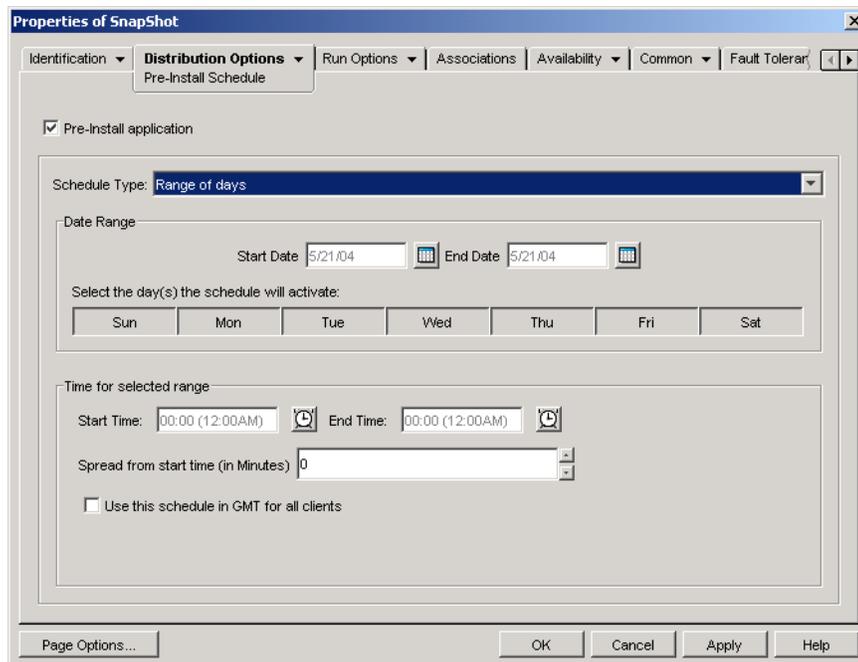
In this example, when you enter 540 minutes in the *Spread from start time* field, the application is distributed randomly for the entire 540 minutes that you have made it available on that scheduled day. Note that this might not be suitable for applications that must be distributed in a timely fashion, such as anti-virus updates. Note also that this is an example only: you can schedule the distribution for any specified amount of time for any day of the week.

Remember that the *Time for selected dates* setting makes the last five minutes of a day un-scheduleable, so you need to consider these five minutes if the application schedule ends at 11:55 p.m. for that day.

4c Continue with **Step 6**.

5 (Conditional) To use the *Range of days* schedule, complete **Step 5a** through **Step 5c**. Otherwise, to use the *Specified days* schedule, return to **Step 4**.

5a In the *Schedule type* list, select *Range of days* to display the *Range of days* options.



5b Fill in the following fields:

- ♦ **Date range:** To define the range of days, select a start date and an end date, then select the days (Sunday through Saturday) within the established date range. By default, all days are selected; a day is selected when the button appears to be pressed in.
- ♦ **Time for selected range:** Select the availability start time and end time. This option works differently depending on whether the date range includes one day, multiple days, or all seven days. If the date range includes one to six days (but not all seven days), the application is available between the start and end times on those days. For example, if you make the application available on Monday between 8:00 and 5:00, it is available during those hours. However, if the date range includes all seven days, the times are ignored and the application is available every day, 24 hours a day.
- ♦ **Spread from start time (in minutes):** The *Spread from start time* option spreads out user access times over the number of minutes specified so the application doesn't become available to all users at the same time. If you anticipate all users launching the application as soon as it becomes available and the application is being distributed or run from the network, you can use this option to avoid possible network overload.

For example, if you have a moderate number of users to whom the application is to be distributed (say about 100), you might specify a one-hour (60 minute) block of time (starting at the scheduled start time) to randomly distribute the application: thus all users will gain access to the application some time during the first sixty minutes after the scheduled start time.

If you want to ease the load of the application distribution on your servers or if you have bandwidth concerns, you might want to make the application distribute randomly throughout the time of availability. To spread out access times across the entire time (*Date range* and *Time for selected dates*) that the application is available, use the total availability time specified for that application in terms of minutes. For example, if a workstation-associated application is configured for an entire 24-hour, three-shift day, you can calculate the total time of availability for that application like this:

Number of days in date range x *Time of availability per day* = Total availability time

Using this equation, and making sure to convert hours to minutes, the example above would be calculated like this:

7 (days) x 24 (hours) = 168 hours of availability

168 x 60 (minutes per hour) = 10,080 minutes of availability

When you enter 10800 minutes in the *Spread from start time* field, the application is distributed randomly for the entire 10800 minutes that you have made it available.

Note that this is not suitable for applications that must be distributed in a timely fashion, such as anti-virus updates.

- ♦ **Use this schedule in GMT for all clients:** The schedule is based on the workstation's time zone. If your network spans different time zones and you schedule an application to run at 1:00 p.m., it runs at 1:00 p.m. in each time zone. You can select this option to have workstations run applications at the same time regardless of their time zones (for example, 1:00 p.m. Rome time and 4:00 a.m. Los Angeles time).

5c Continue with **Step 6**.

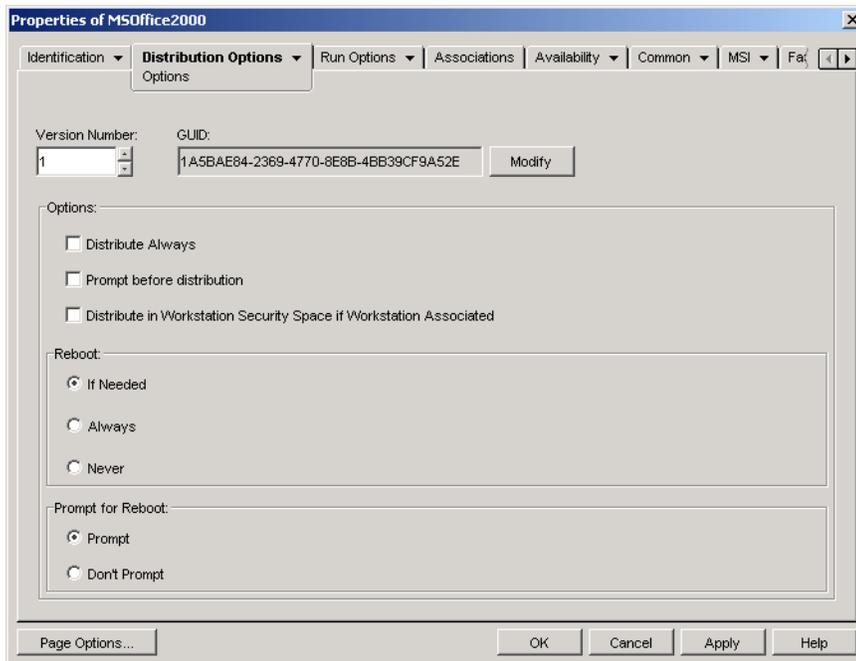
- 6** (Conditional) If you are pre-installing a workstation-associated MSI application and you want to use the workstation's credentials used rather than the logged-in users, you must enable the *Distribute in workstation security space if workstation associated* option.

By default, workstation-associated MSI applications are distributed in the user security space, meaning that Application Launcher uses the user's credentials and file system access. If you want to perform a lights-out distribution without requiring the user to remain logged in at the workstation, you can have Application Launcher turn over distribution to the NAL Workstation Helper, which runs in the system space and uses the workstation's credentials.

Not all MSI applications can be installed using the workstation's credentials. Some MSI applications have dependencies on a logged-in user (for example, to read and write to the HKCU hive in the Windows registry). In this situation, you must deselect this option in order to have the distribution occur in the user security space and not the workstation security space.

To enable pre-install using the workstation's credentials:

- 6a** Click the *Distribution Options* tab > *Options* to display the Options page.



- 6b** Select the *Distribute in workstation security space if workstation associated* check box to enable the option.

It is important to remember that NAL Workstation Helper uses the workstation's credentials, not the user's credentials, to distribute the application. This means that you must assign the workstation the appropriate file system rights to access the network location where the source `.msi` files reside.

If an application requires a reboot during installation, you must select *Reboot if needed* or *Reboot always* in the *Reboot* group box and *Don't Prompt* in the *Prompt for reboot* group box.

- 7** Click *OK* to save your changes.

Advanced Distribution: Configuring Fault Tolerance, Load Balancing, and Site Lists

36

Novell® ZENworks® Desktop Management provides a variety of methods to help you ensure that critical applications are always available for distribution to users. These methods are discussed in the following sections:

- ♦ [Section 36.1, “Setting Up Fault Tolerance,” on page 387](#)
- ♦ [Section 36.2, “Setting Up Load Balancing,” on page 391](#)
- ♦ [Section 36.3, “Setting Up Site Lists,” on page 395](#)

36.1 Setting Up Fault Tolerance

Network problems can make installation packages unavailable. To ensure that an application can be distributed even when its installation package is unavailable, you can establish backup installation packages. How you do so depends on the type of application, as explained in the following sections:

- ♦ [Section 36.1.1, “Setting Up Fault Tolerance for MSI Applications,” on page 387](#)
- ♦ [Section 36.1.2, “Setting Up Fault Tolerance for Simple or AOT/AXT Applications,” on page 388](#)

ZENworks Desktop Management does not provide fault tolerance for Web applications or terminal server applications.

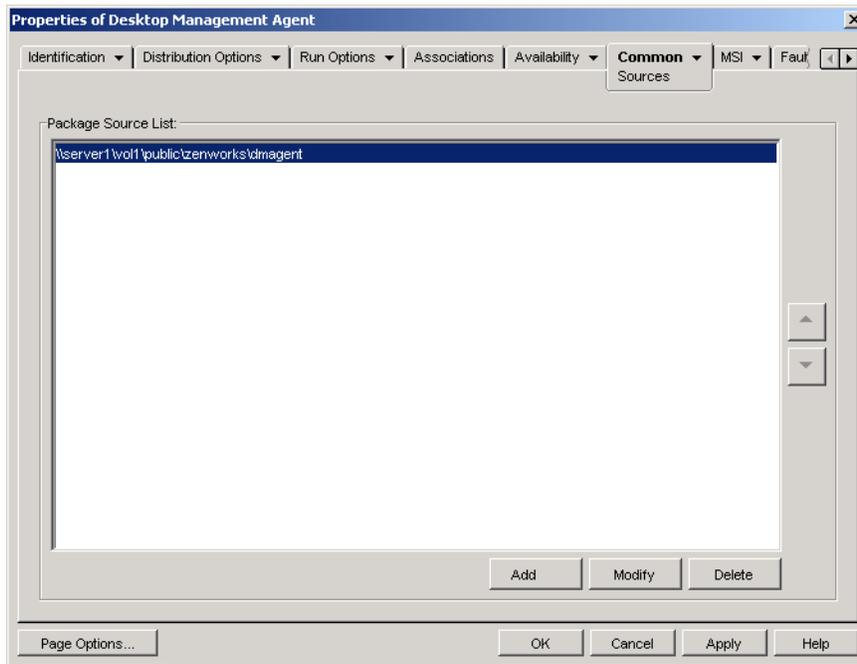
36.1.1 Setting Up Fault Tolerance for MSI Applications

You establish fault tolerance, or source resiliency, for an MSI application by having multiple installation packages available at different server locations. During distribution, if the first source fails, Application Launcher attempts to distribute from an alternate source.

To set up fault tolerance for an MSI application:

- 1** Create the additional MSI installation packages in the server locations you want.
- 2** In ConsoleOne®, right-click the MSI Application object for which you want to establish fault tolerance, then click *Properties* to display the Application object's property pages.

- 3 Click the *Common* tab > *Sources* to display the Sources page.



By default, the Sources page displays the source location that was specified when the MSI Application object was created.

- 4 Click *Add*, specify the source path for an MSI installation package, then click *OK* to add the additional source to the list.
- 5 Repeat **Step 4** until you've added all available sources.
- 6 Use the arrow buttons to list the sources in the order you want them used.
The first source listed is used first, the second is used next, and so forth until either the distribution is successful or all sources have been used.
- 7 Click *OK* to save your changes.

36.1.2 Setting Up Fault Tolerance for Simple or AOT/AXT Applications

You establish fault tolerance for a simple application or an AOT/AXT application by having multiple installation packages at different source locations. During distribution, if the application's first source fails, Application Launcher attempts to distribute from an alternate source.

When setting up fault tolerance for an application, you can link directly to additional installation packages or you can link to additional Application objects that have been created using the installation packages. For example, you want to set up fault tolerance for *App1*, whose installation package is located at `\\server1\vol1\public\zenworks\app1`. You create three additional source locations:

```
\\server2\vol1\public\zenworks\app1  
\\server3\vol1\public\zenworks\app1  
\\server4\vol1\public\zenworks\app1
```

You can define each of the additional installation packages as sources for App1, or you can create three new Application objects using each of the installation packages and then have App1 link to the three new Application objects.

Defining each of the additional installation packages as sources for App1 has the advantage of requiring configuration of only one Application object. Regardless of which installation package is used, the App1 object's distribution settings are applied.

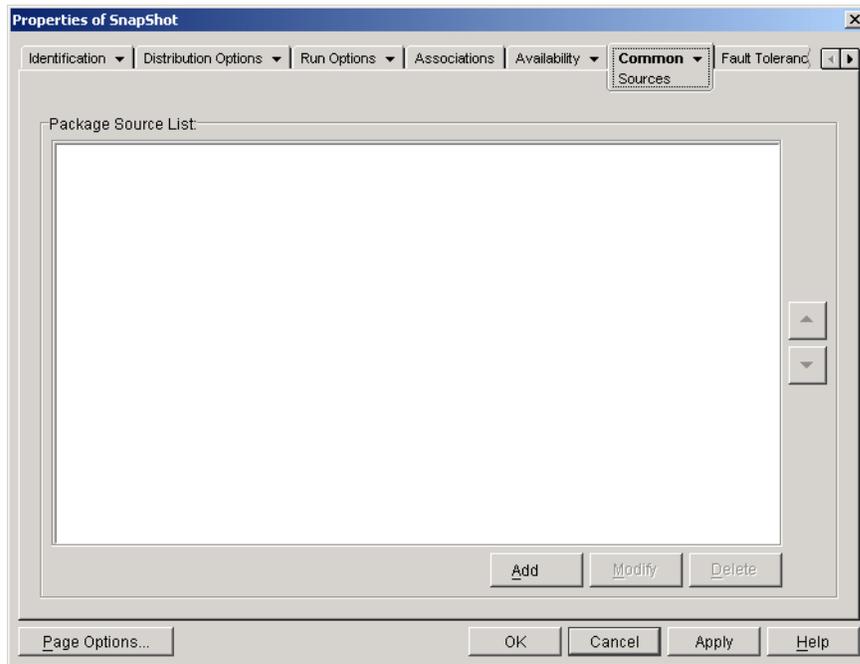
Defining additional Application objects using the installation packages enables you to have different Application object configuration settings for each installation package. If you choose to use additional Application objects rather than additional sources, please be aware of the following limitations:

- ◆ Only one-level-deep fault tolerance is supported. If Application Launcher fails over to a backup Application object and the backup Application object fails, the distribution fails even if fault tolerance has been enabled for the backup Application object.
- ◆ *Remote mode* and *Force cache* distributions do not support fault tolerance through Application objects. Both of these distributions require Application Launcher to copy installation package files to the workstation cache and then install from the cache, which requires the use of additional sources rather than additional Application objects.

To set up fault tolerance for a simple application or an AOT/AXT application:

- 1** Copy the additional installation packages to the server locations you want.
- 2** (Conditional) If you plan to create Application objects for each of the additional installation packages so that you can link to the Application objects, create the Application objects. If necessary, refer to [Section 28.2, “Configuring the Application in eDirectory,” on page 326](#).
- 3** In ConsoleOne, right-click the Application object for which you want to establish fault tolerance, then click *Properties* to display the Application object's property pages.
- 4** (Conditional) If you are using installation packages only (no Application objects for them), define the installation packages as additional sources for the application. To do so:

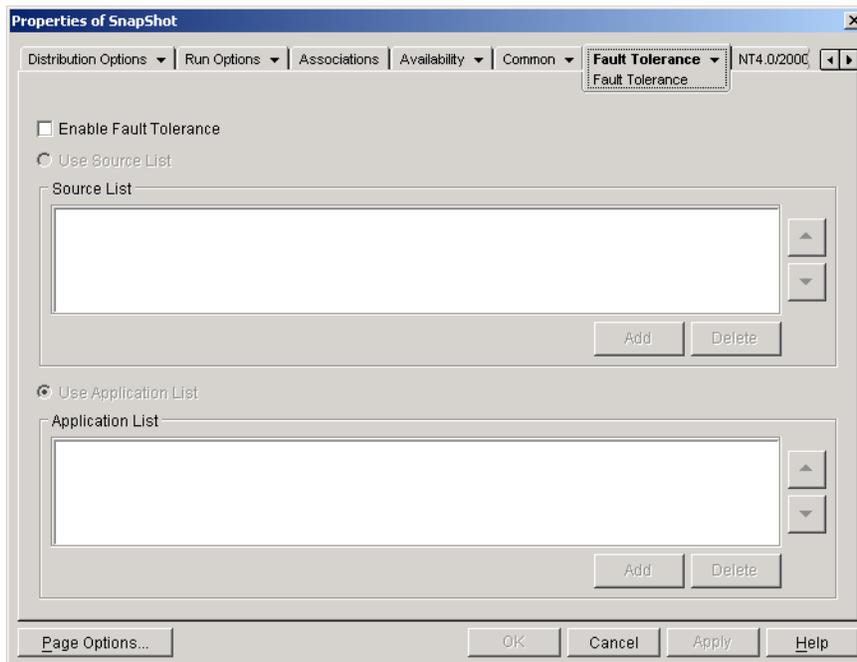
4a Click the *Common* tab > *Sources* to display the Sources page.



4b Click *Add*, specify the source path for an installation package, then click *OK* to add the additional source to the list.

4c Repeat **Step 4b** until you've added all available sources.

5 Click the *Fault Tolerance* tab > *Fault Tolerance* to display the Fault Tolerance page.



6 Fill in the following fields:

Enable fault tolerance: Select this option to enable fault tolerance. The other fault tolerance options are available only if this option is enabled.

Use source list: Select this option to have Application Launcher use a list of sources as the backup. You must have already created at least one installation package source (see [Step 4](#)).

To add a source to the Source List, click *Add*, browse to and select the source, then click *OK*.

To remove a source, select the source from the Source List, then click *Delete*.

If distribution of the application fails, Application Launcher tries the alternate sources in the order they are listed, from top to bottom. To change the order of the sources, select a source from the Source List, then click the up-arrow to move the source up in the list or click the down-arrow to move it down in the list.

Use application list: Select this option to have Application Launcher use Application objects as the backup installation packages. You must have already created additional Application objects for the application, with the installation package being stored on a different server or volume than this application (see [Step 2](#)).

To add an Application object to the Application List, click *Add*, browse to and select the Application object, then click *OK*.

To remove an Application object from the list, select the Application object, then click *Delete*.

If distribution of the application fails, Application Launcher tries the Application objects in the order they are listed, from top to bottom. To change the order of the list, select an Application object from the Application List, then click the up-arrow to move the Application object up in the list or click the down-arrow to move it down in the list.

7 Click *OK* to save your changes.

36.2 Setting Up Load Balancing

Distributing an application to many users at the same time can cause delays with the distribution. To reduce the possibility of delays, you can spread the workload across multiple servers by establishing additional installation packages on each of those servers. How you set up load balancing depends on whether the application is an MSI application or a simple/AOT/AXT application, as explained in the following sections:

- ◆ [Section 36.2.1, “Setting Up Load Balancing for MSI Applications,” on page 391](#)
- ◆ [Section 36.2.2, “Setting Up Load Balancing for Simple or AOT/AXT Applications,” on page 393](#)

ZENworks Desktop Management does not provide load balancing for Web applications or terminal server applications.

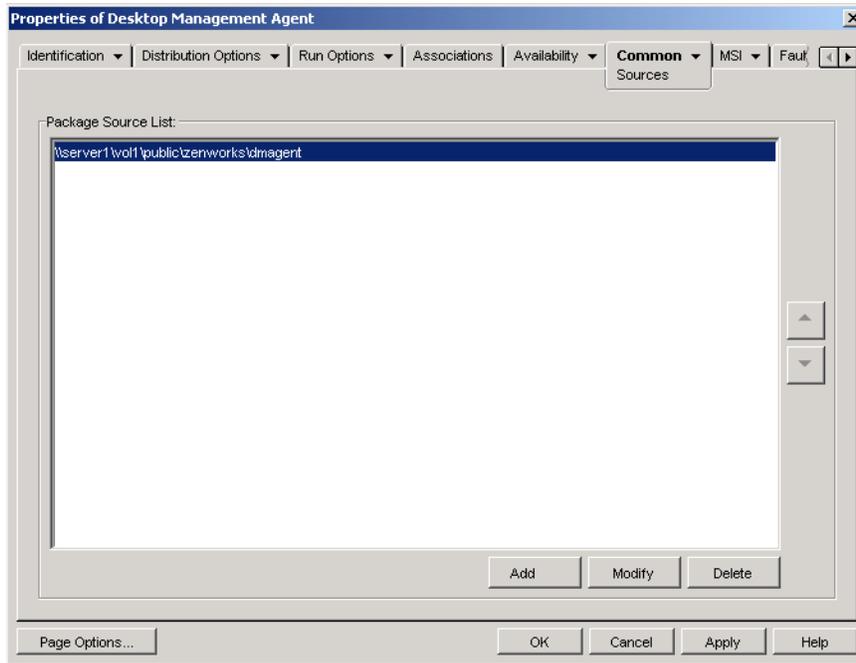
36.2.1 Setting Up Load Balancing for MSI Applications

You establish load balancing for an MSI application by having multiple installation packages available at different server locations. During distribution, if the first source is busy, Application Launcher attempts to distribute from an alternate source.

To set up load balancing for an MSI application:

- 1 Create the additional MSI installation packages in the server locations you want.

-
- 2 In ConsoleOne, right-click the MSI Application object for which you want to establish load balancing, then click *Properties* to display the Application object's property pages.
- 3 Click the *Common* tab > *Sources* to display the Sources page.



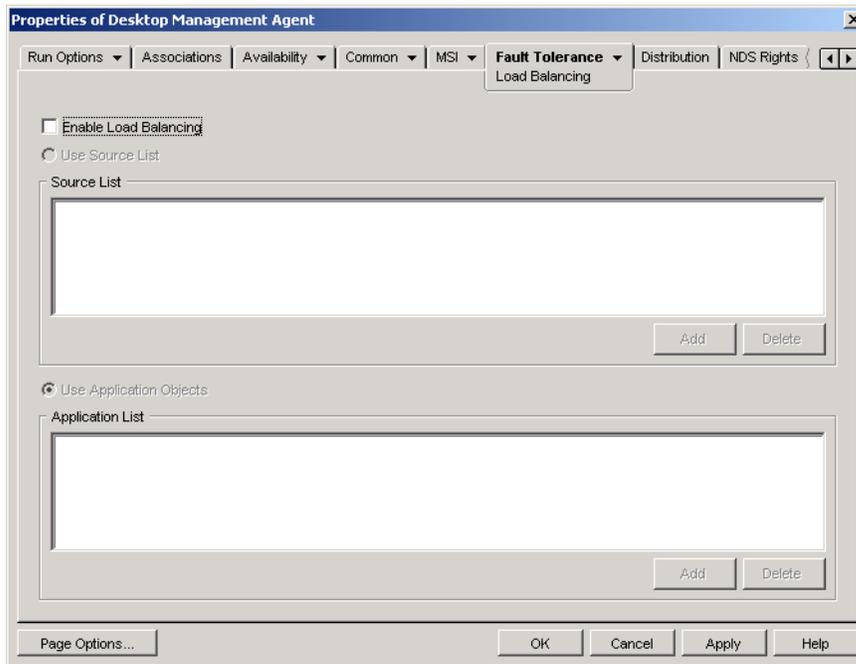
By default, the Sources page displays the source location that was specified when the MSI Application object was created.

-
-
-
- 4 Click *Add*, specify the source path for an MSI installation package, then click *OK* to add the additional source to the list.
- 5 Repeat **Step 4** until you've added all available sources.

For load balancing, the order in which the sources are listed does not matter. Application Launcher randomly selects the source location.

If you are also using this list for fault tolerance, the list determines the order in which sources are used. For more information, see [Section 36.1.1, “Setting Up Fault Tolerance for MSI Applications,” on page 387.](#)

6 Click the *Fault Tolerance* tab > *Load Balancing* to display the Load Balancing page.



7 Select the *Enable load balancing* check box to turn on the option.

8 Click *OK* to save your changes.

36.2.2 Setting Up Load Balancing for Simple or AOT/AXT Applications

You establish load balancing for a simple application or an AOT/AXT application by having multiple installation packages at different source locations. During distribution, if the application's first source is busy, Application Launcher attempts to distribute from an alternate source.

When setting up load balancing for an application, you can link directly to additional installation packages or you can link to additional Application objects that have been created using the installation packages. For example, you want to set up load balancing for App1, whose installation package is located at `\\server1\vol1\public\zenworks\app1`. You create three additional source locations:

```
\\server2\vol1\public\zenworks\app1  
\\server3\vol1\public\zenworks\app1  
\\server4\vol1\public\zenworks\app1
```

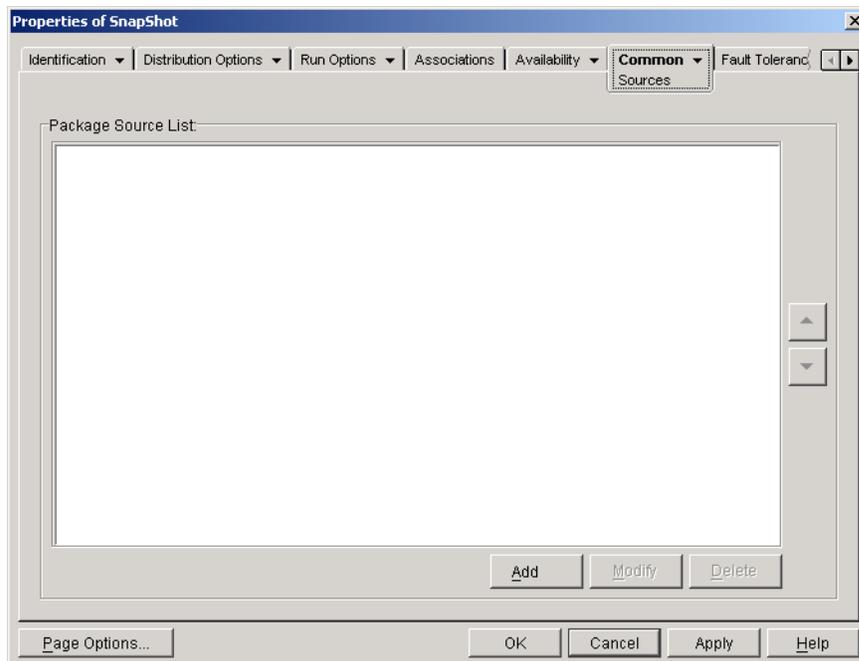
You can define each of the additional installation packages as sources for App1, or you can create three new Application objects using each of the installation packages and then have App1 link to the three new Application objects.

Defining each of the additional installation packages as sources for App1 has the advantage of requiring configuration of only one Application object. Regardless of which installation package is used, the App1 object's distribution settings are applied.

Defining additional Application objects using the installation packages enables you to have different Application object configuration settings for each installation package.

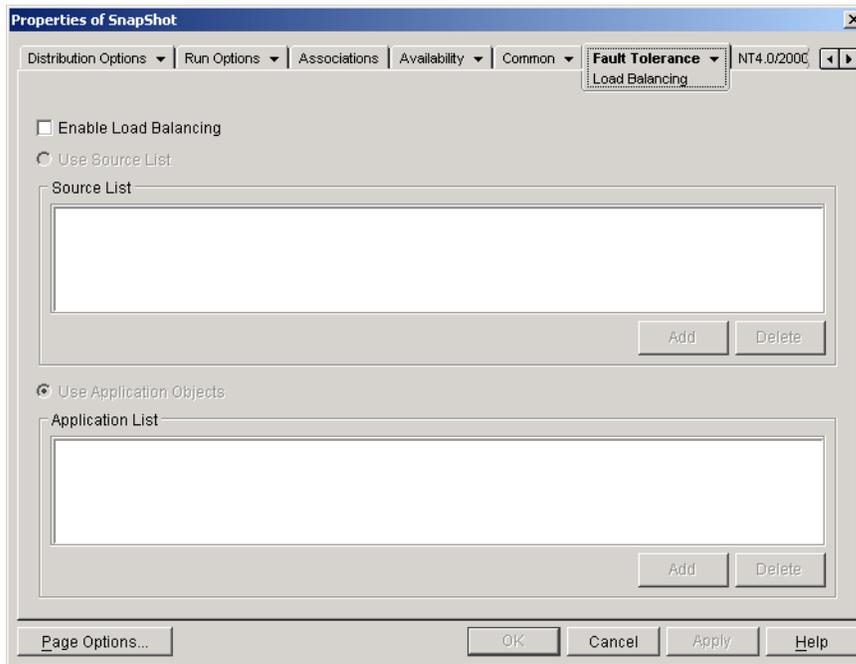
To set up load balancing for a simple application or an AOT/AXT application:

- 1** Copy the additional installation packages to the server locations you want.
- 2** (Conditional) If you plan to create Application objects for each of the additional installation packages so that you can link to the Application objects, create the Application objects. If necessary, refer to [Section 28.2, “Configuring the Application in eDirectory,” on page 326](#).
- 3** In ConsoleOne, right-click the Application object for which you want to establish load balancing, then click *Properties* to display the Application object's property pages.
- 4** (Conditional) If you are using installation packages only (no Application objects for them), define the installation packages as additional sources for the application. To do so:
 - 4a** Click the *Common* tab > *Sources* to display the Sources page.



- 4b** Click *Add*, specify the source path for an installation package, then click *OK* to add the additional source to the list.
- 4c** Repeat [Step 4b](#) until you've added all available sources.

5 Click the *Fault Tolerance* tab > *Load Balancing* to display the Load Balancing page.



6 Fill in the following fields:

Enable load balancing: Select this option to enable load balancing. The other load balancing options are available only if this option is enabled.

Use source list: Select this option to have Application Launcher use a list of sources for load balancing. You must have already created at least one installation package source (see [Step 4](#)). To add a source to the Source List, click *Add*, browse to and select the source, then click *OK*.

To remove a source, select the source from the Source List, click *Delete*.

Use application list: Select this option to have Application Launcher use Application objects for load balancing. You must have already created additional Application objects for the application, with the installation package being stored on a different server or volume than this application (see [Step 2](#)).

To add an Application object to the Application List, click *Add*, browse to and select the Application object, then click *OK*.

To remove an Application object from the list, select the Application object, then click *Delete*.

7 Click *OK* to save your changes.

36.3 Setting Up Site Lists

Site lists let you link an application at one site (Site1) to an application at another site (Site2). By linking applications whose source installation packages are located on servers at different sites, you can distribute the application from the server that is closest to the user.

For example, you have two different sites, Site1 and Site2. Employees at both sites use the same spreadsheet program. Site1 has an Application object (App1) that distributes the spreadsheet program from a Site1 server, and Site2 has its own Application object (App2) to distribute the program from a Site2 server. If you link App1 to App2, any Site1 employee who travels to Site2 and

launches App1 has App2 installed from the Site2 server. Likewise, any Site2 employee who travels to Site1 and launches App2 has App1 installed from the Site1 server.

The site list is a distribution mechanism that applies only to undistributed, non-cached applications. If an application is already distributed or cached on a user's workstation, that application is used regardless of the site list.

Novell Client vs. ZENworks Middle Tier Server

When a user logs in to Novell eDirectory™ through the ZENworks Middle Tier Server rather than the Novell Client™, site lists work slightly differently. Rather than the Novell Client being used to determine the user's location, the Middle Tier Server determines the location. This means that the application closest to the Middle Tier Server is used, which is not necessarily the application closest to the user. In the above example, the Site1 user would continue to have applications distributed from the Site1 server rather than from a Site2 server.

How to Build a Site List

You can link to only one other Application object. However, when you do so, the first Application object is automatically linked to any other Application objects that the second Application object is linked to. For example, you have three identical Application objects (App1, App2, and App3) at three different sites. You link App1 to App2, which creates the following site lists for each of the Application objects:

App1 List	App2 List	App3 List
App2	App1	(none)

Because you can link an Application object to only one other Application object, you now need to link App3 to either App1 or App2. Doing so creates the following site list for each of the Application objects:

App1 List	App2 List	App3 List
App2	App1	App1
App3	App3	App2

Synchronizing Application Object GUIDs

Each Application object has a global unique identifier, referred to as a GUID. When the application is distributed to a workstation, its GUID is added to the Windows registry. This is how Application Launcher knows that the application has been distributed to the workstation.

When using site lists, you need to make sure that all Application objects that belong to the same site list have the same GUID. Otherwise, Application Launcher cannot correctly distribute and uninstall the application. For example:

- Site1 and Site2 have the same application at each site, named App1 and App2. The two applications are part of the same site list, but App1 has GUID 1234 and App2 has GUID 5678. A Site1 user is associated to App1, but has not installed the application. The Site1 user travels to Site2 and attempts to install App1. Because App2 is the closest application, it is installed and GUID 5678 is added to the workstation's registry. The user returns to Site1 and launches App1.

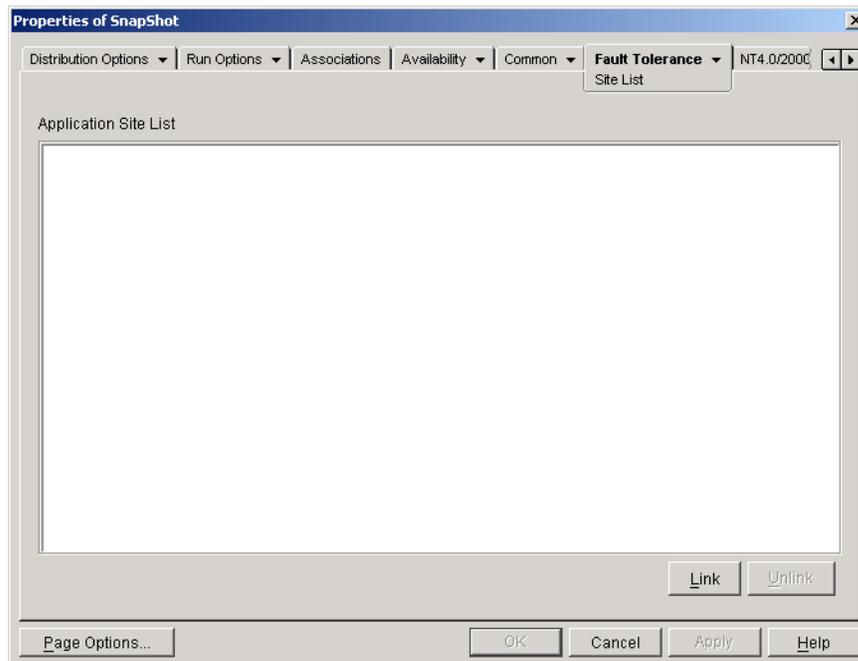
App1's GUID (1234) is not in the registry because App2's GUID is. Therefore, Application Launcher distributes App1 to the workstation even though it is identical to App2.

- ◆ The same situation occurs as listed above. However, this time the user tries to uninstall App1 after returning to Site1. Because App1's GUID is not in the registry, Application Launcher cannot uninstall the application.
- ◆ Site 1 and Site 2 have the same application chains. You must synchronize the GUIDs of each application in the chain at Site 1 with the GUIDs of each matching application at Site 2. For more information about chained applications, see [Section 37.2, “Application Chains,” on page 400](#).

The issues described above can be resolved by both applications in the site list having the same GUID. For information about synchronizing GUIDs, see [Section 50.4, “Manage Distribution GUIDs,” on page 610](#).

Establishing a Site List for an Application

- 1 In ConsoleOne, make sure that all applications to be included in the site list have the same GUID. For information about synchronizing GUIDs, see [Section 50.4, “Manage Distribution GUIDs,” on page 610](#).
- 2 Right-click the Application object for which you want to establish a site list, then click *Properties* to display the Application object's property pages.
- 3 Click the *Fault Tolerance* tab > *Site List* to display the Site List page.



- 4 Click *Link*, browse for and select the Application object you want to link to, then click *OK* to add it to the Application Site List.
If the Application object you added is also linked to other Application objects, those Application objects are also be added to the list
- 5 Repeat the previous step to link to additional Application objects.

6 Click *OK* when you are finished.

Integration with ZENworks Server Management

You can use ZENworks Server Management to distribute an application's source files to other servers and to replicate its Application object to other locations within your eDirectory tree. As part of this distribution process, you can have the site lists automatically generated. For more information, see “[Desktop Application Distribution](#)” in “[Policy and Distribution Services](#)” in the *Novell ZENworks 7 Server Management Administration Guide*.

Advanced Distribution: Configuring Application Dependencies and Chains

37

If you have an application that depends on other applications (or files), you can ensure that the other applications are available by adding them as dependencies.

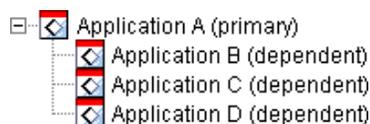
The following sections provide information and instructions to help you successfully create and manage dependencies:

- ◆ [Section 37.1, “Primary Applications vs. Dependent Applications,” on page 399](#)
- ◆ [Section 37.2, “Application Chains,” on page 400](#)
- ◆ [Section 37.3, “What Happens During the Distribution, Launch, and Uninstall of an Application That Has Dependencies,” on page 401](#)
- ◆ [Section 37.4, “Adding an Application Dependency,” on page 401](#)
- ◆ [Section 37.5, “Removing an Application Dependency,” on page 403](#)
- ◆ [Section 37.6, “Deleting an Application That Has Dependencies,” on page 404](#)
- ◆ [Section 37.7, “Creating an Application Chain,” on page 404](#)
- ◆ [Section 37.8, “Viewing an Application Chain,” on page 405](#)

37.1 Primary Applications vs. Dependent Applications

When working with application dependencies, the primary application is the application for which you are establishing dependencies. The applications that are defined as dependencies are called the dependent applications. The following illustration shows this relationship.

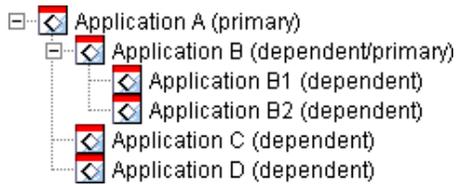
Figure 37-1 *Application A as the Primary Application with Applications B, C, and D as Dependent Applications.*



A primary application can have one dependent application or, as shown in the above example, it can have multiple dependent applications.

In addition, an application can be both a primary application and a dependent application, as shown in the following illustration.

Figure 37-2 Application A as the Primary Application with Applications B, B1, B2, C, and D as Dependent Applications.

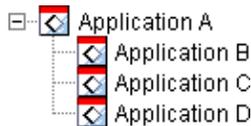


In the above example, Application B is one of Application A's dependent applications. At the same time, Application B has dependencies on two applications, Application B1 and Application B2.

37.2 Application Chains

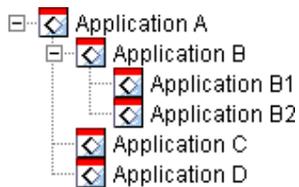
An application chain is two or more applications linked together by dependencies. In its simplest form, an application chain consists of two levels, as shown in the following illustration.

Figure 37-3 A Two-Level Application Chain



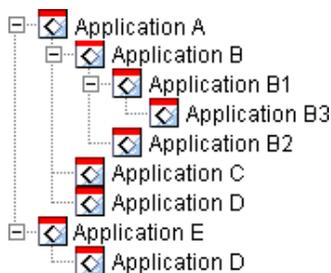
However, when application dependencies are nested, an application chain can grow to include many levels. In the following example, Application A has a dependency on Application B. Application B, in turn, has dependencies on Application B1 and Application B2. As a result, all three applications (B, B1, B2), in addition to applications C and D, must be present in order for Application A to run.

Figure 37-4 A Three-Level Application Chain



In some cases, an application might belong to more than one application chain, as in the following example where Application D is a dependent application for both Application A and Application E.

Figure 37-5 An Application that Belongs to Two Different Application Chains



37.3 What Happens During the Distribution, Launch, and Uninstall of an Application That Has Dependencies

When setting up application dependencies, you should be aware of the following:

- ♦ **Distribution:** When a user launches an application, if its dependent applications have not already been distributed to the user's workstation, Novell Application Launcher™ distributes them. Application Launcher distributes a dependent application only one time, unless the dependent application's version is updated (Application object > *Distribution Options* tab > *Options* page) or the distribution is unsuccessful. If Application Launcher is unable to distribute a dependent application (for example, the user's workstation does not meet the dependent application's system requirements), the primary application is not launched.
- ♦ **Distribution through removable media:** If you distribute an application through removable media such as a CD (see [“Distributing Applications Via Removable Media” on page 421](#)) or through an image add-on (see [“Imaging Page” on page 568](#)), its dependent applications must be included on the removable media or as another image add-on. Otherwise, the distribution of the primary application fails.
- ♦ **Distribution to disconnected workstations:** For workstations running in disconnected mode, the applications must be force cached to the workstation before it becomes disconnected (or they must be distributed through removable media). If changes are made to dependent applications after the applications have been cached to workstations, the version number of the primary application must be updated in order to force a re-cache of the applications.
- ♦ **Launch:** Each time an application is launched, Application Launcher performs any launch operations defined for its dependent applications. For example, if a dependent application has pre-launch and post-launch scripts (Application object > *Run Options* tab > *Launch Scripts* page), Application Launcher runs the pre-launch script, starts the dependent application (based on the executable file or application file listed in the Path to File field on the dependent application's Application page), and then runs the post-launch script.

If a dependent application, such as a setup program, should be run only one time, you should configure the dependent application to run once (Application object > *Run Options* tab > *Application* page).
- ♦ **Uninstall:** If you uninstall an application (Application object > *Common* tab > *Uninstall* page or Application object > *Associations* tab > *Associations* page), its dependent applications are uninstalled only if they are not used by another application.

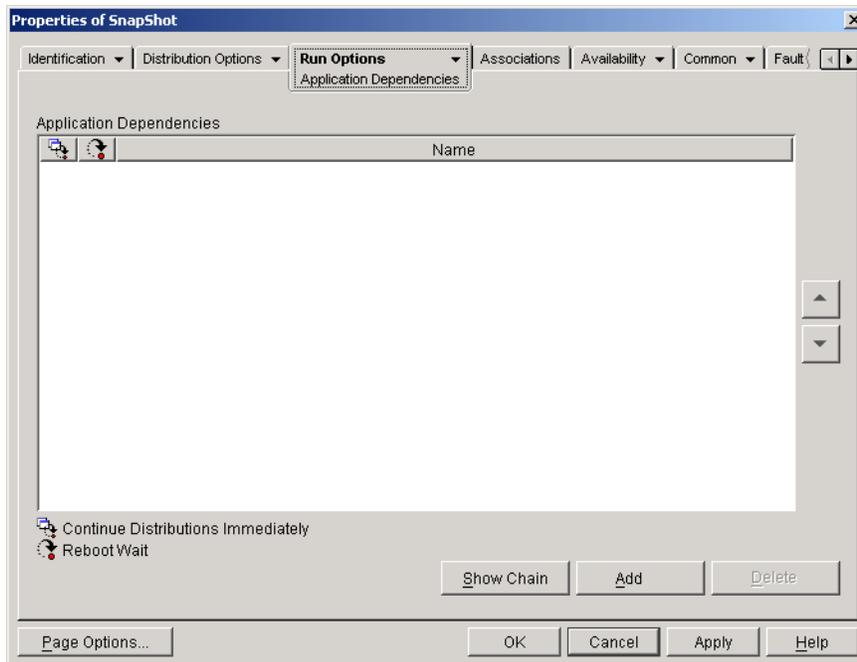
37.4 Adding an Application Dependency

Before adding a dependency to an application, consider the following:

- ♦ The dependent application must already exist as an Application object. If it does not, see [Section 28.2, “Configuring the Application in eDirectory,” on page 326](#).
- ♦ By default, each user who is associated with the primary application is added as a trustee of the dependent application. This provides the user with the eDirectory™ rights required to the dependent application. You should not associate users directly with the dependent application (dependent Application object > *Associations* tab) unless you want the dependent application to be displayed by Application Launcher.

To add a dependency to an application:

- 1 In ConsoleOne[®], right-click the Application object, then click *Properties* to display the Application object's property page.
- 2 Click the *Run Options* tab > *Application Dependencies* to display the Application Dependencies page.



- 3 Add the dependent applications to the list. To do so:
 - 3a Click *Add*, browse for and select the Application object, then click *OK*.
 - 3b If necessary, modify the following distribution attributes for the dependent application:

Continue distributions immediately: By default, Application Launcher waits until it distributes and executes the application before continuing with the action. Select this option if you want Application Launcher to continue with the next action regardless of whether or not the current application's distribution and execution is complete.

For example, if Regedit has a dependency on Notepad, the default action is for Application Launcher to distribute Notepad (if necessary), launch Notepad, and wait for the user to exit Notepad before it launches Regedit. The *Continue distributions immediately* option, however, would cause Application Launcher to distribute and launch Notepad and then, if the distribution and launch of Notepad was successful, continue immediately with Regedit. In other words, it does not wait for execution of the application to be completed.

Reboot wait: If distribution of the application requires the workstation to be rebooted, select this option to postpone the reboot until 1) Application Launcher distributes another application in the list that requires a reboot (and doesn't have this option selected) or 2) Application Launcher completes distribution of all applications in the list. This option also applies when uninstalling the application.
 - 3c Repeat **Step 3a** and **Step 3b** to add another application.
- 4 Arrange the dependent applications in the order in which you want them distributed and launched.

The order in which the dependent applications are listed is the order in which Application Launcher distributes and launches them, going from top to bottom. However, if an application has dependencies, the dependent applications must be distributed and launched before the primary application can be distributed and launched. You can use the up and down arrows to change the order.

5 Click *OK* to save your changes.

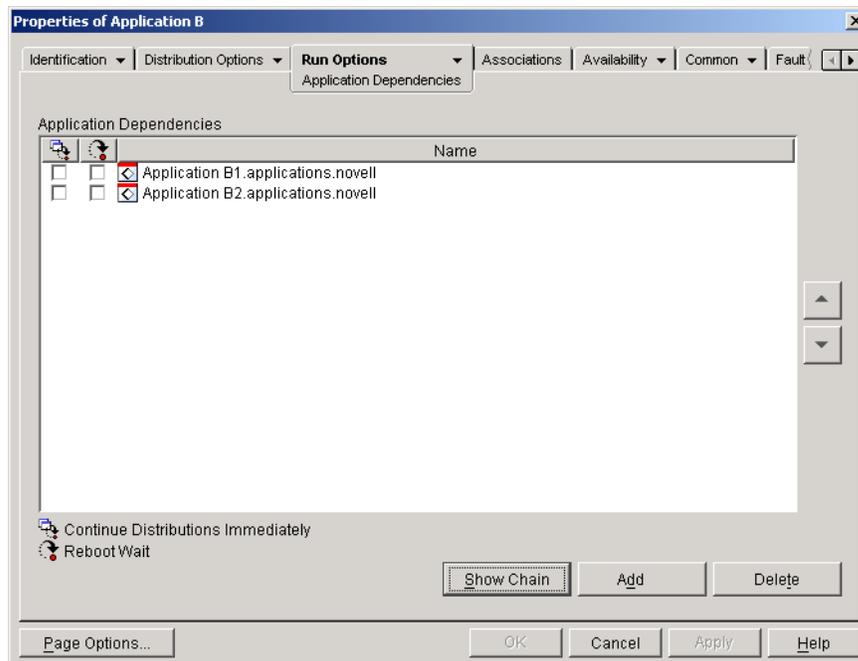
37.5 Removing an Application Dependency

If an application no longer has a dependency on another application, you can remove the dependency. Before doing so, you should be aware of the following:

- ♦ Removing the application dependency does not cause the dependent application's files to be uninstalled from the workstation (if the initial distribution included installing of the files). If you want the dependent application uninstalled from the workstation, you must uninstall the primary application, remove the dependency, then redistribute the primary application.
- ♦ The user must restart or refresh Application Launcher before the change is recognized. Until the restart or refresh occurs, the application dependency remains.

To remove an application dependency:

- 1 In ConsoleOne, right-click the Application object, then click *Properties* to display the Application object's property page.
- 2 Click the *Run Options* tab > *Application Dependencies* to display the Application Dependencies page.



- 3 In the Application Dependencies list, select the dependent application you want to remove, then click *Delete*.

37.6 Deleting an Application That Has Dependencies

Before you delete the Application object for an application that has dependencies, you should remove the dependencies. Doing so ensures the proper removal of user rights from the dependent applications' objects.

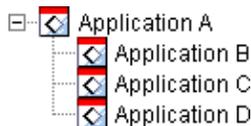
For example, if Application A has two dependent applications, Application B and Application C, users associated with Application A are automatically assigned trustee rights to the Application B object and Application C object. These rights are removed only if you delete Application B and Application C from Application A's dependency list before deleting Application A's object.

If you unknowingly delete an application before you remove its dependencies, you can manually remove users' trustee rights from the dependent application (ConsoleOne > dependent Application object > *NDS Rights* tab > *Trustees of This Object* page).

37.7 Creating an Application Chain

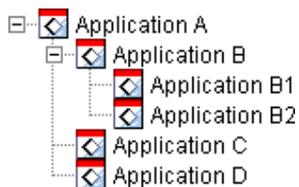
Application chains that consist of only two levels, as shown in the following illustration, are created by simply adding the dependent applications (B, C, and D) to the dependencies list for the primary application (A). For instructions, see [Section 37.4, “Adding an Application Dependency,” on page 401](#).

Figure 37-6 A Two-Level Application Chain



Application chains that consist of more than two levels, as shown in the following illustration, require you to configure dependencies at each level.

Figure 37-7 A Three-Level Application Chain



For example, to create the application chain shown in the above illustration, you would do the following:

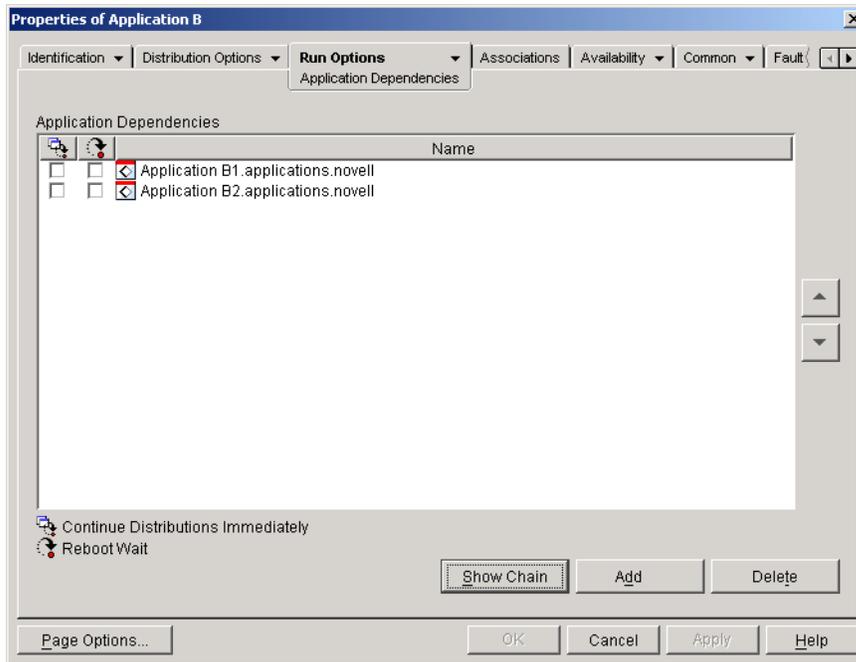
- 1 In the Application object for Application B, add Application B1 and B2 as dependent applications.
- 2 In the Application object for Application A, add Application B, Application C, and Application D as dependent applications.

For information about adding applications as dependent applications, see [Section 37.4, “Adding an Application Dependency,” on page 401](#).

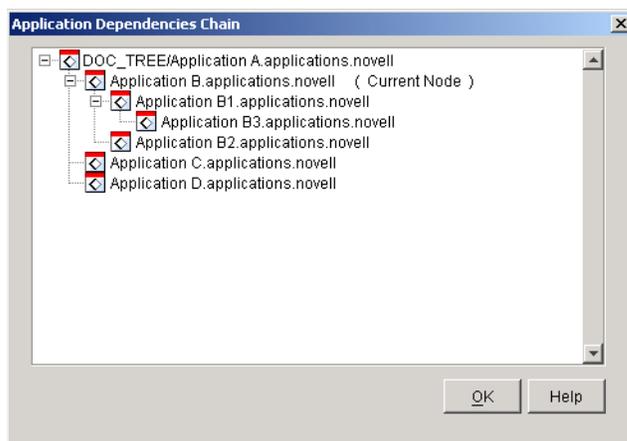
37.8 Viewing an Application Chain

You can view the chains in which the application is a member (either as the primary application or as a dependent application).

- 1 In ConsoleOne, right-click the Application object, then click *Properties* to display the Application object's property page.
- 2 Click the *Run Options* tab > *Application Dependencies* to display the Application Dependencies page.



- 3 Click *Show chain* to display the Application Dependencies Chain dialog box.



In the above example, Application B (the current application, or current node) is a dependent application for Application A. In addition, it has two dependent applications, Application B1 and Application B2.

- 4 When you are finished viewing the chain, click *OK*.

Users: Supporting Terminal Server Users

38

Novell® ZENworks® Desktop Management provides significant flexibility in the way that you can configure application distributions to terminal server users. The following sections provide information to help you establish your distribution method:

- ♦ Section 38.1, “Installing the RDP and ICA Clients,” on page 407
- ♦ Section 38.2, “Determining Where Novell Application Launcher Runs,” on page 407
- ♦ Section 38.3, “Managing Terminal Server User Accounts,” on page 408
- ♦ Section 38.4, “Determining the Best Type of Application Objects and File Packages to Use When Distributing an Application,” on page 409
- ♦ Section 38.5, “Ensuring Terminal Server Access Through a Firewall,” on page 410

38.1 Installing the RDP and ICA Clients

In order for Application Launcher to launch terminal server applications, users must have the appropriate RDP (Remote Desktop Protocol) client or ICA (Independent Computing Architecture) client installed on their workstations. Novell Application Launcher™ calls either the RDP client or ICA client based on the protocol defined in the terminal server's Application object. The RDP or ICA client then establishes the client session to the terminal server and launches the application.

You can use either the full RDP and ICA clients or the Web clients.

- ♦ If you plan to use the full clients, refer to your Citrix and Microsoft Terminal Services documentation for installation information.
- ♦ If you plan to use the Web clients, you can download them from the Citrix and Microsoft Web sites:

Citrix ICA `wficac.cab`: (<http://www.citrix.com>)

NOTE: Select the Minimal Web Client 7.00 option.

Microsoft RDP `msrdp.cab`: (<http://www.microsoft.com/windowsxp/pro/downloads/rdwebconn.asp>)

NOTE: Download the `tswebsetup.exe` file and extract the `msrdp.cab` file from it.

38.2 Determining Where Novell Application Launcher Runs

In a terminal server environment, you can run Application Launcher on each user's workstation or you can run it on each terminal server. If necessary, you can have some users run it on their workstations and other users run it on the terminal server.

Application Launcher on User Workstations

If Application Launcher is installed on the workstation, the user starts Application Launcher from the workstation just as he or she would in a non-terminal server environment.

The user must be able to authenticate to Novell eDirectory™ (unless he or she is running in disconnected mode), which means that the workstation must have the Novell Client™ installed or you must have a ZENworks Middle Tier Server installed.

Application Launcher displays the available terminal server applications the same way it does the available non-terminal server applications. However, when the user starts a terminal server application, Application Launcher opens a client session which in turn launches the application in the session. When the user exits the application, Application Launcher closes the client session.

Application Launcher on Terminal Servers

If Application Launcher is installed on the terminal server, the user opens a client session and then starts Application Launcher in the client session.

The user must be able to authenticate to eDirectory through the client session, which means that the workstation must have the Novell Client installed or you must have a ZENworks Middle Tier Server installed.

You need to be aware that Application Launcher does not create a NAL cache directory on the terminal server. This means that Application Launcher cannot run in disconnected mode, cannot uninstall applications it installed to the terminal server, and cannot use the random refresh startup feature. For more information about the NAL cache and the functionality it enables, see [Chapter 24, “Novell Application Launcher: Managing the Cache,” on page 303](#).

NOTE: After installing Windows 2000 Service Pack 4 (SP4), some programs might not work correctly. For example, the Novell Application Launcher window might fail to display on a Windows 2000 terminal server. To fix this problem, see [TID 10085889 \(http://support.novell.com/cgi-bin/search/searchtid.cgi?/10085889.htm\)](http://support.novell.com/cgi-bin/search/searchtid.cgi?/10085889.htm).

38.3 Managing Terminal Server User Accounts

To run an application from a terminal server, users must have a terminal server user account (either a local user account or a domain user account) and an eDirectory user account. The terminal server user account must provide sufficient file system access to run the applications being hosted on the terminal server. Typically, this requires the user to be a member of the Terminal Server User, Power User, or User group.

In addition, if users are running Application Launcher from the terminal server, the terminal server user account must provide file system access to the Application Launcher files (by default, installed in the `c:\program files\novell\zenworks` directory)

Multiple Users With the Same Terminal Server User Account

Application Launcher supports multiple users logging in to a terminal server through the same user account. However, if users log in through the same terminal server user account at the same time, you should be aware of the following issues:

- ◆ All users must also be logged in to eDirectory through the same eDirectory user account. Otherwise, Application Launcher only displays the applications and uses the configuration settings for the last user who starts or refreshes Application Launcher.
- ◆ All users need to be running Application Launcher. If one or more users is not, the application icons (distributed by Application Launcher) disappear from the desktop and Quick Launch bar of all users. The icons remain in the Application Launcher windows and the System tray. A workaround is to configure each Application object so that Application Launcher only displays the application icons in the Start menu, System Tray, and Application Launcher windows.
- ◆ Set the *Enable automatic icon cleanup* option to *No* (ConsoleOne > User object > ZENworks tab > *Launcher Configuration* page > *Add* button > *User* tab). This restricts Application Launcher from removing application icons when a user exits Application Launcher. For additional information, see [Section 21.3, “Configuring User Settings,” on page 268](#).

38.4 Determining the Best Type of Application Objects and File Packages to Use When Distributing an Application

The type of Application object and file package you create for distributing a terminal server application depends on several factors. The following list describes each type of Application object and package you can use and when you would want to use them:

- ◆ **Terminal server application:** You should use a terminal server Application object when the application is installed on the terminal server and Application Launcher is installed on users' workstations, not on the terminal server. When a user starts Application Launcher on his or her workstation and double-clicks a terminal server Application object, Application Launcher calls the RDP or ICA client on the workstation, which then opens a client session with the terminal server and launches the application in the session.

Do not use a terminal server Application object when users are running Application Launcher inside a client session. This is not supported, even if the application is hosted on a different terminal server than the one running Application Launcher. If you do so, the user receives the following error: “Unable to get attributes for Application object...”. The alternatives are to 1) configure the application as a simple Application object or 2) run Application Launcher on the user workstation, rather than the terminal server, and configure the application as a terminal server Application object.

For instructions about creating a terminal server Application object, see [Section 28.2, “Configuring the Application in eDirectory,” on page 326](#).

- ◆ **Simple application:** You should use a simple Application object when the application is already installed on the terminal server and Application Launcher is running on the terminal server (in other words, the user opens a client session to the terminal server and then runs Application Launcher in the client session). The simple Application object just points to the application executable file on the terminal server.

You could also use a simple Application object to have Application Launcher install an application to the terminal server when the user launches it. This should only be done with applications that require minimal files to be copied or configuration settings to be changed. Otherwise, an AOT/AXT Application object or an MSI Application object should be used.

For instructions about creating a simple Application object, see [Section 28.2, “Configuring the Application in eDirectory,” on page 326.](#)

- ◆ **.AOT/.AXT application:** You should use an AOT/AXT Application object when the user is running Application Launcher from the terminal server, you want the application distributed to the terminal server when the user launches it, and the application is too complex to be distributed as a simple Application object.

The user must have sufficient file system and registry rights to copy files and registry settings to the locations defined in the Application object. If multiple users install the application, only user-specific files and registry settings are distributed after the first distribution.

For instructions about creating an AOT/AXT Application object and file package, see [Section 28.2, “Configuring the Application in eDirectory,” on page 326.](#)

- ◆ **.MSI application:** You should use an MSI Application object when the user is running Application Launcher from the terminal server, the application is a Microsoft Windows Installer (MSI) application, and you want it distributed to the terminal server when the user launches it.

The user must be a member of the Administrator group. Microsoft Windows Installer does not allow non-administrator users to do installations through a terminal server client session.

For instructions about creating an MSI Application object and file package, see [Section 28.2, “Configuring the Application in eDirectory,” on page 326.](#)

38.5 Ensuring Terminal Server Access Through a Firewall

To ensure that users can access terminal servers through a firewall, you need to open the following firewall ports for sending and receiving TCP/IP packets:

- ◆ **3389:** The RDP client and Windows Terminal Services send and receive packets via this port. This is the standard RDP port. If you’ve used a non-standard RDP port, open that port.
- ◆ **1494 (in only):** The ICA client sends packets to Citrix MetaFrame servers through this port. This is the standard ICA port. If you’ve used a non-standard ICA port, open that port. For more information about Citrix firewall requirements, see the Citrix documentation.
- ◆ **1023 and above (out only):** Citrix MetaFrame servers send packets to ICA clients through these ports. For more information about Citrix firewall requirements, see the Citrix documentation.

Users: Supporting Disconnected Users

39

Novell® Application Launcher™ enables users to distribute, launch, verify, and uninstall applications while disconnected from Novell eDirectory™. This enables users to run the same applications when connected to or disconnected from eDirectory.

The following sections provide information and instructions to help you understand the Application Launcher disconnected mode and to support disconnected users:

- ♦ [Section 39.1, “Disconnected Mode Overview,” on page 411](#)
- ♦ [Section 39.2, “Configuring Applications as Disconnectable,” on page 413](#)
- ♦ [Section 39.3, “Configuring Application Launcher to Start Automatically,” on page 414](#)
- ♦ [Section 39.4, “Distributing Applications to Disconnected Workstations,” on page 414](#)

39.1 Disconnected Mode Overview

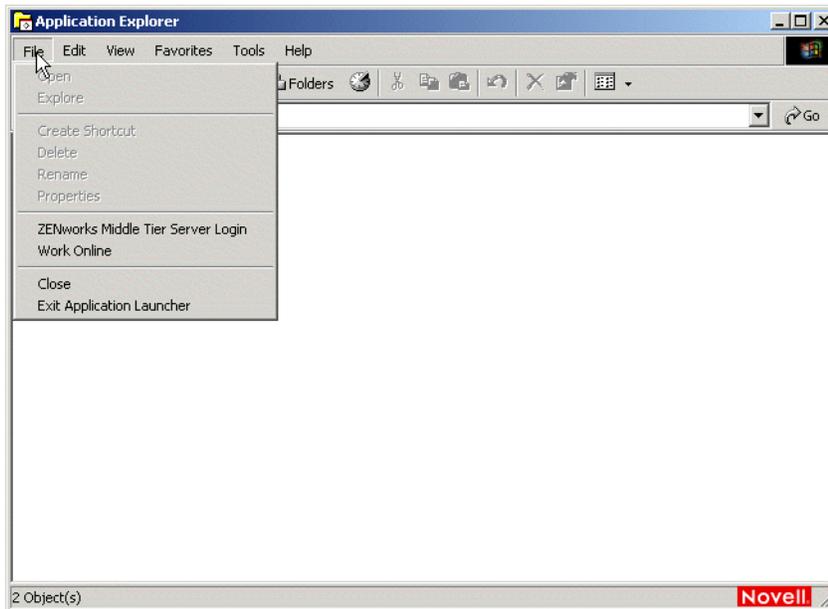
Application Launcher runs in disconnected mode when neither the user nor the workstation is authenticated to eDirectory. User authentication occurs whenever the user logs into eDirectory through the Novell Client™ or the ZENworks® Middle Tier Server. Workstation authentication occurs only if 1) the workstation has been imported as a Workstation eDirectory object and 2) Workstation Manager is installed and can perform the authentication.

Disconnected Mode Indicators

There are several ways you can tell when Application Launcher is running in disconnected mode:

- ♦ **File menu:** The File menu in the Application Window, Application Explorer window, and Application Browser window includes a *Work offline/work online* option. When Application Launcher is in disconnected mode, *Work online* is displayed.

Figure 39-1 Application Explorer Window Showing the Work Online Option Available on the File Menu



- ◆ **Application Explorer:** The Application Explorer desktop icon changes. The icon on the left below indicates connected mode. The icon on the right indicates disconnected mode.



In addition, the Application Explorer system tray icon changes. The icon on the left below indicates connected mode. The icon on the right indicates disconnected mode.



The NAL Cache

In disconnected mode, Application Launcher reads application information from the NAL cache on the workstation's local drive. Any applications that have been distributed or cached to the workstation continue to be displayed on the workstation. The NAL cache is a critical component of disconnected mode. If you have not already done so, you should review the information in [Chapter 24, "Novell Application Launcher: Managing the Cache," on page 303](#).

Work Offline

Application Launcher includes a Work Offline option that lets users force Application Launcher into disconnected mode while remaining authenticated to eDirectory. Work Offline causes Application Launcher to start reading the workstation's NAL cache directory for application information rather than eDirectory. The corresponding Work Online option enables users to switch back to connected mode.

Disconnected Applications

When you create an Application object, the application is automatically configured as being disconnectable. After a disconnectable application is distributed or force cached to a workstation,

Application Launcher continues to display the Application object icon even after the user has disconnected from eDirectory. When the user double-clicks the Application object icon, Application Launcher launches the application.

If an application is not marked disconnectable, Application Launcher does not display the Application object icon when the user disconnects from eDirectory.

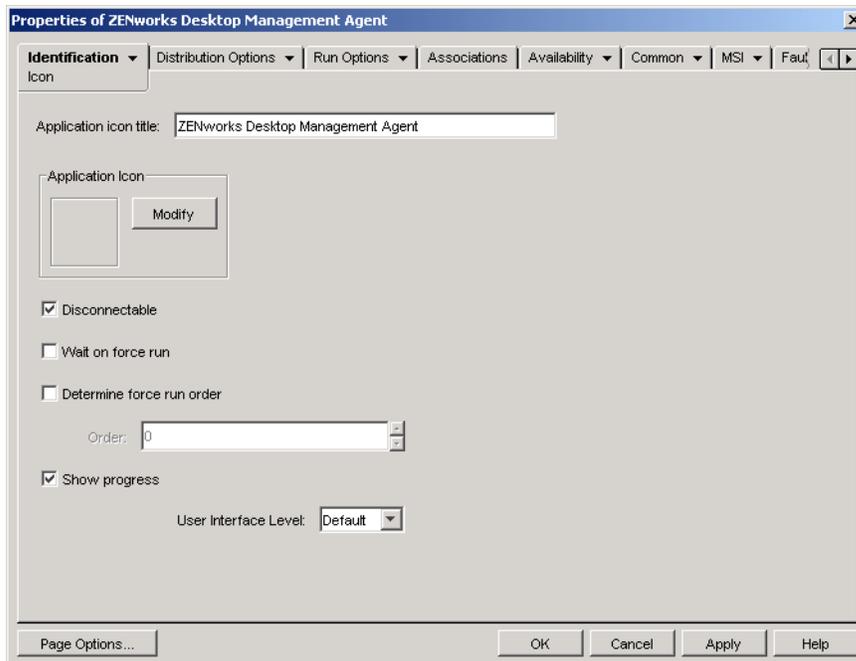
Only applications that can run on a disconnected workstation should be configured as disconnectable. Applications that rely on network resources should not be configured as disconnectable if the network resources is not available. This includes:

- ◆ Applications that access a network database
- ◆ Client/server applications
- ◆ Applications that depend on network drive mappings or print captures
- ◆ Applications that use eDirectory macros (other than %CN%) in any of the Application object property fields
- ◆ Applications that require a persistent connection to the network

39.2 Configuring Applications as Disconnectable

By default, when you create an Application object the application is marked as being disconnectable. To verify this setting or mark the application as not being disconnectable:

- 1 In ConsoleOne[®], right-click the Application object, then click *Properties* to display the Icon page.



- 2 Select the *Disconnectable* option to make the application disconnectable.

or

Deselect the option to make the application not disconnectable.

3 Click *OK* to save the changes.

39.3 Configuring Application Launcher to Start Automatically

To ensure that Application Launcher starts automatically when a user is in disconnected mode, you can add Application Launcher to the Windows Startup folder during installation of the ZENworks Desktop Management Agent. For details, see “[Installing and Configuring the Desktop Management Agent](#)” in the *Novell ZENworks 7 Desktop Management Installation Guide*.

39.4 Distributing Applications to Disconnected Workstations

As long as a disconnected workstation has Application Launcher installed and running, you can distribute applications to the workstation through the use of removable media such as a CD, Jaz disk, or Zip disk. The removable media functions as a second workstation cache, containing the Application object settings and application source files required to install and run the application.

For example, you have a mobile user who seldom connects to the network but needs an application that you've distributed to other users. You burn a CD with the application and send it to the user. The user inserts the CD into his or her workstation's drive, Application Launcher reads the CD and displays the Application object's icon in the places you've configured (Application Launcher window, Start menu, desktop, and so forth). The user then launches the application, which is distributed to the workstation according to the Application object's configuration.

For information about how to create removable media containing applications, see [Section 50.6, “Create Virtual CD,”](#) on page 612.

Novell® ZENworks® Desktop Management includes several features to support users who connect to Novell eDirectory™ through a remote (slow) connection. The following sections provide information and instructions to help you use these features:

- ♦ [Section 40.1, “Configuring How Application Launcher Detects a Remote Connection,” on page 415](#)
- ♦ [Section 40.2, “Disabling Applications,” on page 417](#)
- ♦ [Section 40.3, “Establishing Alternate Applications,” on page 418](#)
- ♦ [Section 40.4, “Distributing Applications,” on page 420](#)

40.1 Configuring How Application Launcher Detects a Remote Connection

When Application Launcher starts, it determines whether it is running in local (fast connection) mode or remote (slow connection) mode. If Application Launcher detects an active dial-up connection, it starts in remote mode. In the case where an active dial-up connection is not detected, Application Launcher can use several different methods to determine whether it is running in local or remote mode. You determine which method Application Launcher uses by configuring the Application Launcher settings in ConsoleOne®.

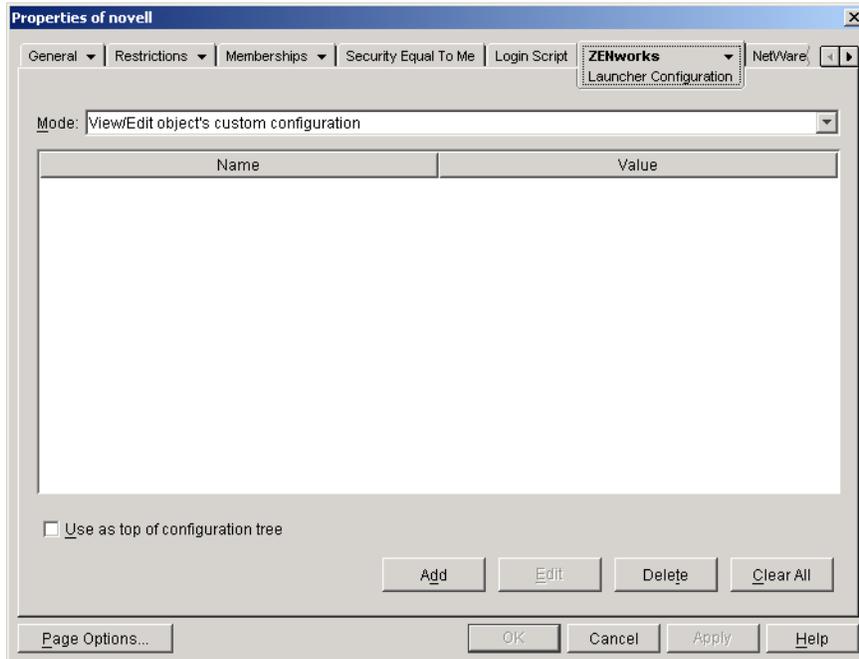
- 1 In ConsoleOne, select a container object if you want to configure the Application Launcher settings for all users in the container.

or

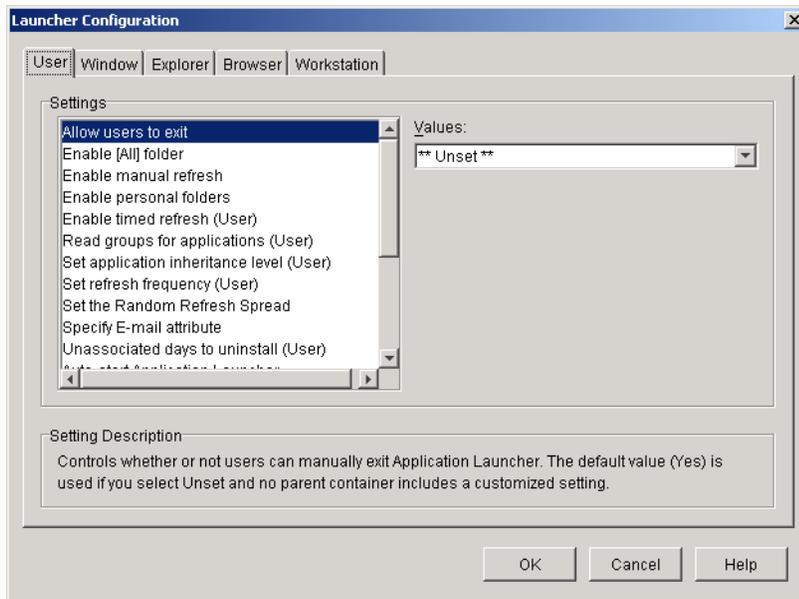
Select a User object if you want to configure an individual user's setting.

- 2 Right-click the object, then click *Properties* to display the object's property pages.

- 3 Click the *ZENworks* tab, then click *Launcher Configuration* to display the Launcher Configuration page.



- 4 Click *Add* to display the Launcher Configuration Settings page.



- 5 In the *Settings* list on the *Users* tab, select the *Configure remote access detection method* option.
- 6 In the *Values* field, select one of the following settings:
 - User will always be local:** Application Launcher functions as if the user is local.
 - User will always be remote:** Application Launcher functions as if the user is remote.

Prompt: Application Launcher prompts the user to select local or remote.

Auto-detect using max interface speed: Application Launcher detects the maximum speed of the network interface card and determines from that speed whether or not the user is local or remote. If you select this option, you must establish the connection speed (threshold) that determines local or remote status.

Detect using network ID: Application Launcher uses the workstation's network ID (also known as the network address) to establish whether or not the user is local or remote.

If you select this option, you must specify the network ID used to establish whether the user is local or remote. To determine the network ID, take the bit-wise logical AND comparison of the 32-bit IP address and 32-bit subnet mask, then convert the resulting 32-bit network ID to dotted decimal notation. In an AND comparison, the result of the two bits being compared is True (1) only when both bits are 1; otherwise, the result is False (0). For example:

10000001 00111000 10111101 00101001	(129.56.189.41 IP address)
11111111 11111111 11110000 00000000	(255.255.240.0 subnet mask)
10000001 00111000 10110000 00000000	(129.56.176.0 network ID)

If you want workstations whose network IDs match the specified network ID to be considered local, select Network ID Is Equal to This Network ID.

If you want workstations whose network IDs do not match the specified network ID to be considered local, select Network ID Is Not Equal to This Network ID.

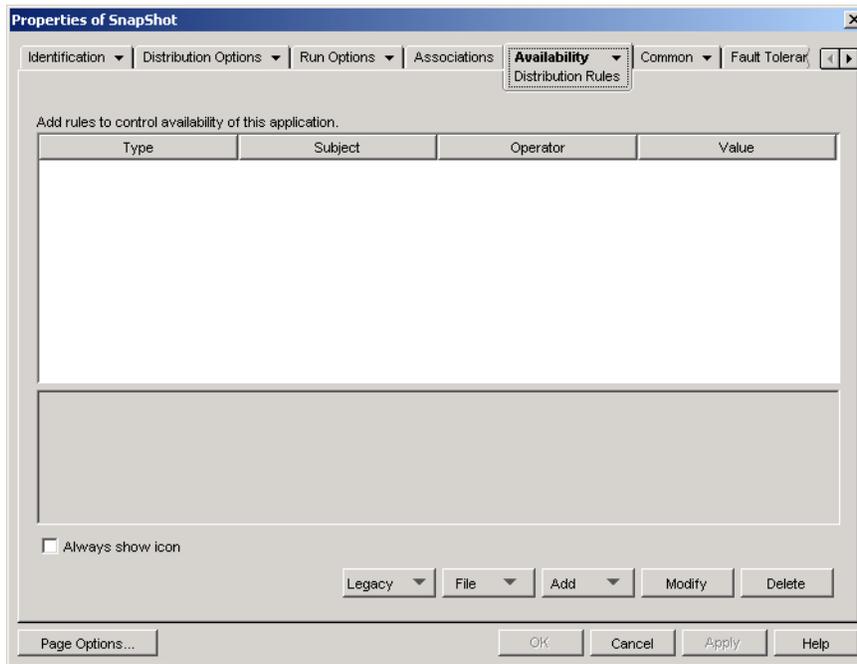
7 Click *OK* to save your changes.

40.2 Disabling Applications

By default, Application Launcher displays an application icon regardless of whether it is running in local or remote mode. If you don't want Application Launcher to display an application in remote mode, you can use the Application object's distribution rules to disable the application.

1 In ConsoleOne, right-click the Application object, then click *Properties* to display the Application object's property pages.

- 2 Click the *Availability* tab > *Distribution Rules* to display the Distribution Rules page.



- 3 Click *Add* > *Remote Access* to display the Remote Access Requirements dialog box.



- 4 Select the *Remote access connection* option if you want the application icon enabled only when Application Launcher is in remote mode.

or

Select the *LAN connection* option if you want the application icon enabled only when Application Launcher is in local (LAN) mode.

- 5 Click *OK* to add the rule to the list of distribution rules.
- 6 Click *OK* to save your changes and close the Application object's property pages.

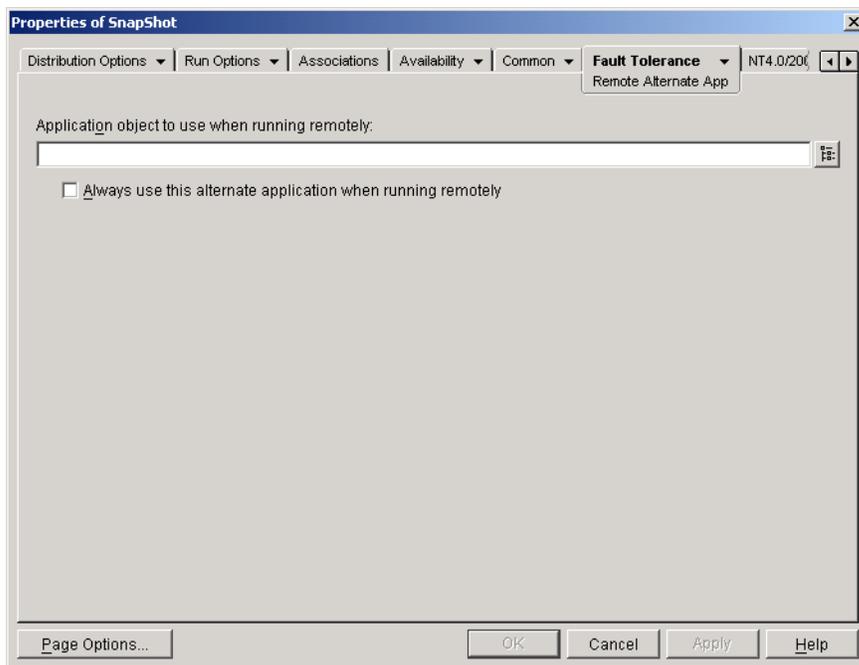
40.3 Establishing Alternate Applications

In some cases, rather than disabling an application altogether (see the previous section, [Disabling Applications](#)), you might want one instance of an application run when users are in local mode and a different instance run when they are in remote mode. To accomplish this, you can configure a “local mode” application to point to an alternate “remote mode” application. When a user launches the “local mode” application while in remote mode, Application Launcher launches the alternate “remote mode” application instead. Generally, this alternate application should be a terminal server application or Web application that is designed for use over slower remote connections.

For example, when a user is accessing an application while in local mode, you might want the application run from a network server installation. However, when the user is accessing the application while in remote mode, you might want it run from a terminal server. You would accomplish this by creating two Application objects, one for the network application and one for the terminal server application, and designating the terminal server application as the remote alternate application for the network server application.

To establish a remote alternate application for an application:

- 1 Make sure the remote alternate application is configured as an Application object in eDirectory.
- 2 In ConsoleOne, right-click the Application object for which you want to establish the remote alternate application, then click *Properties* to display the Application object's property pages.
- 3 Click the *Fault Tolerance* tab > *Remote Alternate App* to display the Remote Alternate App page.



- 4 Fill in the following fields:

Application object to use when running remotely: Select the Application object for the alternate application. Generally, this alternate application should be a terminal server application or Web application that is designed for use over slower remote connections.

Always use this alternate application when running remotely: By default, Application Launcher does not use the alternate application if the original application is installed on the user's workstation; instead, it uses the locally installed application. However, if the application requires access to a database or some other network resource that is only available if the alternate application is used, select this option to force Application Launcher to use the alternate application rather than the locally installed application.

- 5 Click *OK* to save your changes.

40.4 Distributing Applications

When a remote user attempts to run an application for the first time, Application Launcher distributes the application to the user over the slow connection. If this requires many files to be copied to the workstation, this distribution process could take a while.

To speed up the distribution process, you can cache the application to the user's workstation while he or she is still local or distribute the application to the user via removable media such as a CD, Jaz disk, or Zip disk. To let the user control the remote distribution process, you can enable checkpoint restart for the download files. The following sections provide instructions:

- ♦ [Section 40.4.1, “Adding Applications to the Cache,”](#) on page 420
- ♦ [Section 40.4.2, “Distributing Applications Via Removable Media,”](#) on page 421
- ♦ [Section 40.4.3, “Enabling Checkpoint Restart for File Downloads,”](#) on page 421

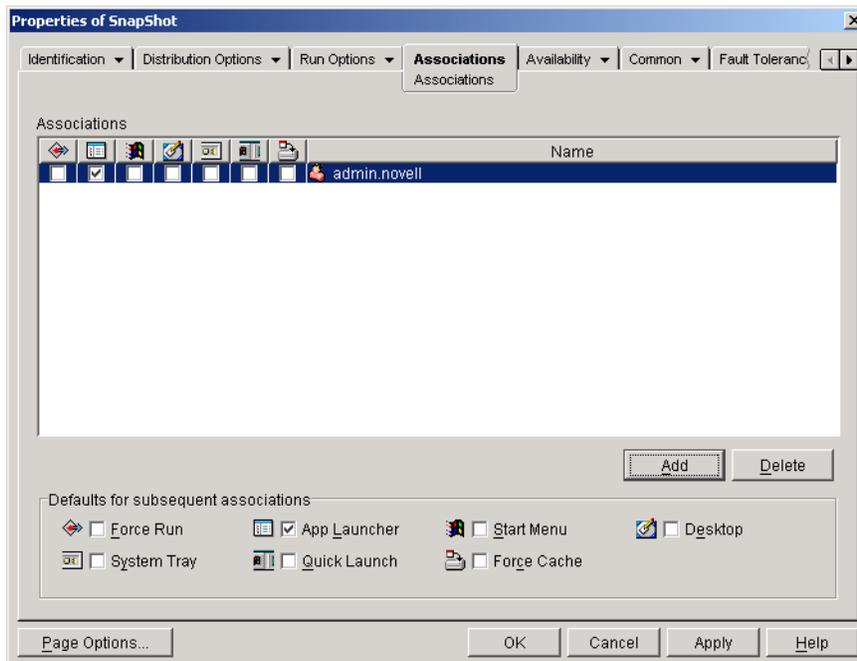
40.4.1 Adding Applications to the Cache

You can enable a user to install or verify an application from the local NAL cache rather than across a slow connection. To do so, you must copy the application's source files to the workstation's local NAL cache directory while the user is still connected via a local (LAN) connection.

For information about the NAL cache, see [Chapter 24, “Novell Application Launcher: Managing the Cache,”](#) on page 303.

To cache an application:

- 1 In ConsoleOne, right-click the Application object, then click *Properties*.
- 2 Click the *Associations* tab to display the Associations page.



- 3 In the Associations list, select the *Force cache* box for the user whose workstation you want to cache the application to.

NOTE: In order to cache an application, it must be marked as *Disconnectable*. If the application is not marked as *Disconnectable*, the *Force cache* option is not available. For instructions, see [Section 39.2, “Configuring Applications as Disconnectable,” on page 413](#).

If you select the *Distribute always* option and a remote connection is detected, ZENworks behaves as if the *Force cache* and *Check point restart* options are enabled (even if you did not enable these options when you configured the Application object).

- 4 Click *OK* to save the changes.

The application is cached to the user's local drive the next time Application Launcher restarts or refreshes.

40.4.2 Distributing Applications Via Removable Media

As long as a workstation has Application Launcher installed and running, you can distribute applications to the workstation through the use of removable media such as a CD, Jaz disk, or Zip disk. The removable media function as a second workstation cache, containing the Application object settings and application source files required to install and run the application.

For example, you have a mobile user who seldom connects to the network but needs an application that you've distributed to other users. You burn a CD with the application and then send it to the user. The user inserts the CD into his or her workstation's drive, and Application Launcher reads the CD and displays the Application object's icon in the places you've configured (Application Launcher window, Start menu, desktop, and so forth). The user then launches the application, which is distributed to the workstation according to the Application object's configuration.

For information about how to create removable media containing applications, see [Section 50.6, “Create Virtual CD,” on page 612](#).

40.4.3 Enabling Checkpoint Restart for File Downloads

When a remote user attempts to run an application for the first time, Application Launcher distributes the application to the user over the slow connection (if the application is not cached to the user's local drive). If this requires many files to be copied to the workstation, this download process could take a while.

To allow a user to interrupt an application's download process and then resume it at a later time, picking up at the point the download left off, Application Launcher supports checkpoint restart. By default, this feature is enabled. If checkpoint restart is not enabled, Application Launcher starts over on the download process rather than resuming where it left off.

NOTE: If you select the *Distribute always* option and a remote connection is detected, ZENworks behaves as if the *Force cache* and *Check point restart* options are enabled (even if you did not enable these options when you configured the Application object).

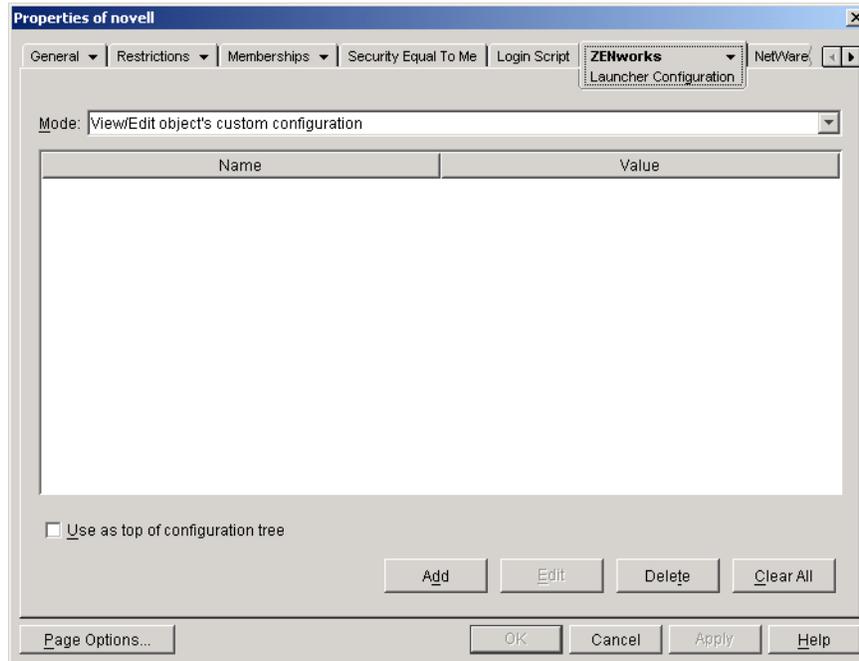
To verify that checkpoint restart is enabled, or to disable it:

- 1 In ConsoleOne, select a container object if you want to configure the checkpoint restart setting for all users in the container.

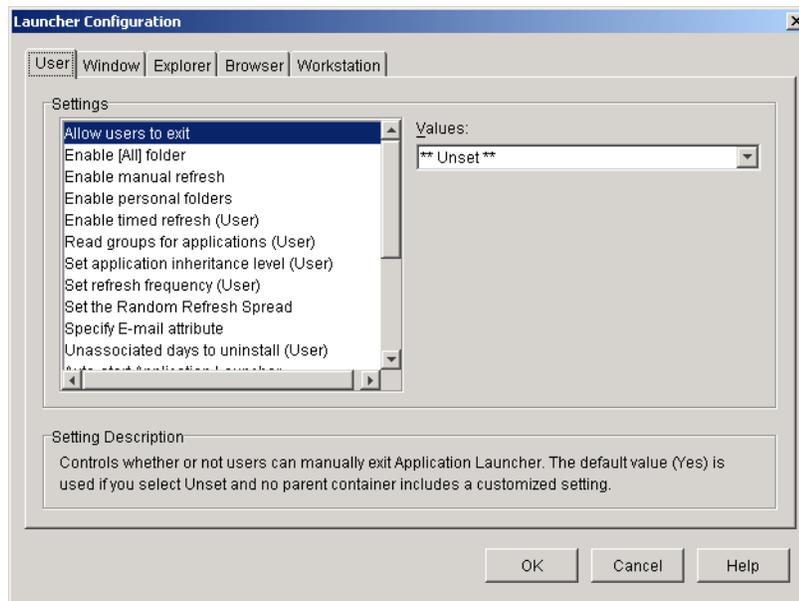
or

Select a User object if you want to configure an individual user's checkpoint restart setting.

- 2 Right-click the object, then click *Properties* to display the object's property pages.
- 3 Click the *ZENworks* tab > *Launcher Configuration* to display the Launcher Configuration page.



- 4 Click *Add* to display the Launcher Configuration Settings page.



- 5 In the *Settings* list on the *Users* tab, select the *Enable the checkpoint restart postpone button* option.

- 6** In the *Values* field, select *Yes* to enable it, *No* to disable it, or *Unset* to inherit the setting from the current object's parent container.
- 7** Click *OK* to save your changes.

Controlling Rogue Processes

41

You can configure Novell® Application Launcher™ to manage rogue processes. Rogue processes are processes that are not started by either Application Launcher or the LocalSystem user.

The following sections provide information to help you understand and implement rogue process management:

- ♦ [Section 41.1, “What Rogue Process Management Does,” on page 425](#)
- ♦ [Section 41.2, “How Rogue Process Management Works,” on page 425](#)
- ♦ [Section 41.3, “Setting Up Rogue Process Management,” on page 426](#)

41.1 What Rogue Process Management Does

To help you manage rogue processes, Application Launcher can do the following:

- ♦ Track and log the use of rogue processes.
- ♦ Ignore all rogue processes (allow them to run).
- ♦ Terminate all rogue processes.
- ♦ Apply an exceptions list. If rogue processes are being ignored, the processes included in the exceptions list are terminated. If rogue processes are being terminated, the processes in the exceptions list are ignored.

For example, if you only want to track rogue processes, you can configure Application Launcher to log rogue process information to a text file on a network server but still allow the processes to run. If, however, you want to shut down the use of rogue processes, you can configure Application Launcher to log the process information and also terminate the processes.

When you enable rogue process management, depending on the configuration setting you use, Application Launcher either ignores all rogue processes or terminates all rogue processes. However, if you want Application Launcher to either ignore or terminate all but a few rogue processes, you can create an exceptions list.

For example, if you want to allow all rogue processes except for the standard Windows games (Solitaire, Minesweep, Freecell, and Pinball), you could configure Application Launcher to ignore all rogue processes and create an exceptions list that includes the four Windows games. Application Launcher would then allow all rogue processes except the four games.

To ensure that user can't bypass the exceptions list by renaming the games' executable files, Application Launcher checks the launched processes' current executable name and the original filename (an internal filename) against the exceptions list.

41.2 How Rogue Process Management Works

The Windows operating system tracks all processes that are currently running. You can see this list by viewing the Processes tab in the Windows Task Manager (right-click the taskbar, then click *Task Manager > Processes*).

Each process has both a process identifier (PID) and a parent process identifier (parent PID). The parent PID identifies the process that launched it. Application Launcher uses a Windows API to retrieve the process list, including the PIDs and parent PIDs, every three seconds. Using the parent PIDs, Application Launcher knows whether or not the process is a rogue process. If the parent PID is not Application Launcher's PID, or if the process is not running as the LocalSystem user, then it is a rogue process.

After Application Launcher identifies the rogue processes, it performs the appropriate management actions, either ignoring or terminating the processes, taking into account any processes identified in the exceptions list. If logging is enabled, it also writes the rogue process information to the log file.

41.3 Setting Up Rogue Process Management

Rogue process management is enabled and configured through the Windows registry. The following sections explain how to manually modify the registry on Windows 98 and Windows 2000/XP workstations and how to create an Application object to distribute the registry changes through Application Launcher:

- ♦ [Section 41.3.1, “Manually Modifying the Registry,” on page 426](#)
- ♦ [Section 41.3.2, “Creating an Application Object to Distribute the Registry Modifications,” on page 428](#)

41.3.1 Manually Modifying the Registry

1 Use `regedit.exe` to open the Windows registry.

2 Locate the following key:

```
HKEY_CURRENT_USER\Software\NetWare\NAL\1.0
```

3 Add a Process Management key under the 1.0 key:

```
HKEY_CURRENT_USER\Software\NetWare\NAL\1.0\Process Management
```

4 Add a Default Action value to the Process Management key using the following information:

Value type: DWORD

Value name: Default Action

Value data: To have Application Launcher ignore all rogue processes, enter 0. To have Application Launcher terminate all rogue processes, enter 1.

NOTE: If you build an exception list, as in [Step 8](#) below, the meaning of this value data changes. See the information in that step for more information.

5 Add a Report Terminated value to the Process Management key using the following information:

Value type: DWORD

Value name: Report Terminated

Value data: To disable reporting of rogue processes that Application Launcher terminates, enter 0. To enable reporting of terminated rogue processes, enter 1.

6 Add a Report Ignored value to the Process Management key using the following information:

Value type: DWORD

Value name: Report Ignored

Value data: To disable reporting of rogue processes that Application Launcher ignores, enter 0. To enable reporting of ignored rogue processes, enter 1.

- 7** If you enabled reporting by adding a *Report terminated* or *Report ignored* value, you need to determine where you want the reports sent. To do so:

- 7a** Add a *Reporting targets* key to the *Process management* key:

```
HKEY_CURRENT_USER\Software\NetWare\NAL\1.0\Process  
Management\Reporting Targets
```

- 7b** To configure database reporting, add a *Database* value to the *Reporting targets* key using the following information:

Value type: DWORD

Value name: Database

Value data: To disable reporting to a database, enter 0. To enable reporting to a database, enter 1. When this option is enabled, Application Launcher writes to the database determined by the Service Location Policy package associated with the user. For more information, see [“Enabling the ZENworks Database Policy” on page 445](#).

For information about queries you can use to generate reports from the database, see [Section 44.6, “Generating Reports from a Database,” on page 461](#).

- 7c** To configure SNMP reporting, add an *SNMP* value to the *Reporting targets* key using the following information:

Value type: DWORD

Value name: SNMP

Value data: To disable SNMP reporting, enter 0. To enable SNMP reporting, enter 1. When this option is enabled, Application Launcher sends to the SNMP trap targets determined by the Service Location Policy package associated with the user. For more information, see [“Enabling the SNMP Trap Targets Policy” on page 448](#).

- 7d** To configure XML reporting, add an *XML* value to the *Reporting Targets* key using the following information:

Value type: DWORD

Value name: XML

Value data: To disable XML reporting, enter 0. To enable XML reporting, enter 1. When this option is enabled, Application Launcher sends to the XML targets determined by the Service Location Policy package associated with the user. For more information, see [“Enabling the SNMP Trap Targets Policy” on page 448](#).

If the XML reports are being processed into a database, see [Section 44.6, “Generating Reports from a Database,” on page 461](#) for information about queries you can use to generate reports from the database.

- 7e** To configure log file reporting, add a *File* value to the *Reporting targets* key using the following information:

Value type: String

Value name: File

Value data: Specify the full path and filename to be used for the log file. This can be a mapped drive or a UNC path to a local or network drive. For example, `\\server1\vol1\process\rogue.txt`. If you do not enter a value, log file reporting is disabled.

For information about understanding the information written to the log file, see [Section 44.7, “Understanding Log File Reports,” on page 464.](#)

8 (Optional) Use an exception list. To configure the list for Rogue Process Management:

8a Create an `Exception list` key under the following Process management key:

```
HKEY_CURRENT_USER\Software\NetWare\NAL\1.0\Process  
Management\Exception List
```

Treat the `Default Action` setting ([Step 4](#) above) as a flag to determine the behavior of Rogue Process Management when exceptions are granted:

Value Data	Behavior
0	Blacklists the applications in the exception list and assumes they are rogues unless they are launched from Novell Application Launcher. Any processes not in the exception list are allowed to run normally.
1	Whitelists the applications listed in the exception list and assumes they are approved. Any other processes not included in the exception list are not allowed to run, unless they are launched from Novell Application Launcher.

8b Add a `DWORD` value to the `Exception list` key for each process you want added to the list.

Value type: `DWORD`

Value name: Specify the process internal filename. If the process has an internal filename, it is listed on the *Version* tab of the executable's Properties dialog box (right-click the executable file > click *Properties* > click *Version*). For example, `sol.exe` has an internal filename of `sol.exe`, so you would name this application `sol.exe` on the exception list. If an application has no internal filename, use its standard filename or its name without the file extension. Do not provide a path for the file; specify only the filename.

Value data: Do not specify anything in this field.

9 Save the registry changes.

41.3.2 Creating an Application Object to Distribute the Registry Modifications

1 In ConsoleOne[®], create a simple Application object (see [Chapter 28, “Distribution: Simple Applications,” on page 325](#) if necessary), using the following information:

Object name: Specify a unique name for the eDirectory object (for example, Rogue Process Management).

Path to file: Do not specify anything in this field.

Distribution rules: If you want to enforce specific rules before the registry changes are distributed to a workstation, define the rules. You can also do this at a later time.

Associations: Select the users or workstations you want the changes distributed to. You can also do this at a later time.

2 After you've created the Application object, right-click the object, then click *Properties* to display the object's property pages.

3 Click *Distribution Options*, then click *Registry* to display the Registry page.

4 Create the following registry key:

```
HKEY_CURRENT_USER\Software\NetWare\NAL\1.0\Process Management
```

To create the key:

4a Select the HKEY_CURRENT_USER key, click *Add*, then click *Key* to add a new entry key called New Key.

4b Rename the key to Software.

4c Repeat the process described in **Step 4a** and **Step 4b** to create the complete key structure.

5 Add a Default Action value to the Process Management key. To do so:

5a Select the Process Management key, click *Add*, click *DWORD* to display the Edit DWORD Value dialog box, then fill in the fields as follows:

Value name: Default Action

Value data: To have Application Launcher ignore all rogue processes, enter 0. To have Application Launcher terminate all rogue processes, enter 1.

NOTE: If you build an exception list, as in **Step 9** below, the meaning of this value data changes. See the information in that step for more information.

5b Click *OK* to add the *Default Action* value to the Process management key.

6 Add a *Report Terminated* value to the Process management key. To do so:

6a Select the Process management key, click *Add*, click *DWORD* to display the Edit DWORD Value dialog box, then fill in the fields as follows:

Value name: Report Terminated

Value data: To disable reporting of rogue processes that Application Launcher terminates, enter 0. To enable reporting of terminated rogue processes, enter 1.

6b Click *OK* to add the Report Terminated value to the Process management key.

7 Add a *Report ignored* value to the Process management key. To do so:

7a Select the Process management key, click *Add*, click *DWORD* to display the Edit DWORD Value dialog box, then fill in the fields as follows:

Value name: Report Ignored

Value data: To disable reporting of rogue processes that Application Launcher ignores, enter 0. To enable reporting of ignored rogue processes, enter 1.

7b Click *OK* to add the *Report ignored* value to the Process management key.

8 If you enabled reporting by adding a *Report terminated* or *Report ignored* value, you need to determine where you want the reports sent. To do so:

8a Add a Reporting targets key to the Process management key:

```
HKEY_CURRENT_USER\Software\NetWare\NAL\1.0\Process Management\Reporting Targets
```

8b To configure database reporting, add a *Database* value to the Reporting targets key using the following information:

Value type: DWORD

Value name: Database

Value data: To disable reporting to a database, enter 0. To enable reporting to a database, enter 1. When this option is enabled, Application Launcher writes to the database determined by the Service Location Policy package associated with the user. For more information, see [“Enabling the ZENworks Database Policy” on page 445](#).

For information about queries you can use to generate reports from the database, see [Section 44.6, “Generating Reports from a Database,” on page 461](#).

- 8c** To configure SNMP reporting, add an *SNMP* value to the `Reporting targets` key using the following information:

Value type: DWORD

Value name: SNMP

Value data: To disable SNMP reporting, enter 0. To enable SNMP reporting, enter 1. When this option is enabled, Application Launcher sends to the SNMP trap targets determined by the Service Location Policy package associated with the user. For more information, see [“Enabling the SNMP Trap Targets Policy” on page 448](#).

- 8d** To configure XML reporting, add an *XML* value to the `Reporting targets` key using the following information:

Value type: DWORD

Value name: XML

Value data: To disable XML reporting, enter 0. To enable XML reporting, enter 1. When this option is enabled, Application Launcher sends to the XML targets determined by the Service Location Policy package associated with the user. For more information, see [“Enabling the SNMP Trap Targets Policy” on page 448](#).

If the XML reports are being processed into a database, see [Section 44.6, “Generating Reports from a Database,” on page 461](#) for information about queries you can use to generate reports from the database.

- 8e** To configure log file reporting, add a *File* value to the `Reporting targets` key using the following information:

Value type: String

Value name: File

Value data: Specify the full path and filename to be used for the log file. This can be a mapped drive or a UNC path to a local or network drive. For example, `\\server1\vol1\process\rogue.txt`. If you do not enter a value, log file reporting is disabled.

For information about understanding the information written to the log file, see [Section 44.7, “Understanding Log File Reports,” on page 464](#).

- 9** (Conditional) Use an exception list. To configure the list for Rogue Process Management:

- 9a** Create an `Exception list` key under the `Process management` key:

```
HKEY_CURRENT_USER\Software\NetWare\NAL\1.0\Process  
Management\Exception List
```

Treat the `Default Action` setting ([Step 5](#) above) as a flag to determine the behavior of Rogue Process Management when exceptions are granted:

Value Data	Behavior
0	Blacklist the applications in the exception list: assume all are rogues except those that were launched from Novell Application Launcher. Any other processes not included in the exception list are allowed to run normally.
1	Whitelist the applications listed in the exception list: assume all are approved, but terminate those that were not launched from Novell Application Launcher. Any other processes not included in the exception list are not allowed to run.

- 9b** Add a `DWORD` value to the `Exception list` key for each process you want added to the list.

Value type: `DWORD`

Value name: Enter the process internal filename. If the process has an “internal” filename, it is listed on the *Version* tab of the executable's Properties dialog box (right-click the executable file > click *Properties* > click *Version*). For example, `sol.exe` has an internal filename of `sol.exe`, so on the exception list you would name this application `sol.exe`. If an application has no internal filename, use its standard filename. Do not enter a path for the file, only the filename.

Value data: Do not enter anything in this field.

- 9c** Add a string value to the `Exception list` key for each process you want added to the list. To do so:

9c1 Select the `Exception list` key, click *Add.*, click `DWORD` to display the Edit `DWORD` Value dialog box, then fill in the fields as follows:

Value name: Enter the process filename. You can enter either the displayed executable name or the original filename. If the process has an original filename, it is listed on the *Version* tab of the executable's Properties dialog box (right-click the executable file, click *Properties*, then click *Version*). Do not enter a path for the file, only the filename.

Value data: Do not enter anything in this field.

9c2 Click *OK* to add the value to the `Exception list` key.

- 10** Click the *Associations* tab to display the Associations page.
- 11** Add the users and workstation you want to distribute the changes to. To make sure the changes are made without requiring any user intervention, enable the *Force run* option on each association.
- 12** Click *OK* to save the changes to the Application object.

The registry modifications are distributed to associated users the next time Application Launcher refreshes and to associated workstations the next time Application Launcher Workstation Helper refreshes.

Novell® ZENworks® Desktop Management includes the ability to verify and, if necessary, repair applications distributed through Novell Application Launcher™. By default, if Application Launcher is unable to launch an application, it prompts the user to verify the application.

The following sections provide information to help you understand how automatic application verification, sometimes referred to as application self-healing, works and how users can use Application Launcher to manually verify applications:

- ♦ [Section 42.1, “How Application Launcher Detects Broken Applications,” on page 433](#)
- ♦ [Section 42.2, “How Application Files are Verified,” on page 433](#)
- ♦ [Section 42.3, “Why Automatic Verification Can Fail,” on page 433](#)
- ♦ [Section 42.4, “Using Application Launcher to Verify an Application,” on page 434](#)

42.1 How Application Launcher Detects Broken Applications

Application Launcher detects a broken application at launch time. If the application file it calls can't launch (in other words, the CreateProcess call Application Launcher makes returns a FALSE), Application Launcher assumes the application is broken and needs to be verified. When this occurs, Application Launcher prompts the user that a problem exists and lets the user choose whether or not to verify the application.

42.2 How Application Files are Verified

If an application launch is not successful and the prompted user chooses to verify the application, Application Launcher redistributes all of the application's files to the workstation. How this is done depends on whether Application Launcher is running in connected or disconnected mode:

- ♦ In connected mode, the application files are redistributed from the network.
- ♦ In disconnected mode, the application files are redistributed from the local NAL cache directory (typically, `c:\nalcache`), provided the application files have been cached to the directory. If they have not been cached, the application cannot be verified until Application Launcher is in connected mode.

42.3 Why Automatic Verification Can Fail

Occasionally, Application Launcher cannot automatically repair a broken application, even when it has access to the application's installation files (from the network in connected mode or from the NAL cache in disconnected mode). This is because of the method Application Launcher uses to detect broken applications.

When Application Launcher successfully launches the file (in other words, the CreateProcess call it makes returns a true), it assumes a successful launch. In some cases, however, the file that Application Launcher calls doesn't actually start the application but in turn calls another file to start

the application. If that file is broken, missing, or cannot launch, the application launch fails without Application Launcher prompting the user to verify the application. For example:

- ♦ If you delete WinZip's `wz32.dll` and then launch `winzip32.exe`, Application Launcher successfully calls `winzip32.exe`, so it assumes a successful launch. However, when `winzip32.exe` calls `wz32.dll`, the launch fails because `wz32.dll` does not exist. Because Application Launcher only automatically prompts users to verify applications that it cannot launch, and it successfully launched `winzip32.exe`, the application is not automatically repaired.
- ♦ You delete `consoleone.exe` and then try to launch the application. Because Application Launcher calls Java, which in turn starts ConsoleOne[®], the launch fails. However, Application Launcher does not automatically verify ConsoleOne because it successfully launched Java.

In these cases, Application Launcher displays the following message:

```
Error message: Application Launcher Status - Could not launch
Application_Object_Name (id=xxx) The filename, directory name, or
volume label syntax is incorrect.
```

Although Application Launcher cannot automatically prompt users to verify applications that fit this scenario, users can manually initiate the verification on their own. For information about how to do this, see the next section, [Using Application Launcher to Verify an Application](#).

42.4 Using Application Launcher to Verify an Application

Users can use Application Launcher's Verify option to initiate the repair of a broken application. To do so, the user must:

- 1 Right-click the application icon.
- 2 Click *Verify*.

Uninstalling Applications

43

Applications distributed through Novell® ZENworks® Desktop Management can be uninstalled. You can choose whether or not to uninstall all files, shortcuts, INI entries, and registry entries that were installed or modified with the application.

Each user's workstation contains a local NAL cache directory that contains information about applications installed on the workstation. When you uninstall an application, this cache is used to ensure that the appropriate files and settings are removed from the workstation. If the application's source files have been copied to the workstation's local cache, the source files are also removed. For more information about the local cache, see [Chapter 24, “Novell Application Launcher: Managing the Cache,” on page 303](#).

By default, the ability to uninstall an application is disabled. You must first enable an application to be uninstalled. After that, you can uninstall an application by disassociating it from users or workstations, you can configure it to be uninstalled after a certain period of inactivity, or users can uninstall it through Novell Application Launcher™ (provided you've enabled this functionality). See the following sections for information:

- ◆ [Section 43.1, “Enabling an Application to be Uninstalled,” on page 435](#)
- ◆ [Section 43.2, “Uninstalling Applications by Disassociating Users or Workstations,” on page 437](#)
- ◆ [Section 43.3, “Uninstalling Unused Applications,” on page 438](#)
- ◆ [Section 43.4, “Using Application Launcher to Uninstall an Application,” on page 439](#)
- ◆ [Section 43.5, “Uninstalling Applications from Terminal Servers,” on page 440](#)

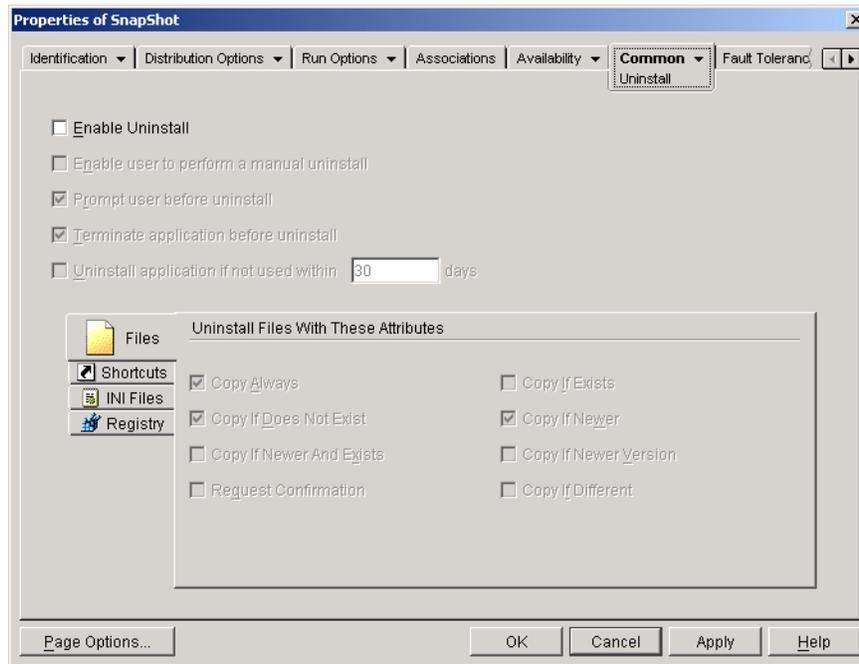
43.1 Enabling an Application to be Uninstalled

Uninstalling an application requires careful configuration of the Application object to ensure that no critical files are accidentally removed from the workstation. For this reason, the ability to uninstall an application is disabled by default.

To enable an application to be uninstalled:

- 1 In ConsoleOne®, right-click the Application object, then click *Properties*.

- 2 Click the *Common* tab > *Uninstall* to display the Uninstall page.



- 3 Click the *Enable uninstall* box to turn on the option.
- 4 If you want users to be able to uninstall the application themselves, click *Enable user to perform a manual uninstall* to turn on the option.

For information about how users can uninstall the application, see [Section 43.4, “Using Application Launcher to Uninstall an Application,”](#) on page 439.

- 5 Set the other Uninstall options as appropriate:

Prompt user before uninstall: Select this option to prompt users before removing the application from their workstations. If users answer *No* to the prompt, the application is not removed.

Prompt user before reboot: In some cases, uninstalling an application might require restarting the workstation. Select this option to prompt users before restarting the workstation. If users answer *No* to the prompt, the uninstallation is not completed until the next time they manually restart.

Terminate application before uninstall: Select this option to ensure that Application Launcher terminates the application (if the application is running) before it begins uninstalling the application files.

Uninstall application if not used within *xx* days: Select this option to automatically remove the application if the user has not run it within the specified number of days (the default is 30). For more information, see [Section 43.3, “Uninstalling Unused Applications,”](#) on page 438.

- 6 Select the attributes you want used when determining whether or not to uninstall files, shortcuts, INI settings, and registry settings.

Files: Click *Files*, then select the attributes you want to use to determine whether or not an application file is removed. The attributes correspond to the attributes listed on the Application Files page (*Distribution Options* tab). The default selections (*Copy always*, *Copy if does not*

exist, Copy if newer) ensure that a file is removed only if it was installed when the application was installed and not at an earlier time by another application that might still need the file.

Shortcuts: Click *Shortcuts*, then select the attributes you want to use to determine whether or not the application's shortcuts are removed. The attributes correspond to the attributes listed on the Shortcuts/Icons page (Distribution Options tab).

INI files: Click *INI files*, then select the attributes you want to use to determine whether or not an INI setting is removed. The attributes correspond to the attributes listed on the INI Settings page (Distribution Options tab). The default selections (*Create always, Create if exists, Create or Add to existing section*) ensure that a setting is removed only if it was created when the application was installed and not at an earlier time by another application that might still need the setting.

Registry: Click *Registry*, then select the attributes you want to use to determine whether or not a registry setting is removed. The attributes correspond to the attributes listed on the Registry Settings page (Distribution Options tab). The default selections (*Create always, Create if exists*) ensure that a setting is removed only if it was created when the application was installed and not at an earlier time by another application that might still need the setting.

The *Uninstall keys/values from these registry hives* option allows you to select two specific registry hives, `HKEY_LOCAL_MACHINE` and `HKEY_CURRENT_USER`, to include or exclude when removing the registry settings that were added by Application Launcher during distribution of the application to the workstation. The default settings cause Application Launcher to remove settings from the `HKEY_CURRENT_USER` hive but not from the `HKEY_LOCAL_MACHINE` hive.

The purpose of this option is to help you ensure that no Windows system settings are removed during the uninstall. This is particularly important if you used snAppShot™ to capture the application's installation. When snAppShot captures an application's installation, it includes all settings that are changed during the installation. If the application's installation program causes the workstation to reboot during the installation process, these changes can include not only settings that apply to the application but also settings that apply to the Windows system. Removing Windows system settings might cause the workstation to encounter problems when starting up. It is strongly recommended that you remove these settings only if you are positive that they will not affect the workstation's ability to function properly.

If you want to use this option, you should examine the registry settings that will be removed from the `HKEY_LOCAL_MACHINE` and `HKEY_CURRENT_USER` hives and delete any settings that apply to anything other than the application. The registry settings are displayed on the Registry Settings page (Distribution Options tab).

7 Click *OK* to save the changes.

43.2 Uninstalling Applications by Disassociating Users or Workstations

By default, when you disassociate an application from a user or workstation, Application Launcher no longer displays the Application object, but the application's files, shortcuts, INI settings, and registry settings are not removed from the workstation.

To configure an application to be uninstalled when it is disassociated from a user or workstation:

- 1 Enable the application to be uninstalled. To do so:
 - 1a In ConsoleOne, right-click the Application object, then click *Properties*.

- 1b** Click the *Common* tab > *Uninstall* to display the Uninstall page.
 - 1c** Select the *Enable uninstall* option to turn it on.
 - 1d** Configure any other uninstall settings as desired. See [Section 43.1, “Enabling an Application to be Uninstalled,”](#) on page 435 or refer to the Help on the Uninstall page.
 - 1e** Click *OK* to save your changes.
- 2** Configure Application Launcher to uninstall the application rather than leave it. To do so:
- You must configure Application Launcher for each user or workstation from which you want uninstallation to occur upon disassociation. You can do this through User, Workstation, Group, Workstation Group, or container objects.
- 2a** Right-click a User, Workstation, Group, Workstation Group, or container object, then click *Properties*.
 - 2b** Click the *ZENworks* tab > *Launcher Configuration* to display the Launcher Configuration page.
 - 2c** Click the *Add* button to display the Launcher Configuration dialog box.
 - 2d** Select the *User* tab, then the *Unassociated days to uninstall* setting.
 - 2e** In the *Values* field, select *Custom*, then specify the number of days after disassociation that you want applications uninstalled.
Specify 0 if you want uninstallation to occur immediately upon Application Launcher refresh or restart.
 - 2f** Repeat [Step 2d](#) and [Step 2e](#) to configure uninstallation of workstation-associated applications, using the Workstation tab rather than the User tab.
 - 2g** Click *OK*, then click *OK* again to save your changes.

The next time you disassociate a user or workstation (for which you've configured Application Launcher) from the application, the application is uninstalled from the workstation.

43.3 Uninstalling Unused Applications

Users might install some applications, run them a few times, and then not use them again for a long period of time. These applications can unnecessarily take up valuable workstation disk space.

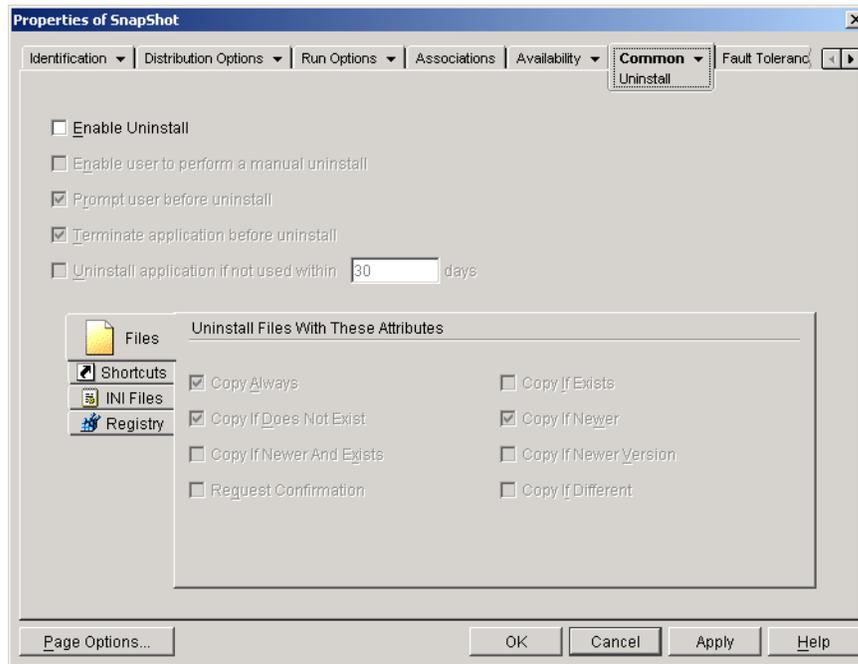
To free up disk space, you can have Application Launcher uninstall an application that has not been run for a period of time. The default time period is 30 days, but you can change the setting to meet your needs. For example, you might want one application removed after 15 days of inactivity, but you might not want to remove another application until it has not been run for 90 days.

After an unused application is uninstalled, Application Launcher continues to display the Application object icon on the workstation unless the user or workstation is disassociated from the Application object. If the user double-clicks the icon, the application is installed again.

To configure an unused application to be removed after a period of time:

- 1** In ConsoleOne, right-click the Application object, then click *Properties*.

2 Click the *Common* tab > *Uninstall* to display the Uninstall page.



3 If the *Enable uninstall* option is not turned on, select the check box to turn it on.

4 Click *Uninstall application if not used within xx days* to turn on the option, then select the number of days you want.

5 If you have not selected the attributes you want to use to determine whether or not the application's files, shortcuts, INI settings, and registry settings are removed, do so now. For information about each option, see [Section 43.1, “Enabling an Application to be Uninstalled,” on page 435](#) or refer to the Uninstall page's Help.

6 Click *OK* to save the changes.

43.4 Using Application Launcher to Uninstall an Application

By default, users are not allowed to uninstall distributed applications. However, if you've enabled users to uninstall applications (see [Section 43.1, “Enabling an Application to be Uninstalled,” on page 435](#)), they can do so through Application Launcher.

1 On the user's workstation, right-click the Application object, then click *Uninstall*.

When Application Launcher uninstalls an application, it removes all application files and settings from the workstation. However, if the Application object is still associated with the user, the Application object's icon remains on the workstation. This enables the user to reinstall the application at a later time.

43.5 Uninstalling Applications from Terminal Servers

When Application Launcher distributes an application to a workstation, it creates a NAL cache directory (typically, `c:\nalcache`) on the workstation. The NAL cache directory contains the information required to successfully uninstall the application.

However, when Application Launcher distributes an application to a terminal server, it does not create a NAL cache directory, which means that it does not have the information to uninstall the application from the terminal server. Therefore, you cannot use Application Launcher to uninstall an application that it has installed to a terminal server. If you no longer want the application on the terminal server, you must manually uninstall it.

Novell® Application Launcher™ can report on the success or failure of the following application events: launching, distributing, filtering, uninstalling, caching, and terminating.

Application Launcher can write events to the Sybase ODBC-compatible database or to a text log file, send them as SNMP traps to a management console, or send them as XML data to a Web URL for processing. Because event reporting is established on a per-application basis, you can use one reporting method for one application and a different reporting method for another. Or, you can use all four methods for the same application.

The following sections provide information about setting up each type of event reporting and configuring Application objects to use one or more reporting methods:

- ◆ [Section 44.1, “Setting Up Database Reporting,” on page 441](#)
- ◆ [Section 44.2, “Setting Up SNMP Trap Reporting,” on page 448](#)
- ◆ [Section 44.3, “Setting Up Log File Reporting,” on page 451](#)
- ◆ [Section 44.4, “Setting Up XML Reporting,” on page 452](#)
- ◆ [Section 44.5, “Configuring Applications to Use Reporting,” on page 458](#)
- ◆ [Section 44.6, “Generating Reports from a Database,” on page 461](#)
- ◆ [Section 44.7, “Understanding Log File Reports,” on page 464](#)
- ◆ [Section 44.8, “Enabling Windows Installer Verbose Logging,” on page 467](#)

NOTE: The reporting described in these sections applies only to applications launched by Application Launcher (in other words, applications that have Application objects). Application Launcher can also report on applications that it does not launch. These applications are referred to as rogue processes. For more information about setting up reporting for rogue processes, see [Chapter 41, “Controlling Rogue Processes,” on page 425](#).

For MSI applications, you can also turn on verbose logging so that the Windows Installer outputs information to a log file on the user's workstation. The following section provides information:

[Section 44.8, “Enabling Windows Installer Verbose Logging,” on page 467](#)

44.1 Setting Up Database Reporting

To set up reporting to the Sybase ODBC-compatible database, you need to complete the following tasks:

- ◆ [Section 44.1.1, “Installing the Sybase Database,” on page 442](#)
- ◆ [Section 44.1.2, “Installing ODBC Drivers to Workstations,” on page 442](#)
- ◆ [Section 44.1.3, “Creating a ZENworks Database Object,” on page 442](#)
- ◆ [Section 44.1.4, “Configuring the Database's ODBC Information,” on page 444](#)
- ◆ [Section 44.1.5, “Granting ODBC Property Rights to Users,” on page 445](#)
- ◆ [Section 44.1.6, “Enabling the ZENworks Database Policy,” on page 445](#)

- ◆ [Section 44.1.7, “Configuring Application Objects to Use Database Reporting,” on page 448](#)

44.1.1 Installing the Sybase Database

The Sybase database included on the *Novell ZENworks 7 Desktop Management* CD is the only supported database for direct ODBC reporting in ZENworks Application Management. For information about installing the Sybase database, see [“Installing the ZENworks Desktop Management Server”](#) in the *Novell ZENworks 7 Desktop Management Installation Guide*.

Sybase is also used for the Workstation Inventory database. If you plan to use a database for Application Management reports and you also plan to use Workstation Inventory, you can use the same database installation for both purposes. Each component creates its own database file. Application Management creates a `na1.db` database file and Workstation Inventory creates a `mgmt.db.db` database file.

Because the main requirement for Application Management reporting is that the database be at the same site as the users, you should follow the instructions provided for Workstation Inventory to deploy your databases, and then choose one or more databases to use for Application Management reporting. For information about database deployment for Workstation Inventory, see [Chapter 73, “Setting Up Workstation Inventory,” on page 901](#).

44.1.2 Installing ODBC Drivers to Workstations

Application Launcher uses an ODBC driver to write application event information to the database. You need to ensure that the correct ODBC driver for the Sybase database is installed on each workstation for which you want to record event information.

The correct ODBC driver for the Sybase database (`dbodbc7.dll`) is installed by default. However, on Windows 98, you might need to install two additional DLLs (`odbc32.dll` and `odbcint.dll`) that are not installed during a Windows minimal installation. Download the files from Microsoft's Web site or another file download site. To install these files, copy to the Windows System directory (for example, `c:\windows\system`). You can also use an Application object to distribute them to the appropriate workstation. You do not need to do this on Windows 2000/XP. The files are already present.

44.1.3 Creating a ZENworks Database Object

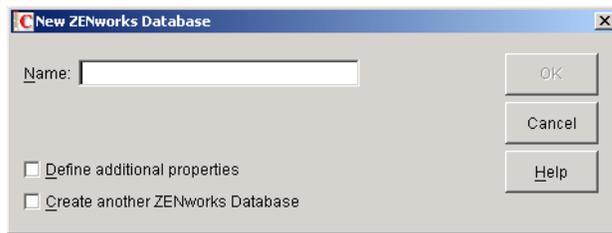
If you are using the Sybase database that comes with ZENworks[®] and have already installed it, the installation program creates a Database object (Desktop Management Database_*servername*) in eDirectory. You can skip to the next section, [“Configuring the Database's ODBC Information” on page 444](#).

If you are using another database, you must create a ZENworks Database object in eDirectory to represent the database.

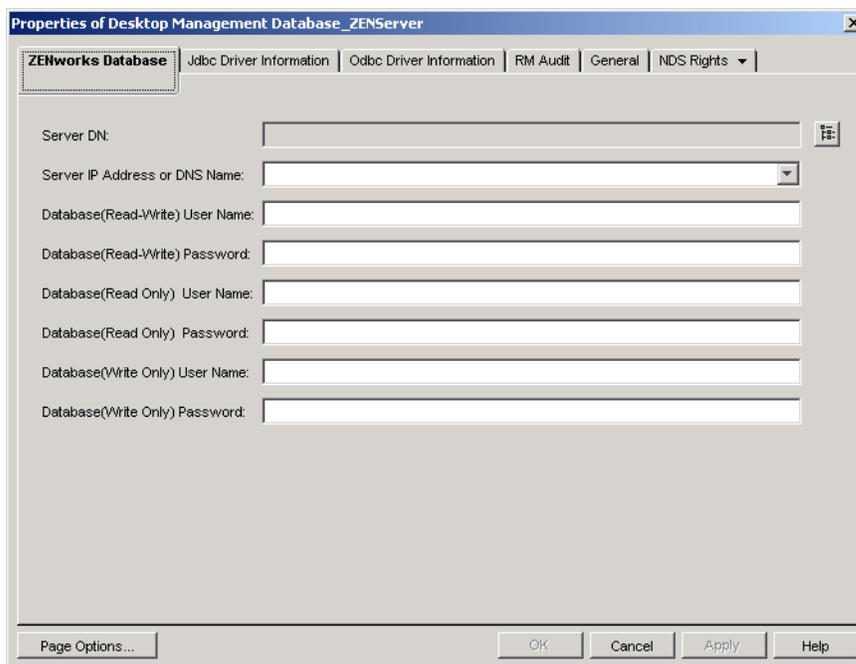
To create a ZENworks Database object:

- 1 In ConsoleOne[®], right-click the container where you want to create the object, click *New*, then click *Object* to display the New Object dialog box.

- 2 Select *ZENworks database*, then click *OK* to display the New ZENworks Database dialog box.



- 3 In the Name box, type a name for the database, such as Desktop Management Database_ZENServer.
- 4 Select the *Define additional properties* check box, then click *OK* to display the ZENworks Database property page.



- 5 In the *Server DN* field, browse for and select the Server object for the server where the database is physically installed and running.
- 6 Fill in the name and password fields for the Read-write, Read only, and Write only users:

Database (read-write) user name: Specify a username to provide both read and write access to the database. If you are using a Sybase database created by the Desktop Management Server installation program, the user name is MW_DBA.

Database (read-write) password: Specify a password for the Read-Write user. If you are using a Sybase database created by the Desktop Management Server installation program, the password is novell.

Database (read only) user name: Specify a username to provide only read access to the database. If you are using a Sybase database created by the Desktop Management Server installation program, the user name is MW_READER.

Database (read only) password: Specify a password for the Read Only user. If you are using a Sybase database created by the Desktop Management Server installation program, the password is novell.

Database (write only) user name: Specify a username to provide only write access to the database. If you are using a Sybase database created by the Desktop Management Server installation program, the user name is MW_UPDATER.

Database (write only) password: Specify a password for the Write Only user. If you are using a Sybase database created by the Desktop Management Server installation program, the password is novell.

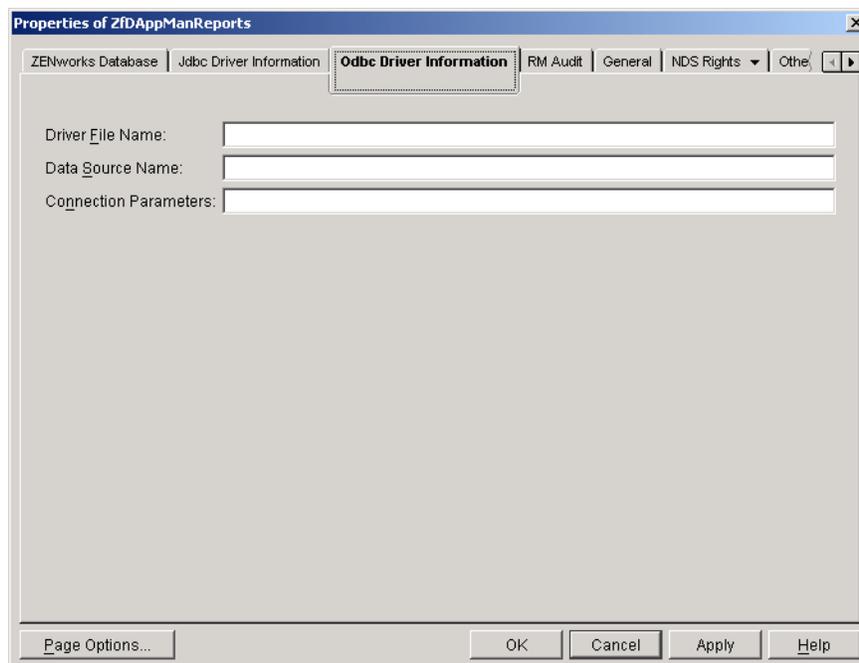
- 7 Click *OK* to save the information.

44.1.4 Configuring the Database's ODBC Information

Application Launcher uses an ODBC driver to write event information to the database. You need to configure the database object with the ODBC driver information that Application Launcher needs.

To provide the ODBC information:

- 1 Right-click the database object, then click *Properties*.
- 2 Click the *ODBC Driver Information* tab.



- 3 Fill in the following fields:

Driver file name: Specify the name of the workstation ODBC driver file that Application Launcher will use to access the database. For Sybase, you can leave the field empty, in which case Application Launcher automatically detects the installed driver.

Data source name: Enter NAL Reporting.

Connection parameters: Specify the appropriate commands to connect to the database. For Sybase, these commands are:

```
CommLinks=TCPIP{Host=IPAddress:2638};AutoStop=Yes;Integrated=No;DB
N=NAL;ENG=IPAddress
```

where *IPAddress* is the actual IP address of the server where the database resides.

- 4 Click *OK* to save the ODBC driver information.

44.1.5 Granting ODBC Property Rights to Users

You need to grant users Read and Compare rights to the ODBC properties you defined for the database object in the previous section. This allows Application Launcher to retrieve the ODBC information it needs to access the database.

To grant rights:

- 1 Right-click the ZENworks Database object, then click *Trustees of this object*.
- 2 Click *Add trustee*.
- 3 Select *[PUBLIC]*, then click *OK* to add *[PUBLIC]* to the list of trustees.

Adding *[PUBLIC]* as a trustee gives every user Read and Compare rights to all of the database object's properties, including the various user names and passwords that can be used to access the database. To avoid this, you need to limit the *[PUBLIC]* access to the three ODBC properties.

- 4 In the Property list, select *[All Attribute Rights]*, then click *Delete property* to remove it from the list.
- 5 Click *Add property* to display the Add Property dialog box, select *zendbODBCConnectionParameters*, then click *OK* to add it to the Property list.

The default rights, Read and Compare, are sufficient. You do not need to change these rights.

- 6 Repeat **Step 5** to add the following properties (keep the default rights: Read and Compare):

```
zendbODBCDataSourceName
zendbODBCDriverFileName
Host Server
Network Address
zendbUser
zendbPassword
```

- 7 Click *OK*, then click *OK* again to save the changes.

44.1.6 Enabling the ZENworks Database Policy

Before Application Launcher can use the database, you need to:

- ♦ In ConsoleOne, activate a Database policy in a Service Location Package. The Database policy simply points to the location of the ZENworks database you are using. A Service Location Package can have only one Database policy, and each Database policy can point to only one database. If you haven't created a Service Location Package, or if your current packages' Database policies are being used for other databases, you need to create a new Service Location Package. Instructions are provided in the steps below.

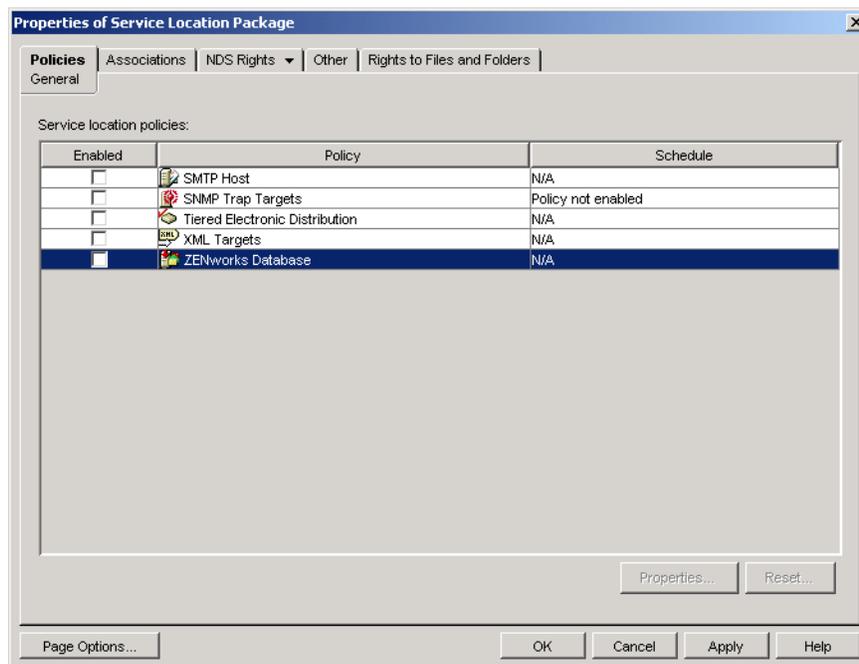
- ◆ Associate the Service Location Package with the containers where the users or workstations reside whose reports you want sent to the database. This association is how Application Launcher knows which database to use when reporting events for a user or workstation.

To activate a ZENworks Database policy and associate containers with it:

- 1 In ConsoleOne[®], right-click the Service Location Package you want to use, then click *Properties*.

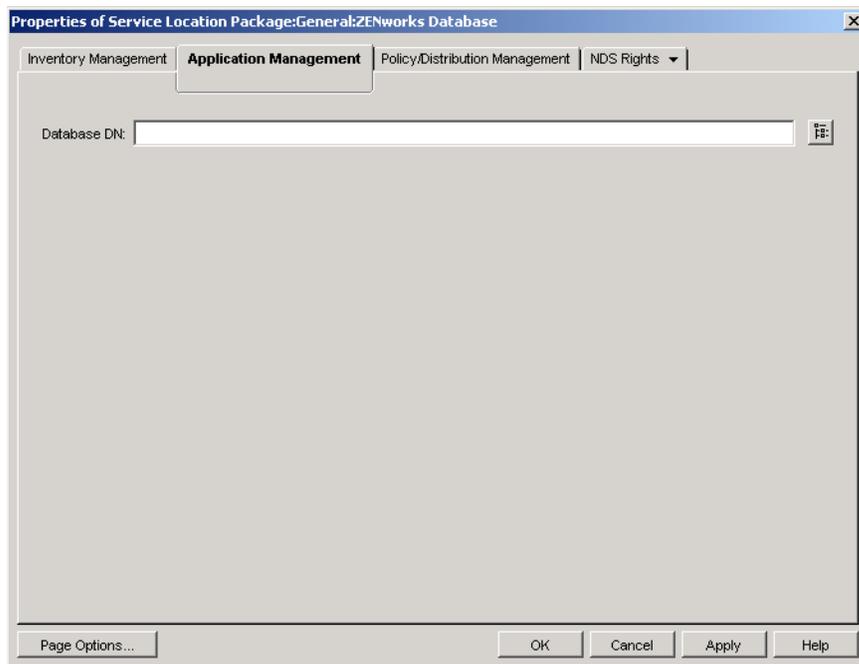
or

If you do not have a Service Location Package, right-click the container where you want to create one, click *New*, then click *Policy Package*. Follow the instructions provided in the Policy Package Wizard to create a Service Location Package.

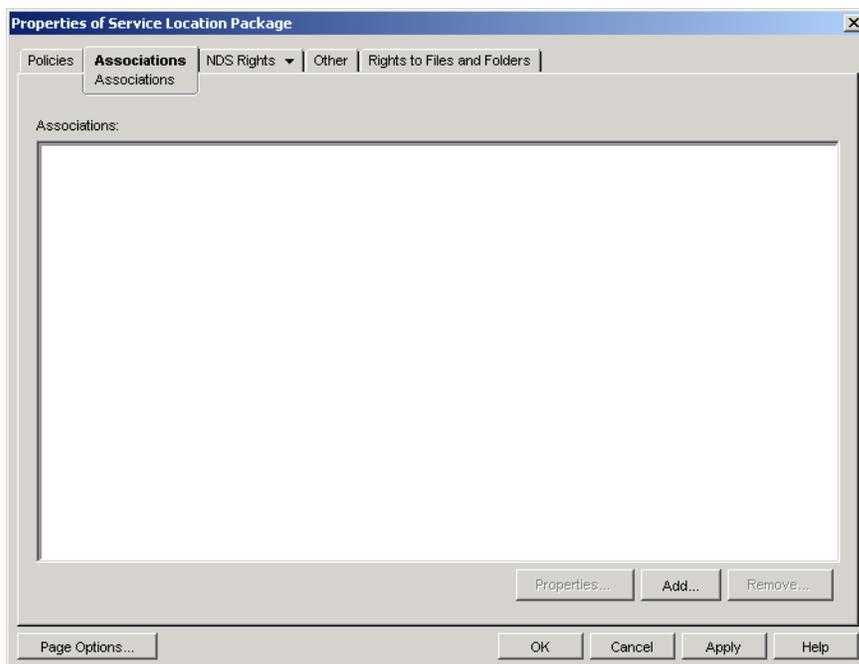


- 2 On the General page, select the check box in the *Enabled* column to enable the ZENworks Database Policy.

- 3 Select the *ZENworks Database Policy* in the list, click *Properties* to display the ZENworks Database properties page, then click the *Application Management* tab to display the Application Management page.



- 4 In the *Database DN* field, browse for and select the database object you want to use for application event reporting. This should be the same object you configured in [“Configuring the Database's ODBC Information”](#) on page 444.
- 5 Click *OK* to return to the General page.
- 6 Click *Associations* to display the Associations page.



You use this page to associate the Service Location Package with the containers where the users or workstations reside whose reports you want sent to the database. If a parent container includes several containers where User and Workstation objects reside, you can select the parent container rather than each individual container.

- 7 Click *Add*, browse for and select the container you want to add, then click *OK* to add it to the list.
- 8 Repeat [Step 7](#) to add additional containers.
- 9 When you've finished adding containers, click *OK* to save the information.

44.1.7 Configuring Application Objects to Use Database Reporting

After you've set up the database you will use for reporting, you can start configuring the Application objects for which you want events reported. Because event reporting is configured on a per-application basis, you can choose which applications you want to collect event reports for and which ones you don't.

For detailed instructions about configuring an application to use database reporting, SNMP trap reporting, or log file reporting, see [Section 44.5, "Configuring Applications to Use Reporting," on page 458](#).

44.2 Setting Up SNMP Trap Reporting

If you have a management console to collect SNMP traps, you can have Application Launcher send SNMP traps to the management console. ZENworks Server Management includes a management console you can use to collect SNMP traps. For information, see "[Management and Monitoring Services](#)" in the *Novell ZENworks 7 Server Management Administration Guide*.

To set up SNMP trap reporting, you need to complete the following tasks:

- ♦ [Section 44.2.1, "Enabling Firewall Access," on page 448](#)
- ♦ [Section 44.2.2, "Enabling the SNMP Trap Targets Policy," on page 448](#)
- ♦ [Section 44.2.3, "Configuring Application Objects to Use SNMP Trap Reporting," on page 451](#)

44.2.1 Enabling Firewall Access

For SNMP traps to be sent from workstations to your management console, you need to ensure that there are no firewalls between the workstations and the management console that filter UDP or TCP on port 161. This includes both personal firewalls on the user workstations and network firewalls.

44.2.2 Enabling the SNMP Trap Targets Policy

Before Application Management can use SNMP traps for event reporting, you need to:

- ♦ In ConsoleOne, activate an SNMP Trap Targets policy in a Service Location Package. The SNMP Trap Targets policy simply points to the IP address (or addresses) of the management console that displays the traps. A Service Location Package can have only one SNMP Trap Targets policy. If you haven't created a Service Location Package, or if your current packages'

SNMP Trap Targets policies are being used for other databases, you need to create a new Service Location Package. Instructions are provided in the steps below.

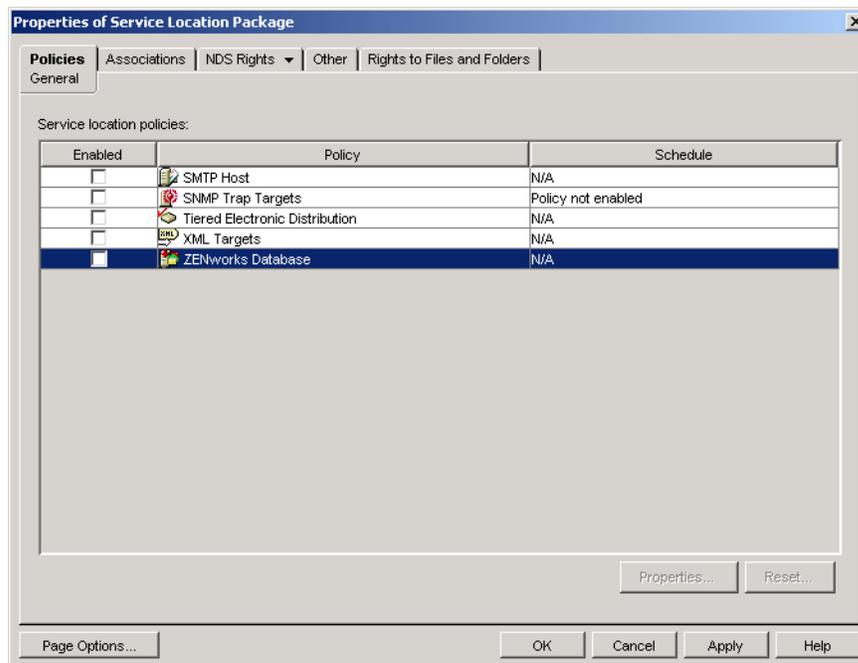
- ◆ Associate the Service Location Package with the containers where the users or workstations reside that you want the SNMP Trap Targets policy applied to. This association is how Application Launcher knows which SNMP trap target to use when reporting events for the users or workstations.

To activate an SNMP Trap Targets policy and associate it with containers:

- 1 In ConsoleOne, right-click the Service Location Package you want to use, then click *Properties*.

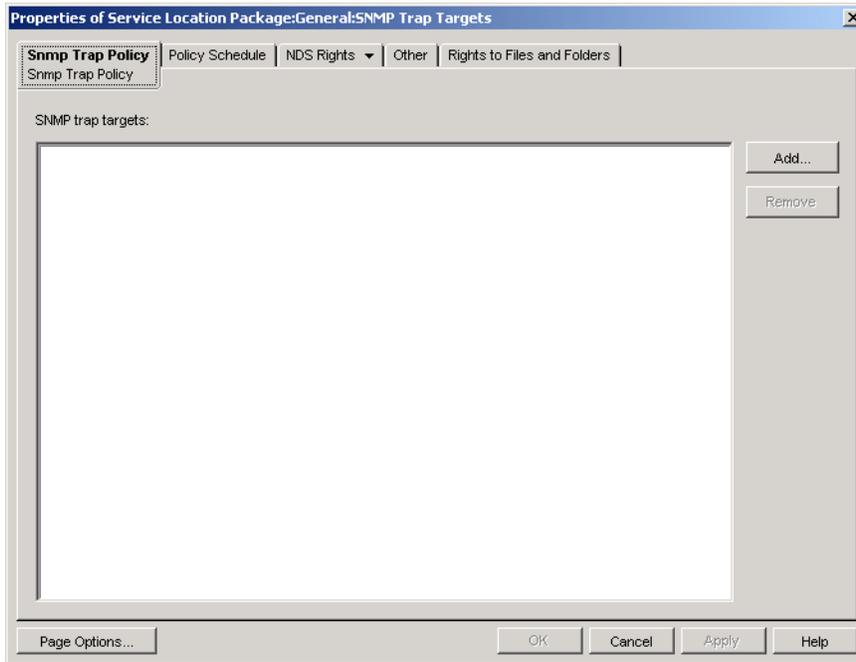
or

If you do not have a Service Location Package, right-click the container where you want to create one, click *New*, then click *Policy package*. Follow the instructions provided in the Policy Package Wizard to create a Service Location Package.

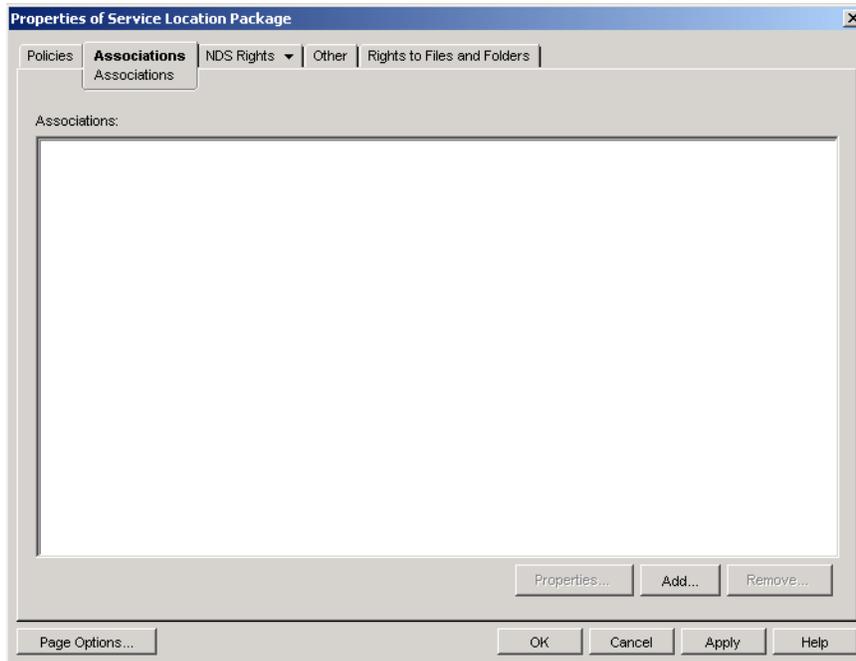


- 2 On the General page, select the box in the *Enabled* column to enable the SNMP Trap Targets Policy.

- 3 Select the policy in the list, then click *Properties* to display the SNMP Trap Targets list.



- 4 Click *Add* to display the SNMP Target dialog box, specify the IP address of the workstation or server where the management console is running, then click *OK* to add the IP address to the list.
- 5 Repeat **Step 4** to add additional targets.
- 6 When you've finished adding targets, click *OK* to return to the General page.
- 7 Click *Associations* to display the Associations page.



You use this page to associate the Service Location Package with the containers where the users or workstations reside that you want the SNMP Trap Targets policy applied to. If a parent container includes several containers where User and Workstation objects reside, you can select the parent container rather than each individual container.

- 8 Click *Add*, browse for and select the container you want to add, then click *OK* to add it to the list.
- 9 Repeat **Step 8** to add additional containers.
- 10 When you've finished adding containers, click *OK* to save the information.

When ZENworks Workstation Manager enforces the SNMP Trap Targets policy on a user's workstation, it adds the SNMP trap targets as values on the following registry key:

```
HKCU\Software\NetWare\NAL\1.0\Reporting\tree_name\TrapTargets\User
```

If events are not being sent to the SNMP management console, you should verify that the trap targets have been added to the registry.

44.2.3 Configuring Application Objects to Use SNMP Trap Reporting

After you've enabled the SNMP Trap Targets policy and have your management console running, you can start configuring Application objects to use SNMP trap reporting. Because event reporting is configured on a per-application basis, you can choose which applications you want to collect event reports for and which ones you don't.

For detailed instructions about configuring an application to use SNMP trap reporting, database reporting, or log file reporting, see [Section 44.5, "Configuring Applications to Use Reporting," on page 458](#).

44.3 Setting Up Log File Reporting

You can have Application Launcher record events to a log file. This can be an individual log file located on the user's workstation or a common log file on a network server. When using a common log file, users must be given Read and Write rights to the log file, but Application Launcher automatically authenticates them to the log file location.

To set up log file reporting, you need to complete the following tasks:

- ♦ [Section 44.3.1, "Setting Up a Common Log File Location," on page 451](#)
- ♦ [Section 44.3.2, "Configuring Application Objects to Use Log File Reporting," on page 452](#)

44.3.1 Setting Up a Common Log File Location

With log file reporting, you have two options. You can have Application Launcher log events for each individual user to a file on the user's local drive, or you can have Application Launcher log events for all users to a file in a common network location.

If you want Application Launcher to log events to a file in a common network location, you need to establish the network directory and provide users with Read and Write rights to files in the directory.

Because log file names are established on a per-application basis, you can have individual log files for each application (by specifying a different log file name for each Application object) or one log

file for all applications (by specifying the same log file name for all Application objects). You cannot have log files on a per-user basis, unless you have Application Launcher save the files to the users' local drives.

44.3.2 Configuring Application Objects to Use Log File Reporting

After you've set up a common log file location (if you plan to use a common log file), you can start configuring Application objects to use log file reporting. Because event reporting is configured on a per-application basis, you can choose which applications you want to collect event reports for and which ones you don't.

For detailed instructions about configuring an application to use log file reporting, database reporting, or SNMP trap reporting, see [Section 44.5, “Configuring Applications to Use Reporting,” on page 458](#).

44.4 Setting Up XML Reporting

Application Launcher can use the HTTP or HTTPS protocol to send event information as XML data to the Desktop Management Application Reporting servlet (`zfdamrServlet`). The servlet processes the messages and adds them to a database compliant with JDBC, such as the Sybase database included with ZENworks.

To set up XML reporting, you need to complete the following tasks:

- ◆ [Section 44.4.1, “Meeting the Reporting Servlet Requirements,” on page 452](#)
- ◆ [Section 44.4.2, “Installing the Reporting Servlet on Linux,” on page 453](#)
- ◆ [Section 44.4.3, “Installing the Reporting Servlet on NetWare and Windows,” on page 453](#)
- ◆ [Section 44.4.4, “Configuring the Reporting Servlet to be Used with an Apache Web Server,” on page 454](#)
- ◆ [Section 44.4.5, “Configuring the Reporting Servlet to Write to the Database,” on page 454](#)
- ◆ [Section 44.4.6, “Enabling the XML Targets Policy,” on page 455](#)
- ◆ [Section 44.4.7, “Configuring Application Objects to Use XML Reporting,” on page 458](#)

44.4.1 Meeting the Reporting Servlet Requirements

The Reporting servlet requires one of the following server environments. The version numbers listed are minimum requirements.

Table 44-1 Reporting Servlet Requirements

Server	Java Servlet Engine	JVM
NetWare® 6.5 (Support Pack 1 or later)	Tomcat 4.1x	Novell JVM* for NetWare 1.4.2
NetWare 6 (Support Pack 2 or later) ¹	Tomcat 3.3a	Novell JVM for NetWare 1.3.1
Windows Server 2003 (with latest service pack)	Tomcat 3.3a	Sun JDK* 1.3.1_01
Windows 2000 (with latest service pack)	Tomcat 3.3a	Sun JDK 1.3.1_01

Server	Java Servlet Engine	JVM
SUSE Linux Enterprise Server (SLES) 9 SP1	Tomcat 4.1x	JVM 1.4.2 ²
Novell Open Enterprise Server (Linux) 1.0	Tomcat 4.1x	JVM 1.4.2 ³

¹ Support Pack 2 or later is required. Without Support Pack 2 or later, the Reporting servlet causes server utilization to jump to 100% and remain there.

² The JVM is installed on the SLES 9 SP1 server by the ZENworks 7 Middle Tier Server installation program. If you want to use a SLES 9 SP1 server other than the Middle Tier Server, you will need to install the correct Tomcat and JVM versions manually.

³ The JVM ships with and is installed by the Novell Open Enterprise Server installation program.

For information about installing Tomcat on NetWare 6.5, see the [NetWare 6.5 documentation \(http://www.novell.com/documentation/nw65/index.html\)](http://www.novell.com/documentation/nw65/index.html).

For information about installing Tomcat on NetWare 6, see the [NetWare 6 documentation \(http://www.novell.com/documentation/lg/nw6p\)](http://www.novell.com/documentation/lg/nw6p).

For information about installing Tomcat on Windows 2000, see the [Apache Tomcat Web site \(http://jakarta.apache.org/tomcat\)](http://jakarta.apache.org/tomcat).

OES Linux ship with Tomcat installed.

44.4.2 Installing the Reporting Servlet on Linux

- 1 Make sure the server environment where you are installing the Reporting servlet meets the software requirements listed in the previous section, [Meeting the Reporting Servlet Requirements](#).
- 2 Copy the `zfdamrServlet.war` file from the Desktop Management server's `/opt/novell/zenworks/zdm/winutils/nalreporting` directory to `/var/opt/novell/tomcat4/webapps`.
- 3 Restart Tomcat to expand the `zfdamrServlet.war` file. To restart Tomcat on Linux, enter `/usr/share/tomcat/bin/startup.sh` at the command line.

or

If you don't want to restart Tomcat, use the tools provided with Tomcat to expand the file.

44.4.3 Installing the Reporting Servlet on NetWare and Windows

- 1 Make sure the server environment where you are installing the Reporting servlet meets the software requirements listed in the previous section, [Meeting the Reporting Servlet Requirements](#).
- 2 Copy the `zfdamrServlet.war` file from the Desktop Management server's `sys:\public\reporting\canned\nal reports` directory to `sys:\tomcat\4\webapps`.
- 3 Restart Tomcat to expand the `zfdamrServlet.war` file.

or

If you don't want to restart Tomcat, use the tools provided with Tomcat to expand the file.

44.4.4 Configuring the Reporting Servlet to be Used with an Apache Web Server

This section applies only if you are using an Apache Web server.

- 1 Open the `sys:/apache2/conf/httpd.conf` file.
- 2 Add the following line to the bottom of the file:

```
include "sys:/public/Reporting/canned/NaI Reports/zfdamrServlet.conf"
```
- 3 Save the `httpd.conf` file.
- 4 Restart the Apache Web server.

44.4.5 Configuring the Reporting Servlet to Write to the Database

Before completing the steps in this section, you need to ensure that the database to which the Reporting servlet will write is set up and configured properly.

The Reporting servlet needs to know information about the database to which it will write the reporting events. To provide this information:

- 1 Open the `web.xml` file. The file is located in the Tomcat `webapps\zfdamrservlet\web-inf` directory
- 2 If you are using a Sybase database, locate the `dbip` parameter, then change the parameter value to the IP address of the database server. The IP address is the only parameter you need to modify for a Sybase database.

or

If you are using a database other than Sybase, modify the parameters, described below, as required by your database. If you are using a Microsoft SQL database, follow the directions in [TID 10094306 \(http://support.novell.com/cgi-bin/search/searchtid.cgi?/10094306.htm\)](http://support.novell.com/cgi-bin/search/searchtid.cgi?/10094306.htm).

dbuser: Specify a user name that has Write access to the database. Read access is not required. The preconfigured user name (`MW_DBA`) is the default Read-Write user name for a Sybase database created by the Desktop Management Server installation program. If you are using a ZENworks Sybase database and have not changed the default Read-Write username, you do not need to modify this parameter.

dbpasswd: Specify the password for the user name specified in the `dbuser` parameter. The preconfigured password (`novell`) is the default Read-Write user's password for a Sybase database created by the Desktop Management Server installation program. If you are using a ZENworks Sybase database and have not changed the default Read-Write user's password, you do not need to modify this parameter.

dbip: Specify the IP address of the server running the database.

dbport: Specify the port where the database is listening. The preconfigured port (2638) is the default NetWare server port for a Sybase database created by the Desktop Management Server installation program. If you are using a ZENworks Sybase database installed on NetWare and have not changed the default port, you do not need to modify this parameter.

dbprotocol: Specify the protocol to use when accessing the database. For Sybase, this is jdbc: (the preconfigured value).

dbsubprotocol: Specify the subprotocol to use when accessing the database. For Sybase, this is sybase: (the preconfigured value).

dbsubname: Specify the subname to use when accessing the database. For Sybase, this is Tds: (the preconfigured value).

dbdriver: Specify the full class of the driver to use when accessing the database. The driver must support the protocol listed in the dbprotocol parameter. For Sybase, this is `com.sybase.jdbc2.jdbc.SybDriver` (the preconfigured value).

3 Save the file.

4 Restart Tomcat.

44.4.6 Enabling the XML Targets Policy

Before Application Launcher can report application events through XML, you need to:

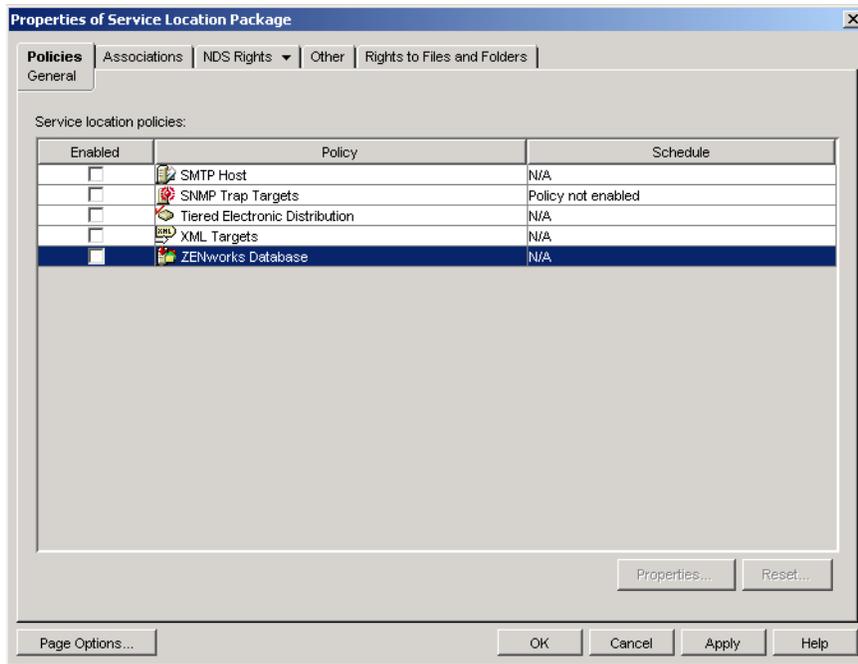
- ◆ In ConsoleOne, activate an XML Targets policy in a Service Location Package. The XML Targets policy identifies the Web locations (URLs) where you want the XML data sent. If you are using the Application Management Reporting servlet, this is the URL of that servlet. A Service Location Package can have only one XML Targets policy. If you haven't created a Service Location Package, or if your current packages' XML Targets policies are being used to specify other locations, you need to create a new Service Location Package. Instructions are provided in the steps below.
- ◆ Associate the Service Location Package with the containers where the users or workstations reside whose reports you want sent to the Web URL.

To activate an XML Targets policy and associate containers with it:

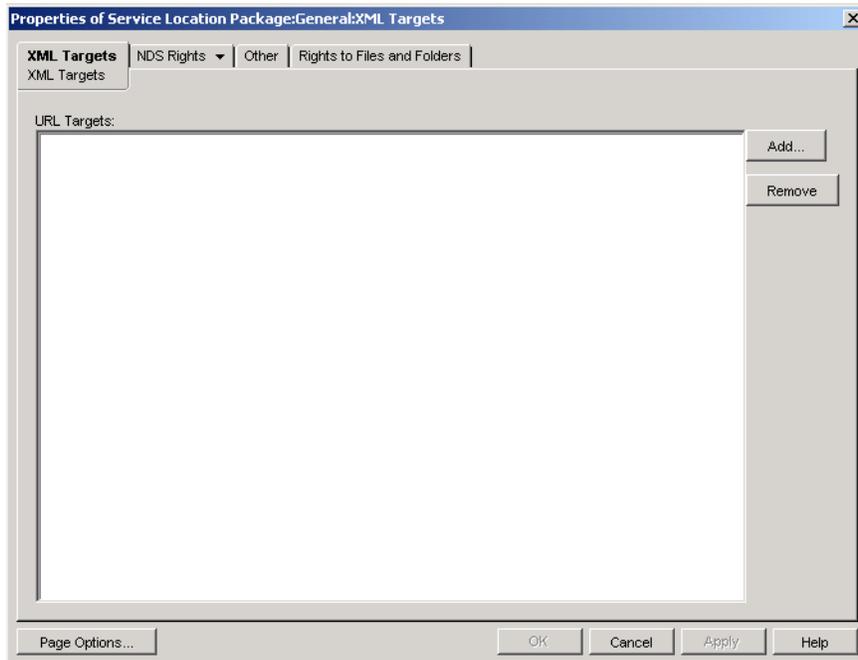
1 In ConsoleOne, right-click the Service Location Package you want to use, then click *Properties*.

or

If you do not have a Service Location Package, right-click the container where you want to create one, click *New*, then click *Policy package*. Follow the instructions provided in the Policy Package Wizard to create a Service Location Package.



- 2 On the General page, select the box in the *Enabled* column to enable the XML Targets policy.
- 3 Select the *XML Targets* policy in the list, then click *Properties* to display the XML Targets properties page.



- 4 Click *Add* to display the XML URL dialog box, enter the URL where you want the XML data sent, then click *OK* to add the URL to the list.

The URL needs to include the port number, which by default is port 8080 on Windows or on NetWare, and 8180 on OES Linux and SLES. This port is the default Tomcat port in these environments, and is necessary for the Service Location Policy for XML reporting to find the `zfdamrServlet` running on Tomcat.

The URL for the Reporting servlet on Windows or NetWare would be:

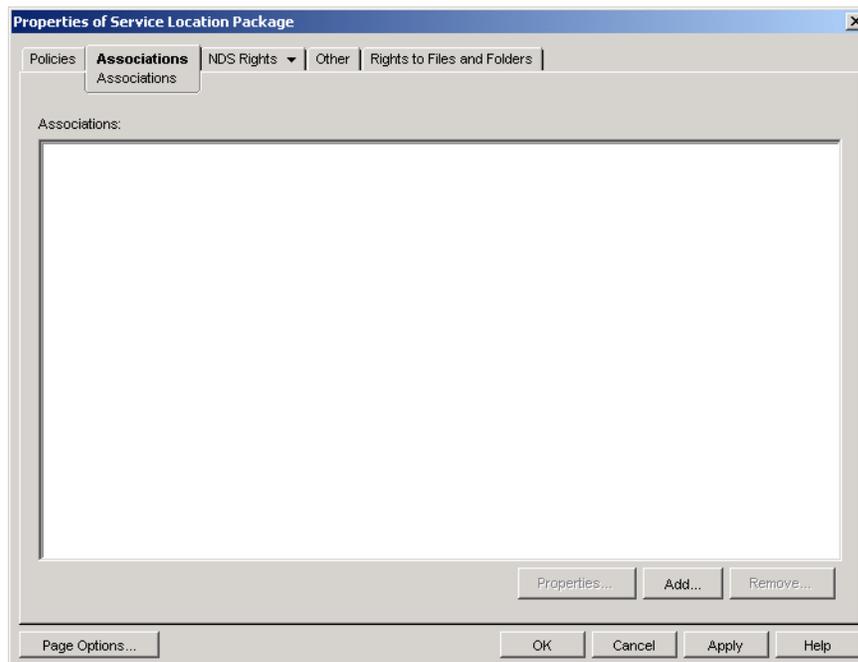
```
http://ip_address:8080/zfdamrServlet/run
```

and the URL for the Reporting servlet on Linux (OES Linux or SLES) would be:

```
http://ip_address:8180/zfdamrServlet/run
```

where *ip_address* is the IP address or DNS host name of the server where the Reporting servlet is running.

- 5 Repeat **Step 4** to add additional URLs.
- 6 When you are finished adding URLs, click *OK* to return to the General page.
- 7 Click *Associations* to display the Associations page.



You use this page to associate the Service Location Package with the containers where the users or workstations reside whose XML reports you want sent to the Web URL. If a parent container includes several containers where User and Workstation objects reside, you can select the parent container rather than each individual container.

- 8 Click *Add*, browse for and select the container you want to add, then click *OK* to add it to the list.
- 9 Repeat **Step 8** to add additional containers.
- 10 When you've finished adding containers, click *OK* to save the information.

44.4.7 Configuring Application Objects to Use XML Reporting

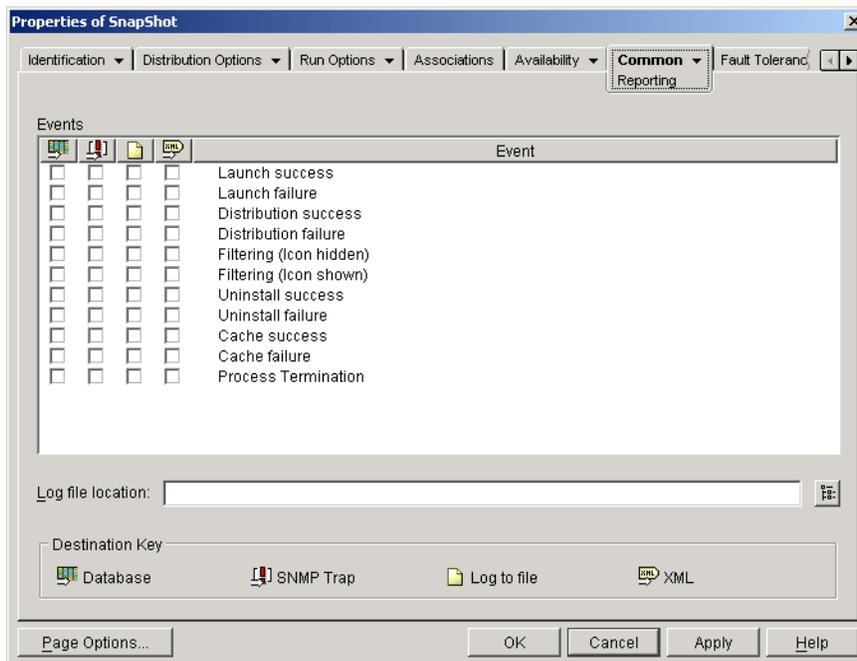
After you've enabled the XML Targets policy, you can start configuring Application objects to use XML reporting. Because event reporting is configured on a per-application basis, you can choose which applications you want to collect event reports for and which ones you don't.

For detailed instructions about configuring an application to use XML reporting, see [Section 44.5, "Configuring Applications to Use Reporting,"](#) on page 458.

44.5 Configuring Applications to Use Reporting

Application Launcher must know which method of reporting (database, SNMP trap, or log file) to use for an application and which events to report for the application. You provide this information through the Application object's Reporting page (Common tab).

- 1 In ConsoleOne, right-click the Application object for which you want to configure reporting, then click *Properties*.
- 2 Click the *Common* tab > *Reporting* to display the Reporting page.



- 3 Fill in the following fields:

Events: Select the events (launch, distribution, filtering, uninstall, cache, and process termination) that you want reported and the destination (database, SNMP trap, log file, or XML) where you want the events sent. If necessary, you can send events to multiple destinations (for example, both a database and a log file). Events and destinations are described in the following tables:

Event	Description
Launch Success	Occurs when a user double-clicks the Application object and Application Launcher successfully starts the application.

Event	Description
Launch Failure	Occurs when a user double-clicks the Application object and Application Launcher cannot start the application.
Distribution Success	Occurs when Application Launcher successfully modifies the workstation to support the application. Modifications include installation of files, changing of settings (registry, INI, etc.), creation of shortcuts, and so forth.
Distribution Failure	Occurs when Application Launcher cannot modify the workstation to support the application. Modifications include installation of files, changing of settings (registry, INI, etc.), creation of shortcuts, and so forth.
Uninstall Success	Occurs when Application Launcher successfully uninstalls the application from the workstation.
Uninstall Failure	Occurs when Application Launcher cannot uninstall the application from the workstation.
Cache Success	Occurs when Application Launcher successfully caches the application to the workstation.
Cache Failure	Occurs when Application Launcher cannot cache the application to the workstation.
Filter (Icon Hidden)	Occurs when Application Launcher cannot display an Application object on a workstation because the workstation does not meet one or more of the Application object's distribution rules (Application object > <i>Availability</i> tab > <i>Distribution Rules</i> page) and the <i>Always show icon</i> option (on the <i>Distribution Rules</i> page) is turned off. The Application object's icon is hidden, or not shown, on the workstation.
Filter (Icon Shown)	Occurs when Application Launcher can only display a disabled (dimmed) Application object on a workstation. This occurs because the workstation does not meet one or more of the distribution rules (Application object > <i>Availability</i> tab > <i>Distribution Rules</i> page) and the <i>Always show icon</i> option (on the <i>Distribution Rules</i> page) is turned on. Users can right-click the disabled icon and click <i>Details</i> to see what system requirements are not being met.
Process Termination	Occurs when a user or Application Launcher terminates the application.

Destination	Description
 Database	<p>Application Launcher can write events to any ODBC-compliant database (for example, the Sybase database included with ZENworks). To use a database, you must also:</p> <ul style="list-style-type: none"> ◆ Create a ZENworks Database object to use for application event reporting. ◆ If necessary, create a Service Location Package. Associate the Service Location Package with the containers where the users or workstations reside whose reports you want sent to the database. ◆ Enable the ZENworks Database policy in the Service Location Package. ◆ Associate the ZENworks Database policy with the ZENworks Database object. ◆ Make sure users have the appropriate ODBC database driver installed and configured. ◆ After you've set up database reporting, you can use one of the predefined reports to see information about specific application events. To access the reports, right-click the ZENworks Database object you created for application event reporting > click <i>Reporting</i>.
 SNMP Traps	<p>Application Launcher can send SNMP traps to any SNMP management console. To use SNMP traps, you must also enable an SNMP Trap Target policy in a Service Location Package. It might be necessary to first create the Service Location Package.</p> <ul style="list-style-type: none"> ◆ Associate the Service Location Package with the containers where the users or workstations reside that you want the SNMP Trap Target policy applied to. ◆ Add the SNMP Trap Targets (IP addresses) for the locations you want the traps sent. ◆ Have a management console that displays the SNMP traps.
 Text Log File	<p>Application Launcher can write events to a text log file. You use the Log File Location field (described below) to specify the location of the log file.</p>
 XML	<p>Application Launcher can send events, as XML data, to a URL using the standard HTTP or HTTPS protocol. XML reporting is the recommended method if you want Application Launcher to report events for users located outside of your firewall.</p> <p>To use XML, you must also:</p> <ul style="list-style-type: none"> ◆ Enable an XML URL policy in a Service Location Package. It might be necessary to first create the Service Location Package. ◆ Associate the Service Location Package with the containers where the users or workstations reside whose reports you want sent as XML data. ◆ Make sure you've set up the XML processing mechanism and data storage mechanism. For example, if you are using the Application Management Reporting servlet and the Sybase database, make sure that each one is set up and configured properly.

Log file location: If you selected a log file as the destination for any of the event reports, enter (or browse to and select) the location and name for the log file. You can specify a local drive or

a network drive. If you enter a location that does not exist, Application Launcher creates it. Do not use extended characters in the path; extended characters are not supported.

If you want Application Launcher to log events to a file in a common network location, you need to establish the network directory and provide users with Read and Write rights to files in the directory. Because log file names are established on a per-application basis, you can have individual log files for each application (by specifying a different log file name for each Application object) or one log file for all applications (by specifying the same log file name for all Application objects).

IMPORTANT: You should only use log file reporting if users have a network client, such as the Novell Client™ or Microsoft Client for Networks, that enables writing to a NetWare, Linux, or Windows server, or if you specify a local drive as the log file location.

- 4 Click *OK* to save the changes.

44.6 Generating Reports from a Database

You can use the predefined reports included with ZENworks Database Management to view information about the application events recorded by Application Launcher, or you can use your own database tools to create your own custom reports. The following sections provide information:

- ♦ [Section 44.6.1, “Predefined Reports,” on page 461](#)
- ♦ [Section 44.6.2, “Custom Reports,” on page 461](#)
- ♦ [Section 44.6.3, “Database Tables and Fields,” on page 462](#)

44.6.1 Predefined Reports

ZENworks Desktop Management includes predefined reports for successful application events and failed application events.

To generate a predefined report:

- 1 Right-click the Desktop Management Database object, then click *Reporting*.
- 2 In the *Available reports* list, expand the *ZENworks Application Launcher* category, then select the report you want.
- 3 Click *Run selected report*.

44.6.2 Custom Reports

Using your database tools of choice, you can create custom database reports to search for information not included in the predefined reports, or to generate different report formats. The following samples are SQL queries you can use to generate reports from your database tools if you are using the Sybase database included with ZENworks. Refer to [“Database Tables and Fields” on page 462](#) for additional information.

All Fields for an Event

The following queries return all information fields for a successful, failed, or rogue process management event stored in the database. Events are sorted by the order in which they were entered into the database.

```
SELECT * FROM T_SUCCESS
SELECT * FROM T_FAILURE
SELECT * FROM T_INFO
```

All Fields for an Event, Sorted on a Specific Field

To sort the list on a specific field, add `ORDER BY field_name`, as in the following examples:

```
SELECT * FROM T_SUCCESS ORDER BY zenWSTDN
SELECT * FROM T_FAILURE ORDER BY zenAppTDN
SELECT * FROM T_INFO ORDER BY zenUserTDN
```

The valid field names are listed in the table in [“Database Tables and Fields” on page 462](#)

All Fields for Specific Event Types

To include only a specific event type (launch, distribute, etc.) for an event (success, failure, etc.), add `WHERE zenEventType="event_type"`, as in the following examples:

```
SELECT * FROM T_SUCCESS WHERE zenEventType="Launch Success"
SELECT * FROM T_FAILURE WHERE zenEventType="Launch Failure"
SELECT * FROM T_INFO WHERE zenEventType="Process Terminated"
```

The valid event types are listed in the table in [“Database Tables and Fields” on page 462](#)

Specific Fields for an Event

To include only specific fields, replace the `*` with a comma-delimited field list, as in the following example:

```
SELECT zenEventType, zenDateTime, zenUserTDN, zenAppTDN FROM T_SUCCESS
WHERE zenEventType="Cache Success" ORDER BY zenUserTDN
```

44.6.3 Database Tables and Fields

The database includes the following three tables:

- ♦ **T_Success:** Stores information about successful events.
- ♦ **T_Failure:** Stores information about failed events.
- ♦ **T_Info:** Stores information about rogue process management events. For information about rogue process management, see [Chapter 41, “Controlling Rogue Processes,” on page 425](#).

Each database table contains as many as 17 information fields. The following table lists the fields and the database tables that include the fields. The data type for all fields is `varchar(256)`.

Table 44-2 Database Tables and Fields

Field	Tables	Description
zenEventType	T_Success T_Failure T_Info	<p>The event that occurred.</p> <ul style="list-style-type: none"> ◆ Successful event types (T_Success) are: <ul style="list-style-type: none"> Launch Success Distribute Success Uninstall Success Cache Success Process Terminated ◆ Failed event types (T_Failure) are: <ul style="list-style-type: none"> Launch Failed Distribute Failed Uninstall Failed Cache Failed Filter Hide Filter Show ◆ Rogue process management event types (T_Info) are: <ul style="list-style-type: none"> Process Terminated Process Ignored
zenDateTime	T_Success T_Failure T_Info	The date and time the event occurred.
zenUserTDN	T_Success T_Failure T_Info	The distinguished name and tree of the user for which the event occurred.
zenWSTDN	T_Success T_Failure T_Info	The distinguished name and tree of the workstation on which the event occurred. If the workstation has not been imported into eDirectory as a Workstation object, the field contains WORKSTATION NOT REGISTERED.
zenWSAddr	T_Success T_Failure T_Info	The IPX™ or IP address of the workstation on which the event occurred.
zenAppTDN	T_Success T_Failure T_Info	<p>The distinguished name and tree of the Application object for which the event occurred.</p> <p>Because rogue processes do not have an Application object, this field is always “ZEN Process Management” in the T_Info table.</p>
zenAppGUID	T_Success T_Failure T_Info	<p>The global unique identifier assigned to the Application object. The GUID is located on the Application object's Options page (Distribution Options tab).</p> <p>Because rogue processes do not have an Application object, this field is always empty in the T_Info table.</p>

Field	Tables	Description
zenAppVer	T_Success T_Failure T_Info	The version number assigned to the Application object. Possible numbers range from 0 to 65535. The version number is located on the Application object's Options page (Distribution Options tab). Because rogue processes do not have an Application object, this field is always empty in the T_Info table.
zenMajor	T_Success T_Failure T_Info	For successful events (T_Success table), this field is always 0. For failed events (T_Failure table), this field lists the error code generated by Application Launcher. For rogue process events (T_Info table), this field is left blank.
zenMinor	T_Success T_Failure T_Info	For successful events (T_Success table), this field is always 0. For failed events (T_Failure table), this field lists additional error code information. For rogue process events (T_Info table), this field is left blank.
zenEventString1	T_Failure T_Info	For failed events (T_Failure table), this field might list additional information describing the reason for the failure. For rogue process events (T_Info table), this field lists the executable path for the rogue process.
zenEventString2	T_Failure T_Info	For failed events (T_Failure table), this field might list additional information describing the reason for the failure. For rogue process events (T_Info table), this field lists the original filename of the process.
zenEventString3	T_Failure T_Info	For failed events (T_Failure table), this field might list additional information describing the reason for the failure. For rogue process events (T_Info table), this field contains the process ID (PID).
zenEventString4	T_Failure T_Info	For failed events (T_Failure table), this field might list additional information describing the reason for the failure. For rogue process events (T_Info table), this field contains the parent process ID (parent PID).
zenEventString5	T_Failure T_Info	For failed events (T_Failure table), this field might list additional information describing the reason for the failure. For rogue process events, this field contains the event action, either "process ignored" or "process terminated successfully."
zenAppFlags	T_Success T_Failure T_Info	For successful events (T_Success table) and failed events (T_Failure table), this field lists the Application object bitmask. For rogue process events (T_Info table), this field is always 0.

44.7 Understanding Log File Reports

The following is a log file entry for a single event. Each field in the entry is described below.

```
"Launch Failure","11","7/25/2002 9:27:52 AM",
"JSMITH.NOVELL.NOVELL_TREE",".WORKSTATION NOT
REGISTERED","137.65.45.25","NOTEPAD.APPS.NOVELL.NOVELL_TREE","3054A94E
-BBFF-4851-9D8E-58973623B728","2","Could not launch
NOTEPAD.APPS.NOVELL.NOVELL_TREE (using c:\winnt\notepa) (id=123)","The
filename, directory name, or volume label syntax is
incorrect.", "c:\winnt\notepa", "", "", "", "", "524288"
```

Table 44-3 Log File Fields and Descriptions

Field	Example	Description
Event Type	Launch Failure	The event that occurred and whether it was successful or failed. Possible event types are: Launch Success Launch Failure Distribution Success Distribution Failure Filter Show Filter Hide Uninstall Success Uninstall Failure Cache Success Cache Failure Application Termination Process Ignored Process Terminated
Event Type Code	11	The code associated with the event. Possible codes are: 10 - Launch Success 11 - Launch Failure 20 - Distribution Success 21 - Distribution Failure 30 - Filter Hide 40 - Filter Show 50 - Uninstall Success 51 - Uninstall Failure 60 - Cache Success 61 - Cache Failure 70 - Application Termination 80 - Process Ignored 81 - Process Terminated
Date and Time	7/25/2002 9:27:52 AM	The date (7/25/2002) and time (9:27:52 AM) the event occurred.
User Distinguished Name and Tree	JSMITH.NOVELL.NOVELL_TREE	The distinguished name and tree of the user for which the event occurred.
Workstation Distinguished Name and Tree	.WORKSTATION NOT REGISTERED	The distinguished name and tree of the workstation on which the event occurred. If the workstation has not been imported into eDirectory as a Workstation object, the field contains WORKSTATION NOT REGISTERED.

Field	Example	Description
Workstation Address	137.65.45.25	The IPX or IP address of the workstation on which the event occurred.
Application Distinguished Name and Tree	NOTEPAD.APPS.NOVELL.NOVELL_TREE	The distinguished name and tree of the Application object for which the event occurred. For rogue processes, this field contains ZEN Process Management.
Application GUID	3054A94E-BBFF-4851-9D8E-58973623B728	The global unique identifier assigned to the Application object. The GUID is located on the Application object's Options page (Distribution Options tab). For rogue processes, this field is left blank.
Application Version Number	2	The version number assigned to the Application object. Possible numbers range from 0 to 65535. The version number is located on the Application object's Options page (Distribution Options tab). For rogue processes, this field is left blank.
Error Code Major	Could not launch NOTEPAD.APPS.NOVELL.NOVELL_TREE (using c:\winnt\notepa) (id=123)	For failed events, the error code generated by Application Launcher. For successful events, this field is always 0. For rogue processes, this field is left blank.
Error Code Minor	The filename, directory name, or volume label syntax is incorrect.	Additional error code information. For successful events, this field is always 0. For rogue processes, this field is left blank.
Event String 1	c:\winnt\notepa	Optional event information. For rogue processes, this field contains the process executable path.
Event String 2	no example	Optional event information. For rogue processes, this field contains the Original Filename of the process.
Event String 3	no example	Optional event information. For rogue processes, this field contains the process ID (PID).
Event String 4	no example	Optional event information. For rogue processes, this field contains the parent process ID (parent PID).
Event String 5	no example	Optional event information. For rogue processes, this field contains the event action, either "process ignored" or "process terminated successfully."

Field	Example	Description
Application Flag	524288	Application object bitmask. For rogue processes, this field is always 0.

44.8 Enabling Windows Installer Verbose Logging

When you distribute an application based on an MSI package rather than an AOT/AXT package, Application Launcher launches the Microsoft Windows Installer so that it can install the application according to the information and files in the MSI package. By default, the Windows Installer creates an `msixxxxx.log` file that includes basic information and messages.

If you need to troubleshoot problems with the installation, you can set up verbose logging for the Windows Installer. The Windows Installer creates a log file, `zappmsi.log`, in the user's temporary directory on the workstation. Enabling verbose logging will slow down installation of MSI applications, so you should do it only as a troubleshooting procedure.

To enable verbose logging on a workstation:

- 1 Modify the Windows registry to add the following key:
`HKEY_LOCAL_MACHINE\Software\NetWare\NAL\1.0\Debug`
- 2 Under the Debug key, add a *DWORD* value. Set the value name to *MSI* and the value data to *1*.
- 3 Save the registry.
- 4 Restart the workstation.

You need to modify the registry of each workstation where you want to enable verbose logging. We recommend that you create an Application object whose only function is to modify the registry.

For information about Windows Installer error messages included in the `msixxxxx.log` file or the `zappmsi.log` file, see the [Windows Installer Error Messages document \(http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msi/erro_89f7.asp\)](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msi/erro_89f7.asp) at the Microsoft Developer Network (MSDN) site.

For information about Windows Installer error codes returned through Application Launcher, see the [Error Codes document \(http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msi/code_13ub.asp\)](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msi/code_13ub.asp) at the Microsoft Developer Network site.

Metering Software Licenses

45

Novell® ZENworks® Desktop Management integrates with Novell Licensing Services to enable you to track an application's usage and comply with the application's license agreement.

Novell Licensing Services is a component of Novell NetWare® 6.x and Novell Open Enterprise Server (OES). To use Licensing Services with ZENworks, you must have a ZENworks Server installed on a NetWare 6.x server or an OES NetWare server. Novell Licensing Services also requires the Novell Client. You can download the latest client from the [Novell Product Downloads site \(http://download.novell.com/index.jsp\)](http://download.novell.com/index.jsp).

When a user launches an application that has been configured as part of Novell Licensing Services, Novell Application Launcher™ checks to make sure that a license is available before running the application.

License metering applies only to simple applications, AOT/AXT applications, and MSI applications. Licensing metering is not available for terminal server applications or Web applications.

To set up software metering, complete the tasks in the following sections:

- ♦ [Section 45.1, “Installing Novell Licensing Services,” on page 469](#)
- ♦ [Section 45.2, “Creating Metered Certificates,” on page 469](#)
- ♦ [Section 45.3, “Installing the Novell Client and Licensing Files to Workstations,” on page 471](#)
- ♦ [Section 45.4, “Configuring Applications to Use License Metering,” on page 471](#)

45.1 Installing Novell Licensing Services

Novell Licensing Services must be installed before you can use Application Launcher to meter software licenses. It is installed by default on NetWare 6.x and OES servers. For information about ensuring that Licensing Services is installed and configured correctly, see the documentation for the appropriate NetWare version:

- ♦ [NetWare 6 \(http://www.novell.com/documentation/nw6p/nls502enu/data/htoz9uzc.html\)](http://www.novell.com/documentation/nw6p/nls502enu/data/htoz9uzc.html)
- ♦ [NetWare 6.5 \(http://www.novell.com/documentation/nw65/nlsadmin/data/htoz9uzc.html\)](http://www.novell.com/documentation/nw65/nlsadmin/data/htoz9uzc.html)
- ♦ [OES NetWare \(http://www.novell.com/documentation/oes/nlsadmin/data/htoz9uzc.html\)](http://www.novell.com/documentation/oes/nlsadmin/data/htoz9uzc.html)

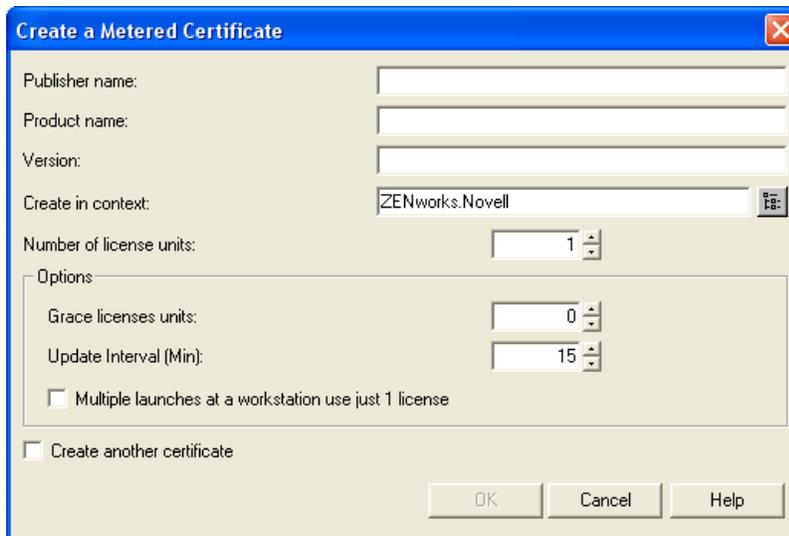
45.2 Creating Metered Certificates

You must create a metered certificate for each application whose usage you want to track. The metered certificate defines the number of licenses you have available for the application.

If necessary, you can have more than one metered certificate for an application. For example, if you initially have 200 licenses for the application, you can create a metered certificate of 200 licenses. Later, if you purchase 100 additional licenses, you can create a second metered certificate of 100 licenses.

You must use NetWare Administrator (NWAdmin), not ConsoleOne or iManager, to create metered certificates. To do so:

- 1 Launch NetWare Administrator from the NetWare 6.x or OES NetWare server (`sys:\public\win32\nwadmin32.exe`).
- 2 Select the container where you want to create the metered certificate, then click the *Tools* menu > *Novell Licensing Services* > *Add Licenses* to display the *Add a License Certificate* dialog box.
- 3 Select the *License Metering* option, then click *OK* to display the *Create a Metered Certificate* dialog box.



- 4 Fill in the following fields:

Publisher Name: Enter the name of the product vendor.

Product Name: Enter the name of the product (for example, Microsoft Office).

Version: Enter the product version.

Create in Context: Verify that this field displays the container where you want the metered certificate created. If not, click the browse button to find and select the correct container.

Grace License Units: Enter the number of units beyond the licensed units that you want to make available for use. This allows additional users to install and run the application while you purchase additional licenses. You can comply with your software license agreement without interrupting users' work flow or productivity.

Update Interval: Enter how often, in minutes, you want Licensing Services to poll the metered certificate to see how many licenses are still available.

Multiple Launches at a Workstation Use Just 1 License: Select this option if only one license should be used even if the application is installed or launched multiple times on the workstation.

- 5 Click *OK* to create the metered license.

NetWare Administrator creates a license container and adds the metered license to the container. The license container is named *publisher+product+version* (for example, Microsoft+Office+2003). You can double-click the license container to view the metered certificate, which is named with the date and time that it was created. You can double-click the metered license to see information such as how many units are in use.

- 6 If you need to add licenses for the product, follow the steps above making sure you do the following:
- ♦ Start by selecting the same container where the product's license container resides.
 - ♦ When you open the *Create a Metered Certificate* dialog box, use the same publisher name, product name, and version that you used for the first metered certificate. This causes the new metered certificate to be created in the same license container as the first certificate.

45.3 Installing the Novell Client and Licensing Files to Workstations

- 1 Make sure that the Novell Client™ is installed on users' workstations.

Novell Licensing Services requires the Novell Client. You can download the latest client from the [Novell Product Downloads site \(http://download.novell.com/index.jsp\)](http://download.novell.com/index.jsp).

- 2 Copy the `nls32.dll` and `nlsapi32.dll` to users' workstations.

- ♦ On Windows 98, copy the files to `c:\windows\system`.
- ♦ On Windows 2000/XP, copy the files to `c:\winnt\system32`.

The Novell Client requires `nls32.dll` for license metering. The `nls32.dll` file is included in the `\license` directory on the *Novell ZENworks 7 Companion 1 CD*.

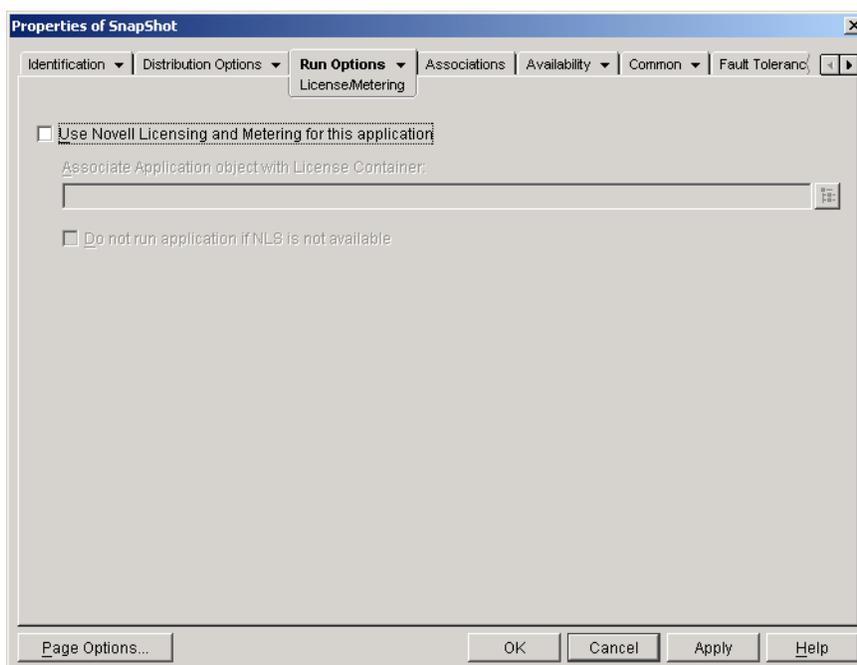
45.4 Configuring Applications to Use License Metering

After you have installed Novell Licensing Services, the Novell Client and its required `nls32.dll`, and have a license container and metered certificate created for an application, you need to configure the Application object to use Novell Licensing Services. This enables Application Launcher to enforce the licensing you've established for the application.

To enable Application Launcher to enforce an application's licensing:

- 1 In ConsoleOne®, right-click the Application object, then click *Properties*.

- 2 Click the *Run Options* tab > *License/Metering* to display the License/Metering page.



- 3 Select the *Use novell licensing and metering for this application* check box to turn on the option.
- 4 In the *Associate application object with License container* field, click the *Browse* button and select the application's license container.
- 5 If you don't want Application Launcher to run the application when Novell Licensing Services is not available, select the *Do not run if NLS is not available* check box. Otherwise, Application Launcher runs the application.
- 6 Click *OK*.

Reference: AdminStudio ZENworks Edition

46

Novell® ZENworks® 7 includes FLEXnet AdminStudio 7.5 ZENworks Edition to help you create and manage Windows Installer (MSI) software packages. AdminStudio ZENworks Edition includes the following tools:

- ♦ **Repackager:** Lets you repackage and convert Novell ZENworks projects (.axt/.aot) and other legacy setups into Windows Installer packages (.msi).
- ♦ **Tuner:** Lets you customize MSI packages. For example, you can create custom transforms that limit the features installed, add custom files to the installation, and set property values. You can also make multiple customizations from the same base setup.
- ♦ **Distribution Wizard:** Lets you deploy your MSI package to the source locations from which users can install the MSI application. It also lets you create an Application object for the MSI application in Novell eDirectory™.
- ♦ [Section 46.1, “Installing AdminStudio ZENworks Edition,” on page 473](#)
- ♦ [Section 46.2, “Using AdminStudio ZENworks Edition,” on page 474](#)

46.1 Installing AdminStudio ZENworks Edition

AdminStudio ZENworks Edition is included on the *Novell ZENworks 7 Software Packaging CD*.

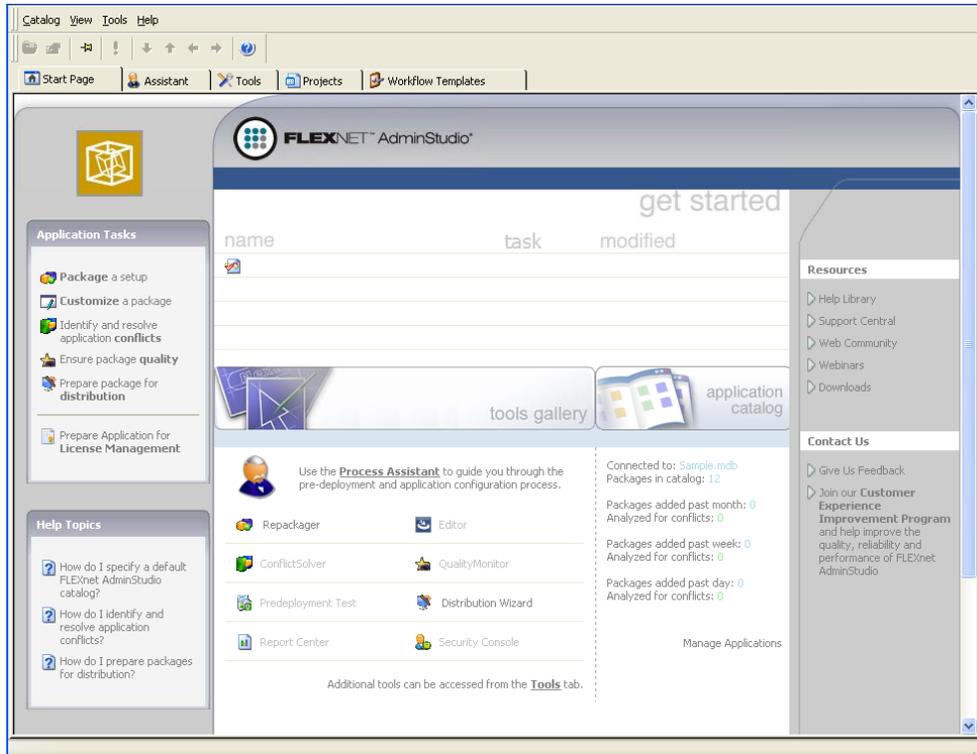
To install AdminStudio ZENworks Edition:

- 1 Make sure the Windows workstation where you want to install AdminStudio ZENworks Edition meets the system requirements. See the [AdminStudio ZENworks Edition Installation Guide](http://www.novell.com/documentation/zenworks7/pdfdoc/spinstall/AS8ZENWorksInstallGuide.pdf) (<http://www.novell.com/documentation/zenworks7/pdfdoc/spinstall/AS8ZENWorksInstallGuide.pdf>) for details.
- 2 At the Windows workstation, insert the *Novell ZENworks 7 Software Packaging CD* to display the ZENworks 7 installation program.

The `winsetup.exe` program automatically runs. If it does not run, launch the program from the root of the CD.
- 3 On the installation program’s main page, click *Software Packaging* to display the Software Packaging page.
- 4 Click *AdminStudio ZENworks Edition* to launch the AdminStudio installation program.
- 5 Follow the prompts to complete the installation, making sure you complete the registration process so that you receive your software key.

46.2 Using AdminStudio ZENworks Edition

- 1 On the Windows workstation where you installed AdminStudio ZENworks Edition, click the *Start* menu > *Programs* > *Macrovision* > *FLEXnet AdminStudio 7.5* to display the following screen.



The Tools Gallery displays the various tools provided by AdminStudio. The Repackager, Tuner, and Distribution Wizard are available in the ZENworks Edition. Only the Repackager and Distribution Wizard are displayed in the Tools Gallery; the Tuner is available on the Tools tab.

The tools that are dimmed are available in the professional or enterprise edition of AdminStudio. For a comparison of editions and information about purchasing the professional or enterprise edition, see the [FLEXnet AdminStudio ZENworks Edition site \(http://www.macrovision.com/products/flexnet_adminstudio/adminstudio/editions/zenworks.shtml\)](http://www.macrovision.com/products/flexnet_adminstudio/adminstudio/editions/zenworks.shtml).

- 2 For instructions about how to use AdminStudio, click the *Help* menu, then click *Contents* to display the Help system.

When distributing applications to Windows 98 or Windows 2000/XP workstations, Novell® ZENworks® Desktop Management supports two types of software packages: Microsoft Windows Installer (MSI) packages and ZENworks snAppShot™ (AOT/AXT) packages.

Based on the benefits of using the native Windows Installer, Novell recommends using Windows Installer packages rather than snAppShot packages. For more details regarding this recommendation, see [Novell snAppShot versus FLEXnet AdminStudio's Repackager \(http://www.macrovision.com/company/news/newsletter/tips/novell_vs_adminstudio.shtml\)](http://www.macrovision.com/company/news/newsletter/tips/novell_vs_adminstudio.shtml).

If you decide to use ZENworks snAppShot, the following sections provide information about snAppShot and instructions for using the utility.

- ♦ [Section 47.1, “Understanding SnAppShot,” on page 475](#)
- ♦ [Section 47.2, “Preparing a SnAppShot Workstation,” on page 477](#)
- ♦ [Section 47.3, “Creating an Installation Package,” on page 477](#)
- ♦ [Section 47.4, “Command Line Switches,” on page 478](#)

47.1 Understanding SnAppShot

SnAppshot records the changes that occur on a workstation as an application is installed. As the installation proceeds, snAppShot captures the differences between the workstation's pre-installation configuration state and the workstation's post-installation state, compares the two pictures, and creates an application installation package that consists of two Application object template (.aot or .axt) files, one or more application source (.fil) files, and one file definition (filedef.txt) file.

47.1.1 Application Object Templates

You use the Application object template file to create the Application object in Novell eDirectory™. Both Application object template (.aot and .axt) files contain the same information, which is used to populate the Application object property fields during creation of the object:

- ♦ The eDirectory name and workstation shortcut name to be given the Application object.
- ♦ The modifications that need to be made to the workstation's configuration settings (registry settings, INI settings, text file modifications, and so forth) during installation of the application.
- ♦ The macro definitions to use during installation.
- ♦ The list of application files to copy to the workstation during installation, including the source location to copy the files from and the target location to copy the files to.

The .aot file is a binary file that cannot be edited; the .axt file is a text file that can be modified with a text editor. If you need to modify the Application object template after snAppShot has created it, you should modify the .axt file and use it to create the Application object. Otherwise, you should use the .aot file because data from the .aot file is imported more quickly.

47.1.2 Application Source Files

SnAppShot also tracks all of the application files that are copied to the workstation. These files, which become the application source files, are copied to a network source location, renamed numerically starting with 1, and given a `.fil` file extension (for example, `1.fil`). Novell Application Launcher™ uses these source files when installing the application to the workstation.

47.1.3 Application File Definition File

To map the `.fil` files to their original files, snAppShot creates a file definition (`filedef.txt`) file. This text file not only maps the `.fil` files to the original files, but also specifies the target location and name to be used when installing the files to the workstation. For example:

```
1.fil=c:\dmi\win32\_deisl1.isu
2.fil=c:\dmi\win32\bin\wdmiutil.dll
```

47.1.4 SnAppShot Limitations

Please be aware of the following before using snAppShot:

- ◆ SnAppShot does not work with MSI applications installed by Microsoft Windows Installer. This includes applications such as Microsoft Office 2000 and Microsoft Office XP.

MSI applications often install portions of the application “on-demand,” which means that a snapshot of an MSI application's installation could possibly not include all features and functionality available in the application. For this reason, MSI applications must be distributed as MSI applications rather than as AOT/AXT (snAppShot) applications. For more information, see [Chapter 29, “Distribution: Complex Applications,” on page 331](#).

- ◆ SnAppShot does not support some of the new Windows XP registry types, such as hex(80000007). Care should be taken when using snAppShot on Windows XP to ensure that the generated `.axt/.aot` file (or Registry page in the resulting Application object) does not include changes that could adversely affect the operation of Windows XP workstations to which the application is distributed.
- ◆ SnAppShot fails when the Windows registry is larger than 30 MB. The following message displays:

```
A critical 'out of memory' error has occurred. snAppShot must close.
```

The recommended solution is to use AdminStudio ZENworks Edition instead. For information, see the [AdminStudio ZENworks Edition Installation Guide \(http://www.novell.com/documentation/zenworks65/pdfdoc/spinstall/AS_ZENworksInstallGuide.pdf\)](http://www.novell.com/documentation/zenworks65/pdfdoc/spinstall/AS_ZENworksInstallGuide.pdf).

- ◆ SnAppShot does not support Windows terminal servers. Do not use snAppShot on a terminal server to create an AOT/AXT application you want to deliver to other terminal servers.
- ◆ The Application object template changed from version 4.0.1 to 6.5 to accommodate several functionality changes, including the change from system requirements to distribution rules. If you have a pre-ZENworks 6.5 environment, you must select the Create Pre 6.5 Application Object Template File option while creating the `.aot` file in order to use the file in your environment. If you use the default option, Create 6.5 Application Object Template File, the resulting `.aot` file is not usable in a pre-ZENworks 6.5 environment.

47.2 Preparing a SnAppShot Workstation

Before running snAppShot on a workstation to create an installation package for an application, you should:

- ♦ Make sure the workstation is clean. A clean workstation has only the operating system and Microsoft Client.
- ♦ Make sure the workstation is representative of the type of workstation to which the application will be distributed. For example, if you are distributing the application to Windows 2000 users on a Dell* OptiPlex* GX110, run snAppShot on a Dell OptiPlex GX110 running Windows 2000. Depending on the application and what occurs during an install, it might be necessary to create different Application objects to be used for different types of workstations.

47.3 Creating an Installation Package

To create an installation package, snAppShot uses the following process:

- ♦ Takes an image of the workstation before the application is installed. You determine which drives are scanned and which configuration settings (registry settings, INI settings, and so forth) are recorded.
- ♦ Has you install the application.
- ♦ Takes another image of the workstation, records the differences between the two images, and creates the installation package (.aot and .axt files, .fil files, and filedef.txt file) in the network location you've specified.

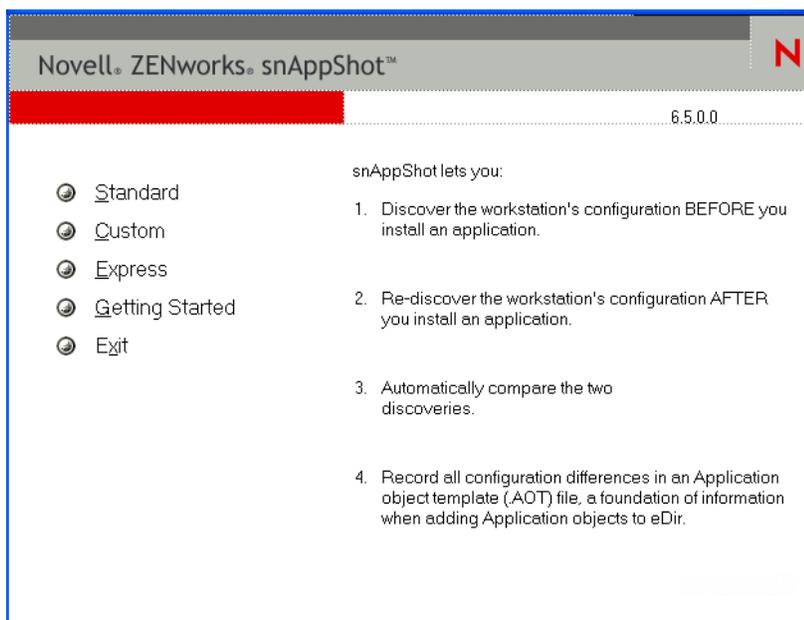
To run snAppShot and create an installation package:

- 1 On the clean and representative workstation (see [Section 47.2, “Preparing a SnAppShot Workstation,” on page 477](#)), start snAppShot (snapshot.exe) from the ZENworks Desktop Management Server.

The location of snapshot.exe depends on the server operating system:

- ♦ NetWare: sys:\public\snapshot
- ♦ Windows: c:\novell\public\snapshot
- ♦ Linux: /opt/novell/zenworks/zdm/winutils/snapshot

For information about switches you can use when starting snAppShot, see [Section 47.4, “Command Line Switches,”](#) on page 478.



2 Select the mode you want to use:

Standard: Uses the snAppShot default preference file during the discovery process. In most cases, these preferences should be sufficient.

Custom: Allows you to choose a previously created preference file to apply to the discovery process or to use the snAppShot default preference file. Unlike the Standard mode, Custom mode lets you make changes to specific preferences to discover drives, files, folders, registry settings, and shortcuts.

Express: Lets you choose a previously created preference file. No modifications can be made to the preferences.

3 Follow the on-screen prompts to create the installation package. If you need more information than is provided on the screen, click the *Help* button.

47.4 Command Line Switches

SnAppShot includes two switches that you can use on the command line. The syntax is:

```
snapshot switch
```

/u:filename.ini

This switch enables you specify the file from which snAppShot retrieves preference settings. You must have already created the preferences file during a previous snAppShot session. Using this switch is similar to running snAppShot, selecting the Express option, and then selecting the preferences file.

If the file is not in the same directory as snAppShot, specify the full path to the file.

/slow

By default, the snAppShot discovery process is optimized for a single-byte operating system. If you are running snAppShot on a double-byte operating system, you should use the */slow* switch. This causes snAppShot to use a string comparison routine optimized for double-byte characters. As a result, SnAppShot runs slower.

Reference: Application Object Settings

48

An Application object includes many settings (properties) you can modify to manage the application. The following sections correspond to each tab on the Application object property page.

- ◆ [Section 48.1, “Identification Tab,” on page 481](#)
- ◆ [Section 48.2, “Distribution Options Tab,” on page 489](#)
- ◆ [Section 48.3, “Run Options Tab,” on page 520](#)
- ◆ [Section 48.4, “Associations Tab,” on page 534](#)
- ◆ [Section 48.5, “Availability Tab,” on page 537](#)
- ◆ [Section 48.6, “Common Tab,” on page 557](#)
- ◆ [Section 48.7, “MSI Tab,” on page 577](#)
- ◆ [Section 48.8, “Terminal Server Client Tab,” on page 583](#)
- ◆ [Section 48.9, “Fault Tolerance Tab,” on page 585](#)

48.1 Identification Tab

The Identification tab includes the following pages to help you configure how the Application object is displayed to users:

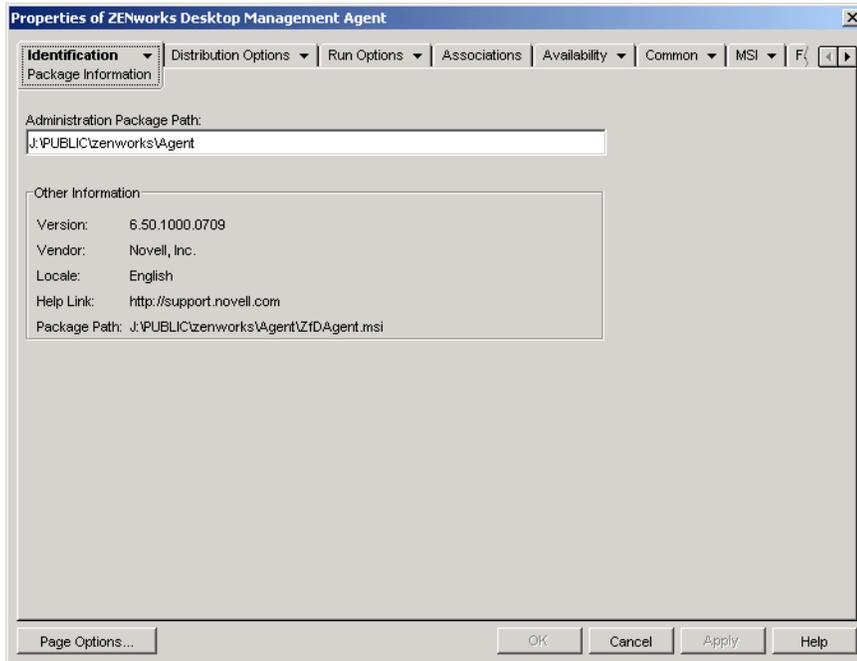
- ◆ [Section 48.1.1, “Package Information Page,” on page 481](#)
- ◆ [Section 48.1.2, “Icon Page,” on page 482](#)
- ◆ [Section 48.1.3, “Description Page,” on page 485](#)
- ◆ [Section 48.1.4, “Folders Page,” on page 485](#)
- ◆ [Section 48.1.5, “Contacts Page,” on page 487](#)
- ◆ [Section 48.1.6, “Administrator Notes Page,” on page 488](#)

48.1.1 Package Information Page

The Package Information property page is available on Application objects created for MSI applications only. It is not available on Application objects created for simple applications, AOT/AXT applications, Web applications, and terminal server applications.

The Package Information property page, shown below, displays information about the Microsoft Windows Installer package file (.msi file) associated with the application. This page is for informational purposes only; you cannot use this page to modify the package’s information.

Figure 48-1 *Application Object > Identification Tab > Package Information Page*



Package path

Displays the location of the .msi file being used by the Application object.

Version

Displays the version of the .msi file.

Vendor

Displays the creator of the .msi file.

Locale

Displays the locale defined in the .msi file.

Help Link

Displays the Web location to visit for information and help about the application.

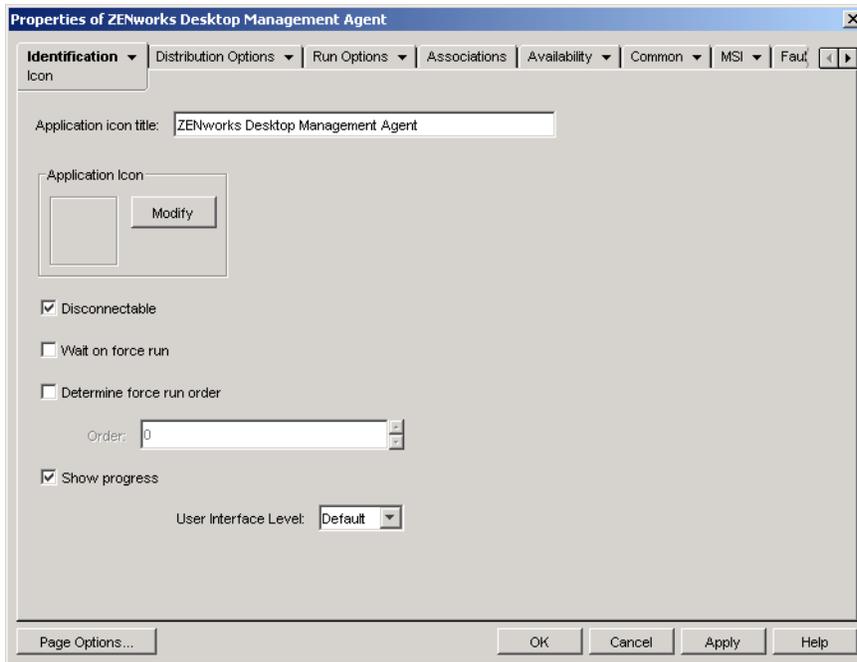
48.1.2 Icon Page

The Icon property page is available on Application objects created for all application types (simple, AOT/AXT, MSI, Web, and terminal server).

The Icon property page, shown below, determines the Application object's icon that Novell® Application Launcher™ displays on a workstation. You can give the application icon a title, select the graphic to use for the icon, and give the icon a display order and force run priority. You can

determine whether or not Application Launcher should continue to display the Application object on the workstation when the user has disconnected from Novell eDirectory™.

Figure 48-2 *Application Object > Identification Tab > Icon Page*



Application Icon Title

Specify the text you want to use as the icon title for the Application object. The icon title must conform to standard Windows folder and filename conventions. If you use the following invalid Windows folder and filename characters, they are replaced on the user's desktop with an underscore (_):

\ / : * ? " < > |

Application Icon

Select the icon you want to appear wherever the Application object's icon is displayed. If you do not specify an icon, a default Application object icon is used.

Disconnectable

Select this option to mark the application as being able to be run on a workstation that is disconnected from eDirectory.

The application must be installed or cached to the workstation before the user can run it in disconnected mode. You can force the application to be installed on the workstation by selecting the *Force run* characteristic (*Associations* page). You can force the application to be cached to the workstation by selecting the *Force cache* characteristic (*Associations* page).

Wait on Force Run

This option applies only if the application and at least one other application are using the *Force run* option (*Associations* page).

Select this option to force the application to wait until the application that started before it terminates. The application order is defined in the *Determine force run order* field. Reboots are queued until the final application has terminated.

Determine Force Run Order

This option dictates the order in which applications set as *Force run* are started. Select this option to enable it, and then use the Order list to select the application's position in the *Force run* order.

You control the order in which the application is started by entering a numeric value in the *Order* box. A value of zero gives the application the highest priority. The maximum value is 9999999. For example, if you want this application to start after two other applications that have been given the order of 0 and 1, you would enter 2 in the *Order* box.

Application Launcher runs the application without waiting for the previously started application to terminate, unless the *Wait on force run* option is enabled.

For Application Launcher to enforce the run order for a group of applications, all of the applications must already be distributed or cached to the workstation; otherwise, if all applications have not yet been distributed or cached to the workstation, the run order is not enforced. For example, if the second application of three that you want to force run has not yet been distributed or cached to the workstation, the first application is started, the distribution process is started for the second application, and then the third application is started. Because of the uncertainty in time required to distribute the second application, the third application might be available before distribution and launching of the second application is completed.

Show Progress

This option displays a progress bar to users when an application is distributed to or removed from their workstations. Turn off this option if you are distributing only a small change, such as a registry modification. Turn on this option if you are distributing or removing a large application and want to give the user an idea of how long the process takes.

With this option turned off, if the workstation must be rebooted to complete the installation process and the Prompt for Reboot setting is enabled (*Distribution Options* tab > *Options* page), the user is not prompted and the workstation automatically reboots. The same is true if the workstation must be rebooted to complete the uninstall process and the Prompt User Before Uninstall setting is enabled (*Common* tab > *Uninstall* page).

User Interface Level

This option appears only if the Application object uses a Microsoft Windows Installer (MSI) package. During distribution of an MSI Application object, Application Launcher launches Windows Installer to install the application. As a result, rather than showing the standard Application Launcher installation progress bar, Windows Installer uses the installation user interface established for the MSI package. You can use the following settings to determine the amount of user interface the Windows Installer displays during installation.

- ◆ Default: Displays an appropriate user interface level (as chosen by Windows Installer).
- ◆ Silent: Displays no user interface.
- ◆ Progress: Displays simple progress information and error messages/prompts.
- ◆ Reduced: Displays a full user interface with Wizard dialog boxes suppressed.

- ◆ Full: Displays a full user interface (Wizard dialog boxes, progress information, error messages and prompts, and so forth).

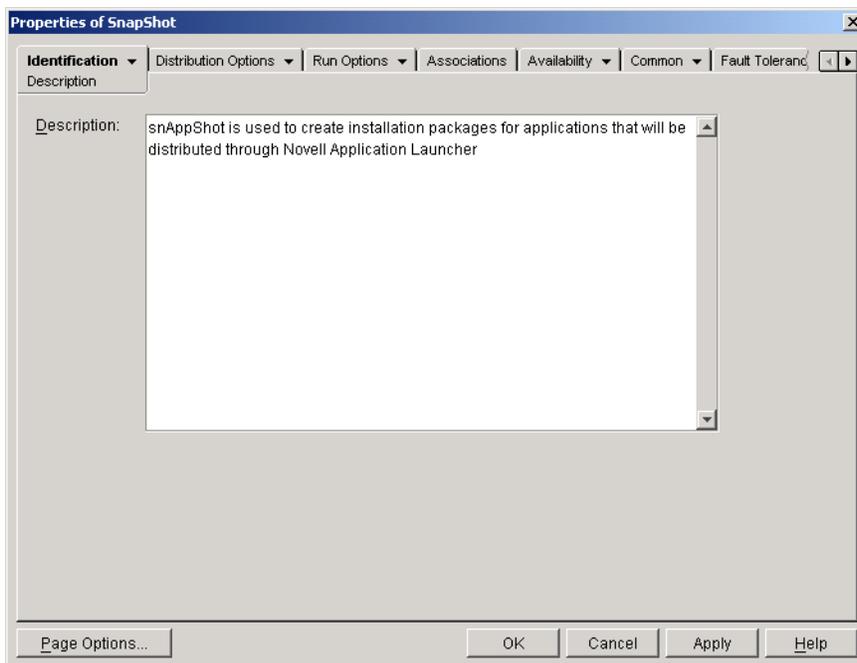
Application Launcher passes the selected setting to Windows Installer as a startup parameter. For more information about these settings, see the Microsoft Windows Installer documentation.

48.1.3 Description Page

The Description property page is available on Application objects created for all application types (simple, AOT/AXT, MSI, Web, and terminal server).

The Description property page, shown below, provides users more complete information about the Application object than the application icon title allows.

Figure 48-3 *Application Object > Identification Tab > Description Page*



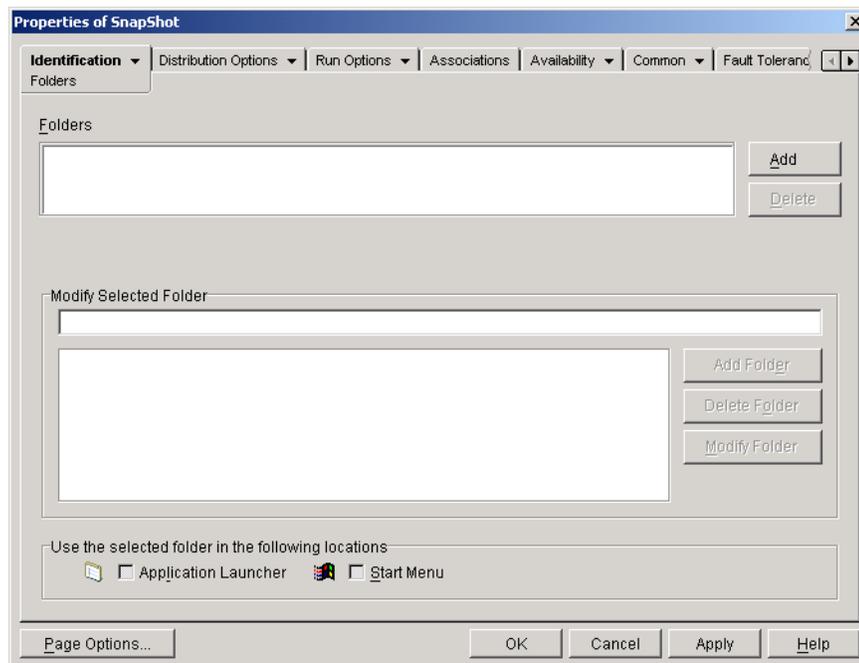
If you have enabled the Prompt User Before Distribution option (*Distribution Options > Options* page), users see this description when Application Launcher first distributes the application to them. They can also view an Application object's properties to see the description. To display the properties, right-click the Application object (on the workstation), then click *Properties*.

48.1.4 Folders Page

The Folders property page is available on Application objects created for all application types (simple, AOT/AXT, MSI, Web, and terminal server).

The Folders property page, shown below, lets you specify the folders where you want Application Launcher to place the application when distributing it to a workstation.

Figure 48-4 Application Object > Identification Tab > Folders Page



You can add the application to two different types of folders: a custom folder and a linked folder.

A custom folder is a folder created specifically for the Application object. No other Application objects can be included in the folder. Custom folders support folders within folders, which means that you can create a custom folder structure. For example, although you could not have Calculator and Notepad in the same custom folder, you could create two subfolders within the same custom folder and place the two programs in the two subfolders (in other words, `winapps\calculator\calc.exe` and `winapps\notepad\notepad.exe`).

A linked folder is simply an association to an existing Application Folder object. The Application Folder object must already exist in eDirectory. If the Application Folder object includes multiple folders (a folder structure) you can add the application to any folder in the structure.

If you plan to create a complex folder structure for the applications you distribute, we recommend that you use an Application Folder object and then link Application objects to the Application Folder object. An Application Folder object requires you to define the folder structure one time only, whereas custom folders must be defined for each Application object. If you choose to use custom folders for your folder structure, make sure you use the same folder names when defining the custom folder structure for each Application object. Any variation causes Application Launcher to create different custom folder structures.

Folders

The Folders list displays the custom folders and linked folders (Application Folder objects) that the application has been added to. Application Launcher creates (if necessary) the listed folders when distributing the application to the workstation.

Add

Click *Add* to add a custom folder or a linked folder to the Folders list. If you want the application to appear in multiple folders, add each folder to the list.

When you add a custom folder, it appears in the list with New Folder as its title. You can select the folder in the list and use the *Modify selected folder* box to change the folder's name or to add subfolders to it (see [Modify Selected Folder](#) below).

Delete

Select a folder in the folder list, then click *Delete* to remove the folder from the list. The change is applied the next time Application Launcher restarts.

Modify Selected Folder

The Modify Selected Folder box lets you modify name and structure information for the folder that is selected in the Folders list. You can modify information for custom folders and linked folders. However, after you modify a linked folder's information, it is converted to a custom folder.

Add Folder

Select the folder in the folder tree, then click *Add Folder* to add a subfolder to the folder. After you add a subfolder and Application Launcher refreshes, users see the application in the subfolder rather than in the folder.

Delete Folder

Select the folder in the folder tree, then click *Delete Folder* to delete the folder. The change is applied the next time Application Launcher restarts.

Modify Folder

Select the folder in the folder tree, then click *Modify Folder* to change the name of the folder.

If you want to add the application to a new subfolder of an existing folder, enter `existing_folder_name\new_subfolder_name`.

Use the Selected Folder in the Following Locations

Application Launcher can display folders on the Windows Start menu and in the Application Launcher windows (Application Window and Application Explorer window), if these locations are enabled on the Application object's Associations page. Select a folder in the Folders list, then select the check boxes for the locations where you want to use the folder.

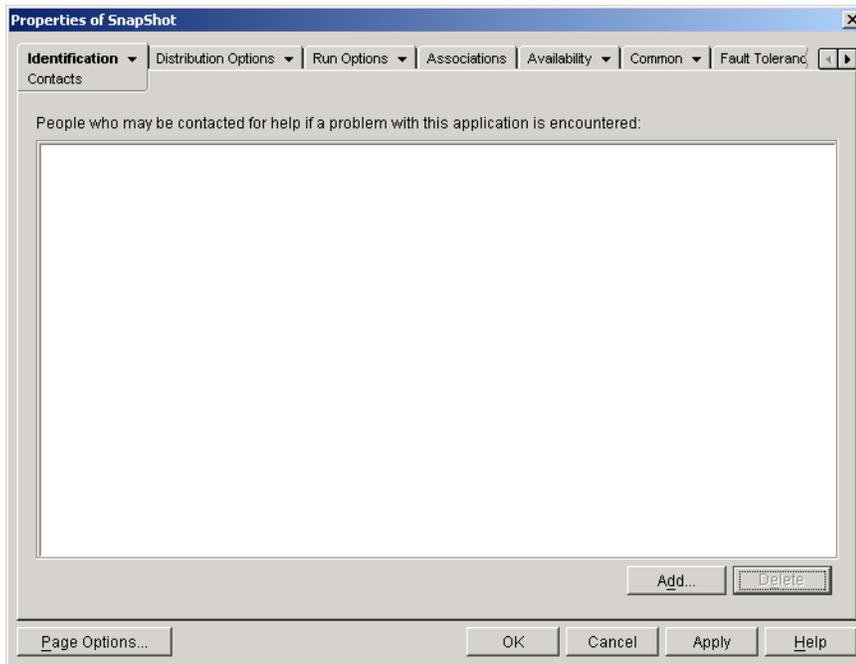
If you don't select either location, Application Launcher still displays the Application object on the Start menu and in the Application Launcher windows, but the object does not appear in the folders you've defined.

48.1.5 Contacts Page

The Contacts property page is available on Application objects created for all application types (simple, AOT/AXT, MSI, Web, and terminal server).

The Contacts property page, shown below, lists the names, e-mail addresses, and phone numbers of the application's support staff. Users can access this information through the Application object's properties. To display the properties, right-click the Application object (on the workstation), click *Properties*, then click *Help contacts*.

Figure 48-5 *Application Object > Identification Tab > Contacts Page*



You can tailor the information to direct users to the support staff at their locations. If you enter the support contact's e-mail address, users can send an e-mail message directly from the *Help Contacts* page of the Properties dialog box.

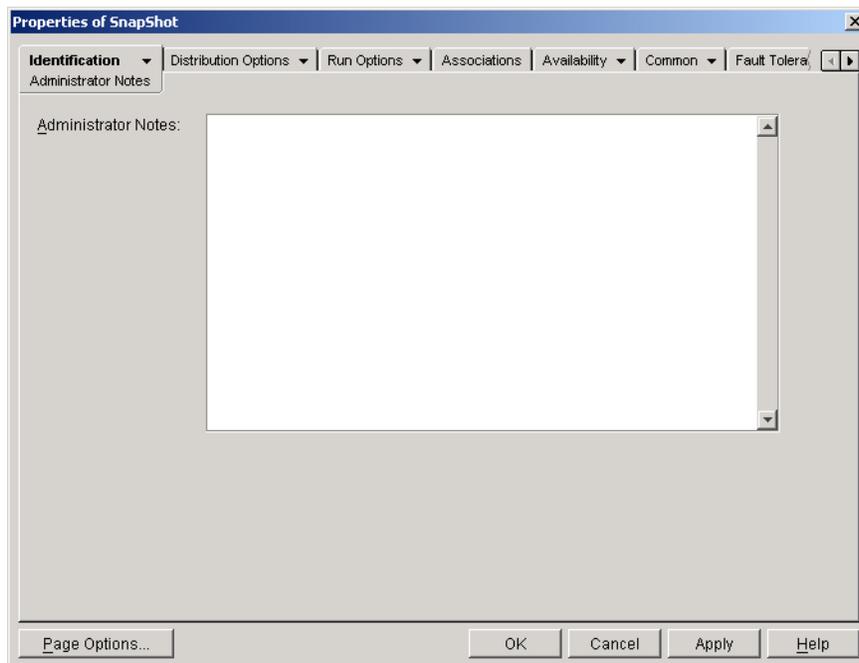
NOTE: Users must have the eDirectory rights required to read the *E-mail address* field (*Internet email address* attribute) and *Telephone* field (*Telephone number* attribute) of the users defined as contacts.

48.1.6 Administrator Notes Page

The Administrator Notes property page is available on Application objects created for all application types (simple, AOT/AXT, MSI, Web, and terminal server).

The Administrator Notes property page, shown below, lets you record notes for yourself or other administrators. For example, you could remind yourself about special settings for an application. Or, if you have several administrators, you could write a history of upgrades and file changes.

Figure 48-6 Application Object > Identification Tab > Administrator Notes Page



48.2 Distribution Options Tab

The Distribution Options tab includes the following pages to help you configure how the Application object is distributed to users:

- ◆ [Section 48.2.1, “Icons/Shortcuts Page,” on page 489](#)
- ◆ [Section 48.2.2, “Registry Page,” on page 492](#)
- ◆ [Section 48.2.3, “Application Files Page,” on page 495](#)
- ◆ [Section 48.2.4, “INI Settings Page,” on page 499](#)
- ◆ [Section 48.2.5, “Text Files Page,” on page 503](#)
- ◆ [Section 48.2.6, “Distribution Scripts Page,” on page 506](#)
- ◆ [Section 48.2.7, “Pre-Install Schedule Page,” on page 509](#)
- ◆ [Section 48.2.8, “Pre-Distribution Process Termination Page,” on page 513](#)
- ◆ [Section 48.2.9, “Options Page,” on page 514](#)
- ◆ [Section 48.2.10, “BITS Settings Page,” on page 518](#)

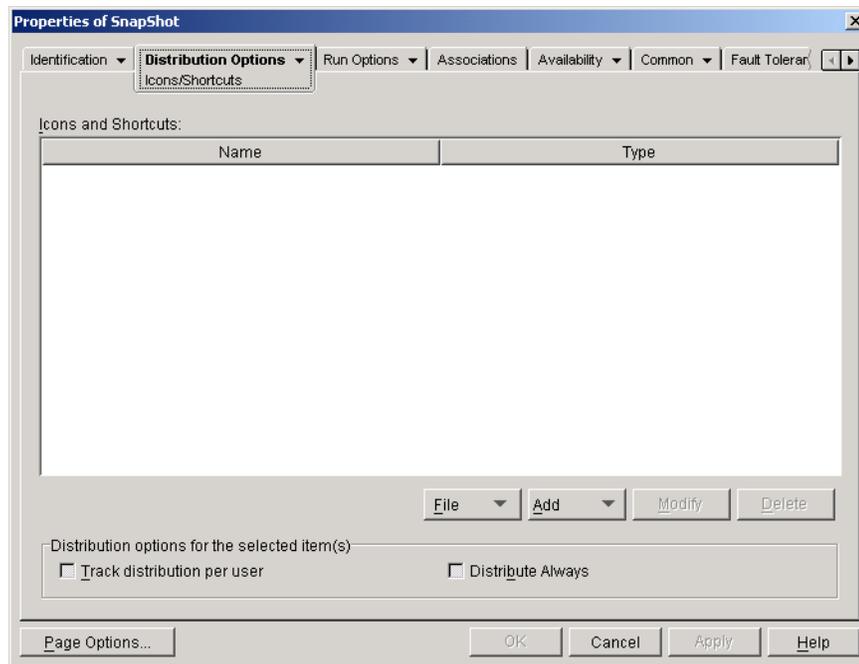
48.2.1 Icons/Shortcuts Page

The Icons/Shortcuts property page is available only on Application objects created for simple applications and AOT/AXT applications. It is not available on Application objects created for MSI applications, Web applications, and terminal server applications.

The Icons/Shortcuts property page, shown below, determines the icons and shortcuts that Application Launcher creates when distributing the application to the workstation. You can add the

application's icon as an item in a program group or as a shortcut on the workstation's desktop or in a folder. You can also delete existing icons, shortcuts, and program groups.

Figure 48-7 Application Object > Distribution Options Tab > Icons/shortcuts Page



The icons and shortcuts you add with this page are in addition to the Application object's icon. Although the Application object's icon might cause various actions to occur, including installing the application or running it, the icons and shortcuts defined on this page link directly to the application's executable file and simply launch the application.

You can use icons and shortcuts in combination with other options to create the user environment you want. For example, you could define the icons and shortcuts you want to be created and configure the Application object to run one time (*Run Options > Applications*). When a user selects the Application object, Application Launcher runs the application one time, creates the icons and shortcuts, performs any other tasks specified by the Application object's properties, and then removes the Application object's icon from the workstation. Thereafter, the user needs to select the icon or shortcut to launch the application.

IMPORTANT: If Application Launcher cannot create a shortcut, the application is not distributed. In this case, all application files and settings are removed. However, if other shortcuts were created before the shortcut that failed, those shortcuts are not removed.

Icons and Shortcuts

This list displays the icons and shortcuts that are created when the application is distributed to a workstation.

Files

Click *File > Find* to search for icon and shortcut definitions that include certain information.

Click *File* > *Import* to import icons and shortcuts from another Application object. The Open dialog box defaults to *.axt for its file type display. If you are importing from an .aot file, you must change the file type display to *.aot or *All files* in order to select the .aot file.

Add

Click *Add* to add a new program group, program group item, or shortcut. Program groups and program group items are supported on Windows 98 workstations but not on Windows 2000/XP workstations. Shortcuts are supported on all Windows versions.

IMPORTANT: When defining the target path for a shortcut, if the application will be distributed to a Windows 2000/XP workstation you must use a UNC path rather than a mapped drive path. Long mapped drive paths are truncated on Windows 2000/XP, resulting in an invalid shortcut that does not work.

Modify

Select an icon or shortcut from the Icons and Shortcuts list, then click *Modify* to change the information associated with it.

Delete

Select an icon or shortcut from the Icons and Shortcuts list, then click *Delete* to delete it from the list.

Track Distribution Per User

If you have implemented roaming user profiles, use this option to ensure that particular icons and shortcuts are distributed to each workstation a user logs in to.

In the Icons and Shortcuts list, select the desired icon or shortcut, then select *Track distribution per user*.

Distribute Always

By default, Application Launcher creates the icons and shortcuts defined in the Icons and Shortcuts list only at the following times:

- ◆ The first time the application is launched on a workstation.
- ◆ The first time the application is launched after the application's version number (*Distribution Options* tab > *Options* page) has been changed.

To force Application Launcher to create an icon or shortcut each time the application is launched, select the icon or shortcut in the Icons and Shortcuts list, then select *Distribute always*.

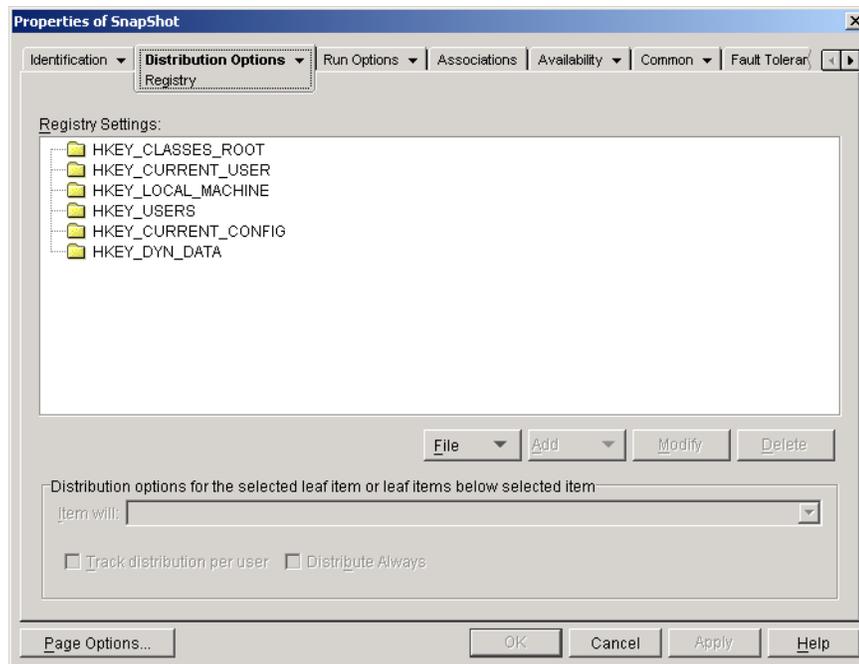
If the user has a NAL cache directory on his or her local machine, Application Launcher uses the information stored in the NAL cache directory to create the icon or shortcut. If the user does not have a NAL cache directory (for example, the user is running Application Launcher through a terminal server client session) or if writing to the cache has been disabled for the user (*User object* > *ZENworks* tab > *Launcher Configuration* page > *Enable writing to cache* option), Application Launcher uses the information stored in eDirectory.

48.2.2 Registry Page

The Registry property page is available only on Application objects created for simple applications, AOT/AXT applications, and MSI applications. It is not available on Application objects created for Web applications and terminal server applications.

The Registry property page, shown below, determines the registry modifications that Application Launcher makes when distributing the application to a workstation.

Figure 48-8 Application Object > Distribution Options Tab > Registry Page



Registry Settings

The Registry Settings tree displays all settings that are modified when the application is distributed to a workstation. If you used an `.aot`, `.axt`, or `.msi` file when creating the Application object, the tree automatically includes all registry settings that are defined in those templates.

If there are additional registry settings you want created or deleted during distribution, you need to add the settings to the *Registry settings* tree and then specify the appropriate action (create or delete) in the *Item will* field.

NOTE: For Application objects created for AOT/AXT applications, the Novell Application Launcher (NAL) handles the distribution of the registry settings and the distribution of the application. If you modify registry settings for an AOT/AXT application and the registry settings fail to distribute, the application itself fails and NAL rolls back the application's installation.

For Application objects created for MSI applications, NAL handles the distribution of registry settings and the Microsoft Windows Installer (MSI) handles distribution of the application. If you modify the Application object's registry settings for a MSI application and the registry settings fail to distribute, the application is installed by Windows Installer, but the registry settings do not roll

back. As a result, the application might not function properly, depending on how the registry settings affect the application.

File

This option lets you search for keys or values in the *Registry settings* tree, import settings into the tree, and export settings from the tree.

Click *File*, then choose one of the following options:

- ◆ **Find:** Searches for specific keys, value names, or value data in the registry.
- ◆ **Find next:** Finds the next occurrence of the key, value name, or value data previously searched for.
- ◆ **Import:** Imports registry settings from another Application object's *.aot* or *.axt* file, or from a registry file (*.reg*). The Open dialog box defaults to **.axt* for its file type display. If you are importing from an *.aot* file or *.reg* file, you must change the file type display to **.aot*, **.reg*, or *All files* in order to select the appropriate file.
- ◆ **Export:** Exports the registry settings to a registry file (*.reg*). To export the settings to an *.aot* or *.axt* file format, you must export the entire Application object using the Export Application Object option located on the *Tools > ZENworks Utilities > Application Launcher Tools* menu.

Add

This option lets you add registry settings to the *Registry settings* tree. Only settings displayed in the Registry Settings tree are created or deleted when the application is distributed.

To add a registry key or value, select the registry folder where you want to add the key, or select the key where you want to add a value, click the Add button, then choose one of the following options:

- ◆ **Key:** Adds a key to the selected registry folder.
- ◆ **Binary:** Adds a binary value to the selected key.
- ◆ **Expand string:** Adds an expand string value to the selected key. The expand string setting does not exist in the Windows 98 registry; if you use this setting, it is changed to a string setting during distribution to Windows 98 workstations.
- ◆ **Default:** Adds a default string value to the selected key.
- ◆ **DWORD:** Adds a DWORD value to the selected key.
- ◆ **Multi-string value:** Adds a multi-value string to the selected key. The multi-value string setting does not exist in the Windows 98 registry; if you use this setting, it is changed to a binary setting during distribution to Windows 98 workstations.
- ◆ **String:** Adds a string value to the selected key.

After the key or value is added to the Registry Settings tree, you can use the Distribution Options list to determine whether or not the key or value is created in the workstation's registry or deleted from the registry.

You can use a macro for a key name, value name, or value data. For information about macros, see [Chapter 49, "Reference: Macros," on page 593](#).

Modify

Select the key or value you want to modify, then click *Modify*.

Delete

Select the key or value you want to delete, then click *Delete*. When you delete a key, everything subordinate to the key is also deleted.

Distribution Options

The distribution options let you determine how individual registry settings (included in the Registry Settings tree) are handled during distribution of the application.

Item Will

In the Registry Settings tree, select the setting, then select the action that you want to occur for the setting when the application is distributed:

- ◆ **Create always:** The setting is always created in the registry, even if it already exists. If it exists, the setting's current values are overwritten. For example, if `PATH=C:\` already exists, `PATH=C:\TEMP` replaces it.
- ◆ **Create if does not exist:** The setting is created only if it does not already exist.
- ◆ **Create if exists:** The setting is created only if it already exists. The setting's current values are overwritten. For example, if `PATH=C:\` already exists, `PATH=C:\TEMP` replaces it.
- ◆ **Delete:** The setting is deleted. If the registry setting has subordinate settings, Application Launcher deletes the subordinate settings also.
- ◆ **Append if exists otherwise create:** This option applies only to string values (*String*, *Default string*, *Expanded Sstring*, and *Multi-value string*). The string value's data is added to the existing string as the last entry. If the string value (or its key) does not exist, it is created.

When specifying the string value, you must include a semicolon (;) delimiter before the value if the string has existing values. For example, assume that the registry already includes a `string1=value1` setting. You want to append a second value (`value2`). When specifying the string value, you must specify `;value2` so that the resulting string is `string1=value1;value2`.

- ◆ **Prepend if exists otherwise create:** This option applies only to string values (*String*, *Default string*, *Expanded string*, and *Multi-value string*). The string value's data is added to the existing string as the first entry. If the string value (or its key) does not exist, it is created.

When specifying the string value, you must include a semicolon (;) delimiter after the value if the string has existing values. For example, assume that the registry already includes a `string1=value1` setting. You want to prepend a second value (`value2`). When specifying the string value, you must specify `value2;` so that the resulting string is `string1=value2;value1`.

Track Distribution Per User

If you have implemented roaming user profiles, use this option to ensure that particular registry settings are distributed to each workstation a user logs in to. You should enable this option for all registry settings that are not saved as part of roaming user profiles.

In the Registry Settings list, select the desired registry modification, then select the Track Distribution Per User check box.

Distribute Always

By default, Application Launcher distributes the registry modifications defined in the Registry Settings list only at the following times:

- ◆ The first time the application is launched on a workstation.
- ◆ The first time the application is launched after the application's version number (*Distribution Options* tab > *Options* page) has been changed.

To force Application Launcher to distribute a registry modification each time the application is launched, select the registry setting in the *Registry settings* list, then select the *Distribute always* check box.

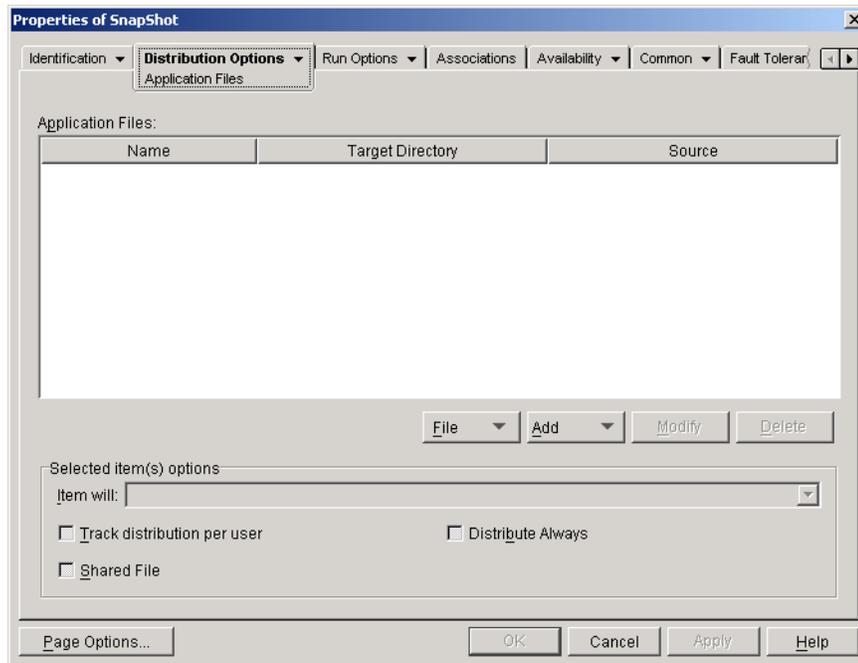
If the user has a NAL cache directory on his or her local machine, Application Launcher uses the setting information stored in the NAL cache directory to modify the registry. If the user does not have a NAL cache directory (for example, the user is running Application Launcher through a terminal server client session) or if writing to the cache has been disabled for the user (User object > *ZENworks* tab > *Launcher Configuration* page > *Enable writing to cache* option), Application Launcher uses the setting information stored in eDirectory.

48.2.3 Application Files Page

The Application Files property page is available only on Application objects created for simple applications and AOT/AXT applications. It is not available on Application objects created for MSI applications, Web applications, and terminal server applications.

The Application Files property page, shown below, specifies the application files that Application Launcher installs or removes when distributing the application to a workstation.

Figure 48-9 Application Object > Distribution Options Tab > Application Files Page



Application Files

The Application Files list displays all files and directories that are installed, removed or copied during distribution. The name, target directory (the location on the workstation where the file is installed), and source (the file or directory that is being used to install the file) are listed for each application file or directory.

If you used an `.aot` or `.axt` file when creating the Application object, the list automatically includes all files and directories that are defined in those templates.

File

This option lets you search for items in the Application Files list and import files and directories into the list.

Click *File* > choose one of the following options:

- ◆ **Find:** Searches for specific items in the list. You can search for text in the *Name*, *Target Directory*, or *Source* fields. For example, you could search for all files and directories that are being installed under the `Program Files` directory.
- ◆ **Find next:** Finds the next occurrence of the item specified by the initial search.
- ◆ **Import:** Imports application files and directories from another Application object's `.aot` or `.axt` file. The Open dialog box defaults to `*.axt` for its file type display. If you are importing from an `.aot` file, you must change the file type display to `*.aot` or *All files* in order to select the `.aot` file.

Add

This option lets you add files or directories to the *Application files* list. Only files and directories displayed in the *Application files* list are installed to, removed from, or copied to the workstation during distribution.

- ◆ **File:** To add a file to the list, click *Add*, then click *File* to display the Edit Files dialog box.

In the *Source file* field, specify the file to be used as the source of the installation. You can use a mapped drive, UNC path, or macro, or you can browse and select the file (for example, \\server1\vol1\bookmarks\bookmark.htm or %SOURCE_PATH%\bookmark.htm). The source file can be a single file that is copied, multiple files (for example, %SOURCE_PATH%*.*), or a snAppShot™ application source (.fil) file.

If you are deleting a file from the workstation, leave this field blank.

In the *Target file* field, specify the file on the workstation where the source file is to be copied (for example, c:\program files\novell\browser\bookmark.htm). You can also substitute a macro for the target path (for example, %TARGET_PATH%\bookmark.htm). If you are copying multiple files using a wildcard (*.*), specify the destination directory only (for example, c:\program files\novell\browser\).

If you are deleting the file from a workstation, enter the full path for the file from the perspective of the workstation, then select the *Target file to be deleted* check box.

- ◆ **Directory:** To add a directory to the list, click *Add*, then click *Directory* to display the Edit Directory dialog box.

In the *Source directory* field, specify the directory to be used as the source, if you are copying the directory. You can use a mapped drive, UNC path, or macro, or you can browse and select the directory (for example, c:\program files\novell). You can also substitute a macro for the directory path (for example, %DIRECTORY_TARGET_PATH%\novell).

If you are creating or deleting a directory, the *Source directory* field is disabled.

In the *Target directory* field, specify the directory to create or delete, or you can specify the directory on the workstation where the source directory is to be copied (for example, c:\program files\novell). You can also substitute a macro for the directory path (for example, %DIRECTORY_TARGET_PATH%\novell).

Select *Create directory* to create the directory on the workstation.

Select *Delete directory* to delete the directory from the workstation.

Select *Copy directory* to copy the directory to the workstation. When you select *Copy directory*, the *Include subdirectories* option becomes available. Click the *Include subdirectories* check box if you want to copy the subdirectories of the directory listed in the *Source directory* field.

Modify

Select the file or folder you want to modify, then click *Modify*.

Delete

Select the file or folder you want to delete, then click *Delete* to remove it from the *Application files* list.

NOTE: If there are files and subfolders in the folder to be deleted, those files and subfolders must be deleted before the parent folder can be deleted.

Selected Item(s) Options

Use these options to set individual distribution options for files and directories included in the *Application files* list.

Item Will

Select a file in the *Application files* list > select one of the following options from the *Item will* list:

- ◆ **Copy always:** Copies the file regardless of whether the file currently exists on the workstation.
- ◆ **Copy if exists:** Copies the file only if the file currently exists on the workstation.
- ◆ **Copy if does not exist:** Copies the file only if the file does not currently exist on the workstation.
- ◆ **Copy if newer:** Copies the file only if its date and time are newer than the existing file's date and time, or if the file does not currently exist on the workstation.
- ◆ **Copy if newer and exists:** Copies the file only if it already exists on the workstation and has an older date or time.
- ◆ **Copy if newer version:** Copies the file only if its internal version is newer than the existing file's version (if version information is present). This is useful if you want to update the version of an `.exe` or `.dll` file based on the compiled version information.
- ◆ **Request confirmation:** Prompts the user to verify that the file should be copied.
- ◆ **Copy if different:** Copies the file if its date, time, or size is different than the existing file's date, time, or size.
- ◆ **Delete:** Deletes the file from the workstation.

Select a folder in the *Application files* list, then select one of the following options from the *Item will* list:

- ◆ **Create:** Creates the directory on the workstation.
- ◆ **Delete:** Deletes the directory from the workstation.

Track Distribution Per User

If you have implemented roaming user profiles, use this option to ensure that application files are distributed to each workstation a user logs in to. You should enable this option for all application files that are not saved as part of roaming user profiles.

In the *Application files* list, select the desired application files, then select *Track distribution per user*.

Distribute Always

By default, Application Launcher distributes the file and folder modifications defined in the *Application files* list only at the following times:

- ◆ The first time the application is launched on a workstation.

- ◆ The first time the application is launched after the application's version number (*Distribution Options* tab > *Options* page) has been changed.

To force Application Launcher to distribute a file or folder modification each time the application is launched, select the file or folder in the *Application files* list, then select *Distribute always*.

If the user has a NAL cache directory on his or her local machine, Application Launcher uses the information stored in the NAL cache directory to install or remove the file or folder. If the user does not have a NAL cache directory (for example, the user is running Application Launcher through a terminal server client session) or if writing to the cache has been disabled for the user (User object > *ZENworks* tab > *Launcher Configuration* page > *Enable writing to cache* option), Application Launcher uses the information stored in eDirectory.

Shared File

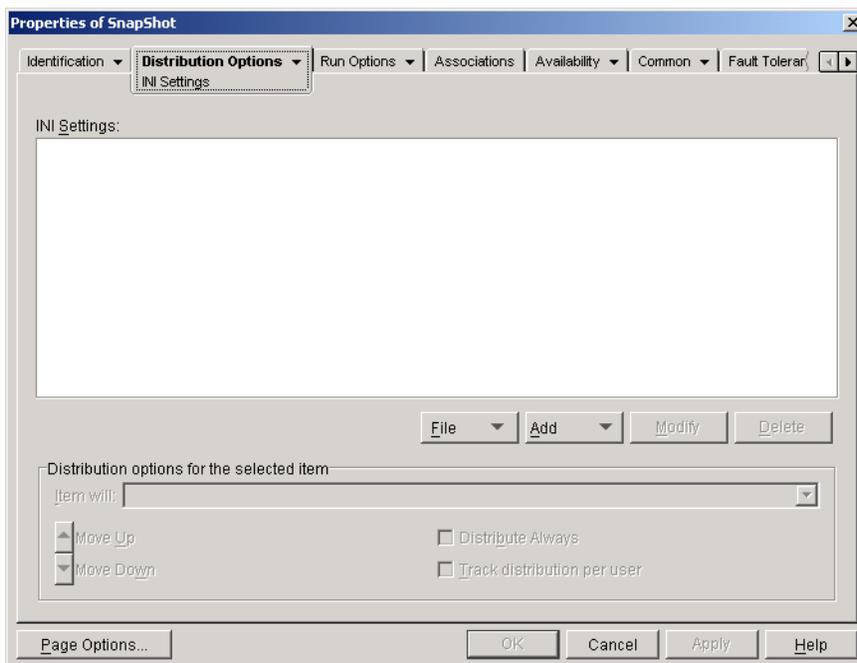
Use this option to mark a file as a shared file (that is, one that is used by more than one application). Shared files are usually Windows DLL files. SnAppShot detects shared files when it discovers application installation changes on a workstation.

48.2.4 INI Settings Page

The INI Settings property page is available only on Application objects created for simple applications, AOT/AXT applications, and MSI applications. It is not available on Application objects created for Web applications and terminal server applications.

The INI Settings property page, shown below, determines the INI settings that Application Launcher creates or deletes when distributing the application to a workstation.

Figure 48-10 INI Settings Page



INI Settings

The INI Settings tree displays the INI settings that are modified when the application is distributed to the workstation. If you used an `.aot`, `.axt`, or `.msi` file when creating the Application object, the tree automatically includes all INI settings defined in those templates.

If there are additional INI settings you want created or deleted during distribution, you need to add the settings to the *INI settings* tree and then specify the appropriate action (create or delete) in the *Item will* field.

The INI Settings tree can include multiple INI files and each file can include multiple sections. When you add a setting to the INI Settings tree, you must add it to a file and section. This means that you might need to add new files and new sections to the tree before you can add new settings.

For example, assume that you want to add a `CLASSPATH=` setting to the `ENVIRONMENT` section of the `sample.ini` file. You would 1) add a file entry to the *INI settings* tree for the `sample.ini` file, 2) add an `ENVIRONMENT` section under the `sample.ini` file, 3) add the `CLASSPATH=` setting under the `ENVIRONMENT` section, and 4) select the `CLASSPATH=` setting and choose the appropriate *Create* action in the *Item will* field.

If, instead of adding the `CLASSPATH=` setting, you wanted to delete it, you would follow the same process but then choose the appropriate *Delete* action in the *Item will* field.

NOTE: For Application objects created for AOT/AXT applications, the Novell Application Launcher (NAL) handles the distribution of the INI settings and the distribution of the application. If you modify INI settings for an AOT/AXT application and the INI settings fail to distribute, the application itself fails and NAL rolls back the application's installation.

For Application objects created for MSI applications, NAL handles the distribution of INI settings and the Microsoft Windows Installer (MSI) handles distribution of the application. If you modify the Application object's INI settings for a MSI application and the INI settings fail to distribute, the application is installed by Windows Installer, but the INI settings do not roll back. As a result, the application might not function properly, depending on how the INI settings affect the application.

File

This option lets you search for files, sections, or values in the INI Settings tree, import settings into the tree, export settings from the tree, or view a file's INI settings.

Click *File* > choose one of the following options:

- ◆ **Find:** Searches for specific files, sections, or values.
- ◆ **Find next:** Finds the next occurrence of the item specified by the initial search.
- ◆ **Import:** Imports INI settings from another Application object's `.aot` or `.axt` file, or from a `.ini` file. The Open dialog box defaults to `*.axt` for its file type display. If you are importing from an `.aot` file or `.ini` file, you must change the file type display to `*.aot`, `*.ini`, or *All files* in order to select the appropriate file.
- ◆ **Export:** Exports the settings to an `.ini` file. To export the settings to an `.aot` or `.axt` file, you must export the entire Application object using the *Export application object* option located on the *Tools > ZENworks Utilities > Application Launcher Tools* menu.
- ◆ **View file:** Displays the INI settings for a specific file that are modified when the application is distributed. You must select the file from the INI Settings tree before you click *File > View File*.

Add

This option lets you add INI settings to the INI Settings tree. Only settings displayed in the INI Settings tree are created or deleted when the application is distributed. You can add a file to the tree, a section to a file, or a value to a section.

To do so, select the appropriate item in the tree, click the *Add* button, then choose one of the following options:

- ◆ **File:** Adds a file to the INI Settings tree. In addition to providing a file name, you can specify the target location for the file. By default, the `%*WINDIR%` macro is used, which represents the workstation's Windows directory (typically `c:\windows` or `c:\winnt`). After you name the file, you can begin adding sections to it.
- ◆ **Section:** Adds a section to the selected file. After you name the section, you can begin adding values to it.
- ◆ **Value:** Adds a value to the selected section. You must specify the value name and value data.

After you've added a value to the INI Settings tree, you can use the *Distribution options* list to determine whether or not it is created or deleted from the workstation. If the value needs to be created but the file or section does not exist, Application Launcher creates the file or section before adding the value.

You can use a macro for a section name, value name, or value data. For more information about macros, see [Chapter 49, "Reference: Macros," on page 593](#).

Modify

You can modify a file's name, a section's name, or a value's name and data. Select the file, section, or value you want to modify, then click *Modify*.

Delete

Select the file, section, or value you want to delete from the *INI settings* tree, then click *Delete*. When deleting a file or section, everything subordinate to it is also deleted.

Distribution Options for the Selected Item

The distribution options let you determine how individual INI settings (included in the INI Settings tree) are handled during distribution of the application.

Item Will

Use this option to determine whether a setting is created or deleted when the application is distributed. Select a value in the *INI settings* tree, then select one of the following options from the *Item will* list:

- ◆ **Create always (default):** Creates the value regardless of whether the value currently exists in the section.
- ◆ **Create if does not exist:** Creates the value only if the value does not currently exist in the section.
- ◆ **Create if exists:** Creates the value only if the value currently exists in the section.
- ◆ **Create or add to existing section:** Creates the value if the value does not currently exist in the section. If the value exists, it adds this value to the section in addition to the one that already

exists. This is useful, for example, if you need multiple values of the same type, such as two "DEVICE=" values.

- ♦ **Create or append to existing value:** Creates the value if the value does not currently exist in the section. If the value exists, it appends the data for the new value to the existing value. The first character in the value data needs to be the separator character, such as a space.
- ♦ **Delete:** Deletes the value from the section.
- ♦ **Delete or remove from existing value:** Deletes the value from the section, or, if the value has multiple data entries, removes this value's data entry from the value. For example, suppose the following setting is in the `win.ini` file: `Run = sol.exe calc.exe`. Using this option, you can remove just `calc.exe`, leaving the following: `Run = sol.exe`. The first character in the value data needs to be a separator character, such as a space.

Move Up / Move Down

Use these options to position sections and values in the order in which you want them to be created, modified, or deleted.

In the *INI settings* tree, select the section or value to move, then click *Move Up* or *Move Down*.

Distribute Always

By default, Application Launcher distributes the modifications defined in the INI Settings list only at the following times:

- ♦ The first time the application is launched on a workstation.
- ♦ The first time the application is launched after the application's version number (*Distribution Options* tab > *Options* page) has been changed.

If the user has a NAL cache directory on his or her local machine, Application Launcher uses the information stored in the NAL cache directory to make the INI modification. If the user does not have a NAL cache directory (for example, the user is running Application Launcher through a terminal server client session) or if writing to the cache has been disabled for the user (User object > *ZENworks* tab > *Launcher Configuration* page > *Enable writing to cache* option), Application Launcher uses the information stored in eDirectory.

To force Application Launcher to distribute an INI modification each time the application is launched, select the INI setting in the *INI settings* list, then select *Distribute always*.

NOTE: After you save the Application object and open it again, settings that you mark as *Distribute always* are grouped after settings that are not marked *Distribute always*, regardless of their creation order or their forced order (by using the *Move Up* and *Move Down* arrows).

For example, if you have a section with two *Distribute always* values (DAValue1 and DAValue2) and two non-Distribute Always value (Value3 and Value 4), the values are listed in the following order: Value3, Value4, DAValue1, DAValue2.

You can use the *Move Up* and *Move Down* arrows to change the order within the two groupings, but the *Distribute always* group is always listed second. For example, using the previous ordering (Value3, Value4, DAValue1, DAValue 2), you could change the order of the first two values and the order of the second two values to get the following order: Value4, Value3, DAValue2, DAValue1. However, if you change the order to list the Distribute Always values first (DAValue2, DAValue1,

Value4, Value3), when you save the Application object the order reverts to Value4, Value3, DAValue2, DAValue1.

Track Distribution Per User

If you have implemented roaming user profiles, use this option to ensure that particular `.ini` file settings are distributed to each workstation a user logs in to. You should enable this option for all `.ini` file settings that are not saved as part of roaming user profiles.

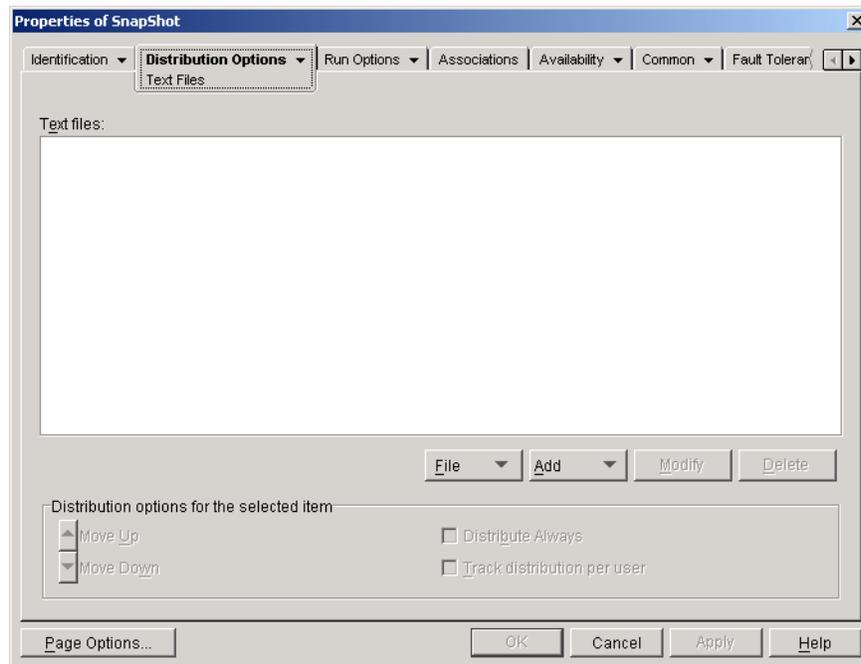
In the *INI settings* tree, select the setting you want to track, then select *Track distribution per user*.

48.2.5 Text Files Page

The Text Files property page is available only on Application objects created for simple applications and AOT/AXT applications. It is not available on Application objects created for MSI applications, Web applications, and terminal server applications.

The Text Files property page, shown below, determines the modifications that Application Launcher makes to text files (such as `config.sys` and `autoexec.bat`) when distributing the application to a workstation.

Figure 48-11 Application Object > Distribution Options Tab > Text Files Page



Text Files

The Text Files tree shows the text files that Application Launcher modifies. Each modification to a file is displayed subordinate to the file.

File

This option lets you search for files or text in the *Text files* tree and import files into the tree.

Click *File*, then choose one of the following options:

- ◆ **Find:** Searches for specific files or information in the *Text files* tree.
- ◆ **Find next:** Finds the next occurrence of the item specified by the initial search.
- ◆ **Import:** Imports text files from another Application object's `.aot` or `.axt` file. The Open dialog box defaults to `*.axt` for its file type display. If you are importing from an `.aot` file, you must change the file type display to `*.aot` or *All files* in order to select the `.aot` file.

Add

This option lets you add text file modifications to the *Text files* tree. Only the modifications displayed in the *Text files* tree are made when the application is distributed.

- ◆ **File:** To add a text file to the tree, click *Add > File* to create the file entry. You can type the filename or the path and filename (for example, `autoexec.bat` or `c:\autoexec.bat`). Only local workstation drives, UNC server paths, and macros are valid.

You should specify a path if possible. If you enter only the filename, Application Launcher searches all directories specified in the workstation's PATH environment variable. If it does not find a matching filename, it assumes the file doesn't exist and creates it in the first directory specified in the PATH variable.

- ◆ **Change:** To add a change to a file that is in the Text Files list, select the file, click *Add > Change* to display the Edit Text File dialog box. Make the desired changes. Click *Help* in the Edit Text File dialog box for information about each of the dialog box fields.

You can add multiple modifications to a text file. For example, you might want to make one modification that replaces text in the file and another modification that adds text to the end of the file. Each modification you add is displayed beneath the text file in the *Text files* list.

IMPORTANT: If you make changes to a text file (adding text, for example), you can add only one line at a time. If you press Enter to create a line break, your change saves.

Modify

To change the name of a text file, select the file in the *Text files* tree, click *Modify*, then specify the new name.

To edit one of the text file's modifications, select the modification in the *Text files* tree, click *Modify* to display the Edit Text File dialog box, then make the desired changes. Click *Help* in the Edit Text File dialog box for information about each of the dialog box fields.

Delete

In the *Text file* tree, select the text file or text file modification you want to delete > click *Delete*.

Distribution Options for the Selected Item

Use these options to set individual distribution options for text files and text file modifications. The options change depending on whether you have selected a text file or a text file modification in the Text Files tree.

No Reboot Needed If Modified

This option appears only when you have selected a text file. Select this option if you don't want users to reboot after you make changes to the selected text file. The *Reboot* and *Prompt reboot* options on the *Distribution Options > Options* page override this setting.

Move Up / Move Down

These options appear only when you have selected a text file modification. Click *Move Up* or *Move Down* to position the modification according to the order in which you want it applied.

Distribute Always

By default, Application Launcher distributes the text file modifications defined in the *Text files* list only at the following times:

- ◆ The first time the application is launched on a workstation.
- ◆ The first time the application is launched after the application's version number (*Distribution Options* tab > *Options* page) has been changed.

If the user has a NAL cache directory on his or her local machine, Application Launcher uses the information stored in the NAL cache directory to make the modification. If the user does not have a NAL cache directory (for example, the user is running Application Launcher through a terminal server client session) or if writing to the cache has been disabled for the user (User object > *ZENworks* tab > *Launcher Configuration* page > *Enable writing to cache* option), Application Launcher uses the information stored in eDirectory.

To force Application Launcher to distribute a text file modification each time the application is launched, select the modification in the *Text files* list, then select *Distribute always*.

NOTE: After you save the Application object and open it again, modifications that you mark as *Distribute always* are grouped after modifications that are not marked *Distribute always*, regardless of their creation order or their forced order (by using the *Move Up* and *Move Down* arrows).

For example, if you have a file with two *Distribute always* modifications (DAMod1 and DAMod2) and two non-Distribute Always modifications (Mod3 and Mod4), the modifications are listed in the following order: Mod3, Mod4, DAMod1, DAMod2.

You can use the *Move Up* and *Move Down* arrows to change the order within the two groupings, but the *Distribute always* group is always listed second. For example, using the previous ordering (Mod3, Mod4, DAMod1, DAMod 2), you could change the order of the first two modifications and the order of the second two modifications to get the following order: Mod4, Mod3, DAMod2, DAMod1. However, if you change the order to list the *Distribute always* modifications first (DAMod2, DAMod1, Mod4, Mod3), when you save the Application object the order reverts to Mod4, Mod3, DAMod2, DAMod1.

Track Distribution Per User

If you have implemented roaming user profiles, use this option to ensure that particular text file modifications are distributed to each workstation a user logs in to. You should enable this option for all modifications that are not saved as part of roaming user profiles.

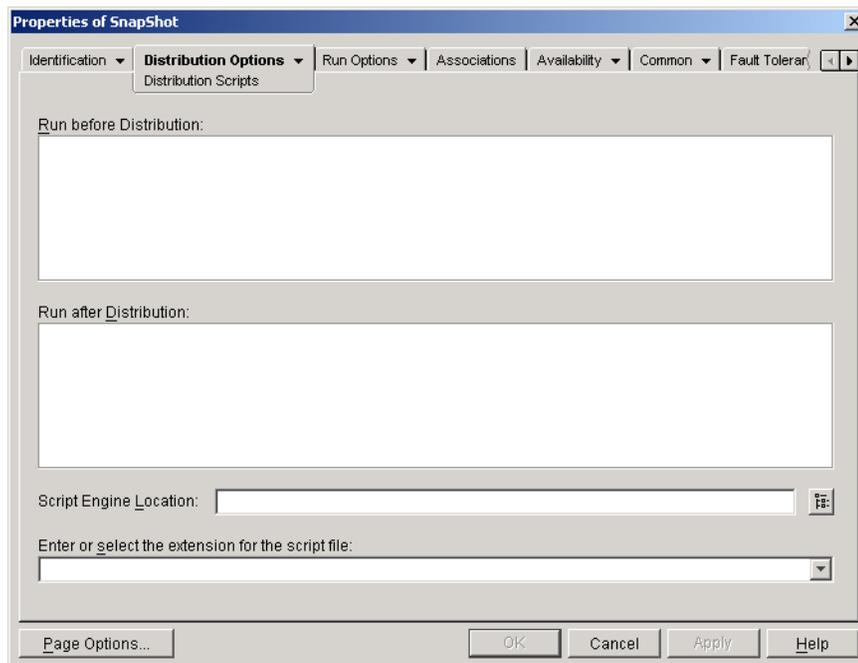
In the *Text files* list, select the desired modification, then select *Track distribution per user*.

48.2.6 Distribution Scripts Page

The Distribution Scripts property page is available on Application objects created for simple applications, AOT/AXT applications, and MSI applications only. It is not available on Application objects created for Web applications and terminal server applications.

As part of the process of distributing an application, Application Launcher can launch a script engine to execute a “before distribution” script and an “after distribution” script (for details about the order of script execution, see [“Script Execution Order” on page 508](#)). The Distribution Scripts property page, shown below, defines the script engine that you want Application Launcher to use and the scripts you want executed.

Figure 48-12 Application Object > Distribution Options Tab > Distribution Scripts Page



On Windows 2000/XP, distribution scripts are run in the secure system space, which means that users do not see any of the script commands or command results. Therefore, you should not include any commands that require or initiate user interaction. If you do so, the script halts at that point. For example, you would not want to include a command to run a program that requires user interaction because the program, which runs in secure system space, is never displayed to the user. On Windows 98, distribution scripts are run in the user space (because there is no system space).

This section includes information about the following items available on the Distribution Scripts page:

- ◆ [“Run Before Distribution” on page 507](#)
- ◆ [“Run After Distribution” on page 507](#)
- ◆ [“Script Engine Location” on page 507](#)
- ◆ [“Script File Extension” on page 507](#)
- ◆ [“Script Example” on page 508](#)
- ◆ [“Script Execution Order” on page 508](#)

- ◆ “Supported Novell Client Login Script Commands” on page 509

Run Before Distribution

Use this text window to enter any script commands you want executed before the application is distributed. Do not use extended characters in the script; extended characters are not supported. For script example, see “Script Example” on page 508.

Run After Distribution

Use this text window to enter any script commands you want executed after the application is distributed. Do not use extended characters in the script; extended characters are not supported. For script example, see “Script Example” on page 508.

NOTE: If you configure an application to launch a script using the *Run After Distribution* method, and if that application is associated to a workstation, the distribution script runs every time a new user initially logs in. Distribution items (such as files and registry keys) are not run for the new user if the distribution has already been delivered to the workstation.

For more information, see TID 3591452 (*Run After Distribution Application Script Is Not Working as Expected*) in the [Novell Support Knowledgebase \(http://www.novell.com/support\)](http://www.novell.com/support).

Script Engine Location

The script engine determines the script commands and scripting language you need to use. If you do not define a script engine in the *Script engine location* field, Application Launcher uses the Novell Client™ as the script engine (if the workstation has the Novell Client installed), which means that you can use most Novell Client login script commands (see “Supported Novell Client Login Script Commands” on page 509).

If you want to use a script engine other than the Novell Client, specify the alternate script engine. The script engine must reside in a location that is always available to users, such as their local drives. The script engine can reside on a network server only if users can map a drive to the server (for example, through the Novell Client or the Client for Microsoft Networks). If Application Launcher cannot find the script engine, it displays an error to the user and fails to distribute the application.

If you use the Windows command interpreter as the script engine, you must include the */C* switch, as shown in the following examples:

- ◆ Windows 2000/XP: `%*winsysdir%\cmd.exe /c`
- ◆ Windows 98: `%*windir%\command.com /c`

The `%*winsysdir%` and `%*windir%` variables specify the Windows system directory (for example, `c:\winnt\system32`), and the `/c` switch instructs the command interpreter to execute the script and then stop. If the `/c` switch is not used, the script does not complete.

For a script example, see “Script Example” on page 508.

Script File Extension

This applies only if you specified a script engine in the Script Engine Location field.

When the application is distributed, Application Launcher creates temporary script files for the *Run before distribution* scripts and *Run after distribution* scripts. These files are passed to the script engine, which then executes the script. You need to specify the file extension that the script engine requires for its script files.

For a script example, see “[Script Example](#)” on page 508.

Script Example

The following script uses the Windows 2000/XP command interpreter as the script engine. Before the distribution occurs, a listing of the `c:\` directory is saved to a text file and the `autoexec.bat` file is backed up.

Run Before Distribution Field

```
dir c:\ >c:\1.txt  
copy autoexec.bat autoexec.bak /y
```

Script Engine Location Field

```
cmd.exe /c
```

Enter or Select the Extension of the Script File Field

```
.bat
```

Script Execution Order

Application Launcher can execute up to four different scripts when distributing and launching an application:

- ◆ **Distribution scripts:** Run Before Distribution and Run After Distribution (*Distribution Options* tab > *Distribution Scripts* page)
- ◆ **Launch scripts:** *Run before launching* and *Run after termination* (*Run Options* tab > *Launch Scripts* page)

Application Launcher executes the scripts in the following order:

1. Run Before Launching script executed
2. Run Before Distribution script executed
3. Application distributed (files copied, settings modified, etc.)
4. Run After Distribution script executed
5. Application launched
6. Application closed (by user)
7. Run After Termination script executed

NOTE: Selecting *Distribute Always* for individual items on certain pages of the Distribution Options Tab causes those items to be processed after the post distribution objects, so the order above becomes invalid. The pages where you will see this behavior include:

- ◆ Icons/Shortcuts page
- ◆ Registry page

- ◆ Application Files page
 - ◆ INI Settings page
 - ◆ Text Files page
-

Supported Novell Client Login Script Commands

When using the Novell Client as the script engine, you can use all but the following script commands:

Table 48-1 *Supported Novell Client Login Script Commands*

CLS	INCLUDE	PCOMPATIBLE
DISPLAY	LASTLOGINTIME	SCRIPT_SERVER
EXIT	NO_DEFAULT	SET_TIME
FDISPLAY	NOSWAP	SWAP
IF MEMBER OF	PAUSE	WRITE

Application Launcher does not output anything to the screen or display script errors.

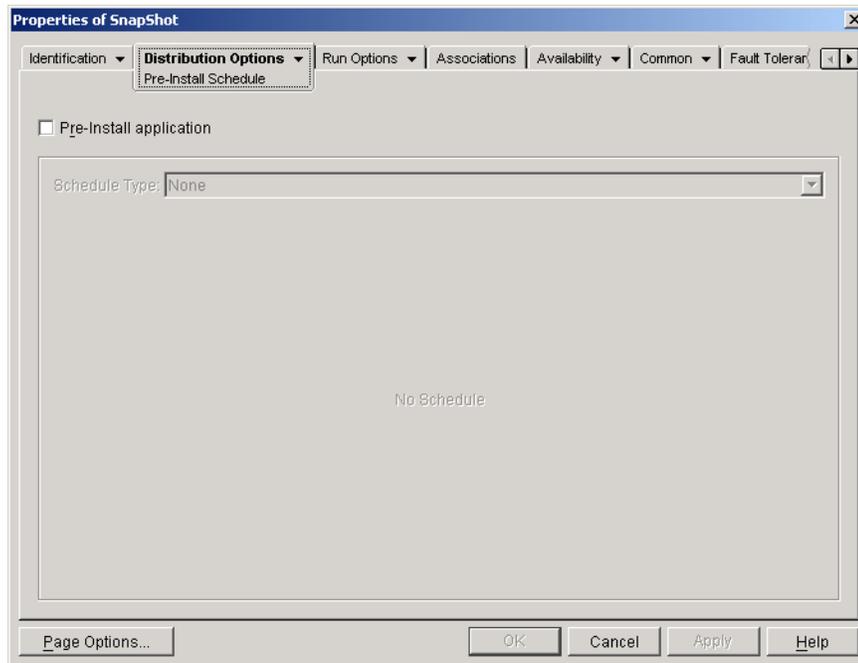
For script commands, syntax, and examples, see the [Novell Client documentation \(http://www.novell.com/documentation/lg/noclienu/docui/index.html\)](http://www.novell.com/documentation/lg/noclienu/docui/index.html) on the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

48.2.7 Pre-Install Schedule Page

The Pre-Install Schedule property page is available only on Application objects created for simple applications, AOT/AXT applications, and MSI applications. It is not available on Application objects created for Web applications and terminal server applications.

The Pre-Install Schedule property page, shown below, enables you to distribute portions of the application to a workstation before the user launches the application the first time. Because you can schedule the distribution, you can perform an off-line, or lights-out, distribution of the application and save the user some of the wait typically associated with a distribution. For example, you could pre-install the application after work hours so the application is ready to use the next day.

Figure 48-13 Application Object > Distribution Options Tab > Pre-install Schedule Page



With a pre-install, all workstation-related distribution processes (file copying, modifying text files, .ini files, and workstation registry settings) are performed prior to launching of the application. When the user launches the application, the user-specific distribution processes (modifying user registry keys and so forth) are completed.

You can pre-install an application that is associated with either workstations or users:

- ◆ For user-associated applications, the user must be logged in and Application Launcher must be running. Application Launcher uses the logged-in user's credentials (authentication and file system access) to distribute the application.
- ◆ For workstation-associated applications, the workstation must be running but Application Launcher does not need to be running. If the application is a non-MSI application (for example, an AOT application), NAL Workstation Helper uses the workstation's credentials to distribute the application. If the application is an MSI application, NAL Workstation Helper uses the logged-in user's credentials. If you want it to use the workstation's credentials rather than require a user to be logged in (for example, to perform a lights-out distribution of the MSI application), you must enable the *Distribute in workstation security space if workstation associated* option (*Distribution Options* tab > *Options* page).

When pre-installing a workstation-associated application, you should also be aware of the following:

- ◆ Any shortcuts, registry settings, application files, INI settings, and text file modifications that are marked as *Track distribution per user* must also be marked as *Distribute always*. Otherwise, they are not distributed. To do so, use the *Track distribution per user* and *Distribute always* options on the **Icons/Shortcuts** page, **Registry** page, **Application Files** page, **INI Settings** page, and **Text Files** page.
- ◆ On Windows 2000/XP workstations, if a user is not logged in, any user-specific macros (*Common* > *Macros* page) point to the default user directories. This scenario affects the ability

to place folders and icons on the Start menu. There are two ways to solve this issue: 1) Mark the macro entries in the Application object as *Track distribution per user* or 2) change the user-specific macro to an All Users macro (such as %*PROGRAMS% to %*COMMONPROGRAMS%).

- ♦ If an application requires a reboot during installation, you must select *Reboot if needed* or *Reboot always* in the Reboot group box and *Don't prompt* in the *Prompt for reboot* group box.

Pre-Install Application

Select this option to enable the application to be pre-installed. If you don't select this option, the application is not pre-installed, even if you establish a schedule.

Schedule Type

Select the type of schedule you want to use. You can choose *None*, *Specified days*, or *Range of days*.

None

Use this option to indicate no schedule. The application is pre-installed as soon as the application is associated with a user or workstation (Associations page).

Specified Days

Use this option to select specific dates when you want to pre-install the application. You cannot select more than 350 specific dates.

- ♦ **Date range:** The Date Range list displays all dates when the application can be pre-installed. To add a date, click *Add*, select the date you want, then click *OK* to display it in the list.
- ♦ **Time for selected dates:** Select the availability start time and end time. The times apply to all dates in the *Date range* list.

NOTE: The time increments in 5 minute intervals, with the earliest available start time being 00:00 (12:00 a.m.) and the latest end time being 23:55 (11:55 p.m.). This means there is always a 5-minute time period from 11:55 p.m. to 12:00 midnight when the application is unavailable. If you want the application to be available the entire day, you need to use the *Range of days* schedule type. For more information, see [“Range of Days” on page 554](#).

- ♦ **Spread from start time (in minutes):** The *Spread from start time* option spreads out user access times over the number of minutes specified so the application doesn't become available to all users at the same time. If you anticipate all users launching the application as soon as it becomes available and the application is being distributed or run from the network, you can use this option to avoid possible network overload.

For example, if you have a moderate number of users to whom the application is to be distributed (say about 100), you might specify a one-hour (60 minute) block of time (starting at the scheduled start time) to randomly distribute the application: thus all users will gain access to the application some time during the first sixty minutes after the scheduled start time.

If you want to substantially ease the load on your servers caused by the application distribution, or if you have bandwidth concerns, you might want to make the application distribute randomly throughout the time of availability. To spread out access times across the entire time (*Specified days* and *Time for selected dates*) that the application is available, use the total availability time specified for that application in terms of minutes. This will require that you make the maximum time available for each day you specify. For example, if an application is configured for a

typical business day in the United States (9 hours per day: 8:00 a.m. to 5:00 p.m.), you calculate the total time of availability for that application like this:

Number of specified hours x 60 minutes per hour = Total availability time per day

Using this equation, the example above would be calculated like this:

9 x 60 (minutes per hour) = 540 minutes of availability

In this example, when you enter 540 minutes in the *Spread from start time* field, the application is distributed randomly for the entire 540 minutes that you have made it available on that scheduled day. Note that this might not be suitable for applications that must be distributed in a timely fashion, such as anti-virus updates. Note also that this is an example only: you can schedule the distribution for any specified amount of time for any day of the week.

Remember that the *Time for selected dates* setting makes the last five minutes of a day unscheduleable, so you need to consider these five minutes if the application schedule ends at 11:55 p.m. for that day.

Range of Days

Use this option to select a range of days to pre-install the application. You can also use this option to pre-install the application only on certain days of the week within a given range of dates.

- ♦ **Date range:** To define the range of days, select a start date and an end date > select the days (Sunday through Saturday) within the established date range. By default, all days are selected; a day is selected when the button appears to be pressed in.
- ♦ **Time for selected range:** Select the availability start time and end time. This option works differently depending on whether the date range includes one day, multiple days, or all seven days. If the date range includes one to six days (but not all seven days), the application is available between the start and end times on those days. For example, if you make the application available on Monday between 8:00 and 5:00, it is available during those hours. However, if the date range includes all seven days, the times are ignored and the application is available every day, 24 hours a day.
- ♦ **Spread from start time (in minutes):** The *Spread from start time* option spreads out user access times over the number of minutes specified so the application doesn't become available to all users at the same time. If you anticipate all users launching the application as soon as it becomes available and the application is being distributed or run from the network, you can use this option to avoid possible network overload.

For example, if you have a moderate number of users to whom the application is to be distributed (say about 100), you might specify a one-hour (60 minute) block of time (starting at the scheduled start time) to randomly distribute the application: thus all users will gain access to the application some time during the first sixty minutes after the scheduled start time.

If you want to ease the load of the application distribution on your servers or if you have bandwidth concerns, you might want to make the application distribute randomly throughout the time of availability. To spread out access times across the entire time (*Date range* and *Time for selected dates*) that the application is available, use the total availability time specified for that application in terms of minutes. For example, if a workstation-associated application is configured for an entire 24-hour, three-shift day, you can calculate the total time of availability for that application like this:

Number of days in date range x *Time of availability per day* = Total availability time

Using this equation, and making sure to convert hours to minutes, the example above would be calculated like this:

7 (days) x 24 (hours) = 168 hours of availability

168 x 60 (minutes per hour) = 10,080 minutes of availability

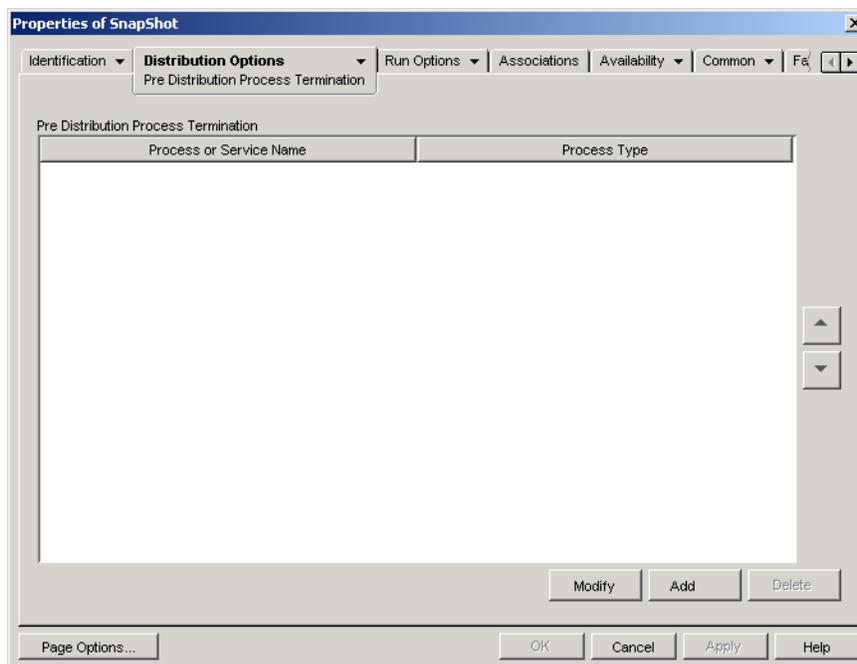
When you enter 10800 minutes in the *Spread from start time* field, the application is distributed randomly for the entire 10800 minutes that you have made it available. Note that this is not suitable for applications that must be distributed in a timely fashion, such as anti-virus updates.

- ♦ **Use this schedule in GMT for all clients:** The schedule is based on the workstation's time zone. If your network spans different time zones and you schedule an application to run at 1:00 p.m., it runs at 1:00 p.m. in each time zone. You can select this option to have workstations run applications at the same time regardless of their time zones (for example, 1:00 p.m. Rome time and 4:00 a.m. Los Angeles time).

48.2.8 Pre-Distribution Process Termination Page

The Pre-Distribution Process Termination property page, shown below, determines the executables and services that Application Launcher terminates before distributing the application to a workstation.

Figure 48-14 Application Object > Distribution Options > Pre-distribution Process Termination Page



Application Launcher can terminate any process running in the user space. In addition, it can terminate any service running in the system space (provided the service is displayed in the Microsoft Management Console's Services list and you use that service name). Application Launcher cannot terminate executables running in the system space.

When terminating a process, Application Launcher terminates all processes that match the specified filename. For example, if you specify `notepad.exe` as the process executable to terminate, all instances of `notepad.exe` terminate. In other words, if both `c:\notepad.exe` and `c:\winnt\notepad.exe` are running, both terminate. You cannot target specific instances of a process (for example, only `c:\notepad.exe` or `c:\winnt\notepad.exe`).

Add

This option lets you add processes to the list. Only processes displayed in the list are terminated before the application is distributed.

Click *Add* to display the Edit Processes dialog box. In the *Process* or *Service name* box, enter the executable (.exe) filename or enter the service name (as defined in the Services list in the Microsoft Management Console). Do not include the full file paths; if you do so, the termination fails.

Click *Service* if the process is a Windows service, then click *OK* to add the process to the *Pre-distribution process termination* list.

Modify

You can modify a process name and type. Select the process in the list, then click *Modify*.

Delete

Select the process you want to delete from the list, then click *Delete*.

Up-Arrow and Down-Arrow

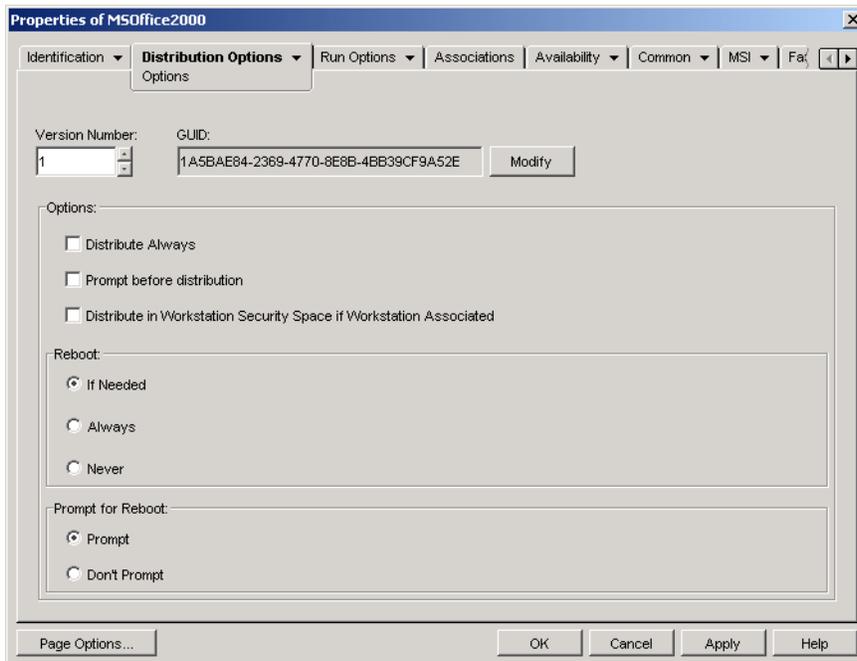
Select a process from the list, then click the up-arrow to move the process up in the list or click the down-arrow to move it down in the list. Application Launcher terminates the processes in the order they are listed, from top to bottom.

48.2.9 Options Page

The Options property page is available on Application objects created for all application types (simple, AOT/AXT, MSI, Web, and terminal server).

The Options property page, shown below, determines general options to be used by Application Launcher when distributing the application to a workstation.

Figure 48-15 Application Object > Distribution Options Tab > Options Page



GUID and Version Number

Application Launcher uses the application's GUID (global unique identifier) and version number to manage the distribution of the application. When Application Launcher distributes an application to a workstation, it adds the GUID and the version number to the workstation's Windows registry. If either the GUID or version number changes, Application Launcher redistributes the application.

The GUID is randomly generated when the Application object is created. In general, you should not need to change the GUID. There are, however, situations such as the following that might necessitate changing an application's GUID:

- ◆ The Application object is accidentally deleted from eDirectory. You re-create the Application object, but when you do so it is given a new, unique GUID. Because the new GUID causes the application to be redistributed to all users and workstations associated with the application, you use the GUID Manager (available by clicking the *Modify* button) to change the new GUID to the old GUID.
- ◆ You have multiple Application objects for the same application (to enable fault tolerance, load balancing, site lists, and so forth). You want to make sure that all Application objects have the same GUID so that the application is distributed one time only regardless of the Application object that is used. You use the GUID Manager to synchronize the GUIDs.
- ◆ Site 1 and Site 2 have the same applications chains. You must synchronize the GUIDs of each application in the chain at Site 1 with the GUIDs of each matching application at Site 2. Use the GUID Manager to synchronize the GUIDs.

The version number is a unique number between 0 and 65535 (0 is assigned when the Application object is first created) that you can increment as you make revisions to the Application object. If you make a change to the Application object's information, you should increment the version number so that Application Launcher redistributes the application. Application Launcher redistributes the

application only if the new version number is larger than the current version number in the workstation's Windows registry.

Options

These options let you determine if Application Launcher should redistribute the application each time the application is run and if Application Launcher should prompt the user to accept or reject the distribution. The Options fields are not displayed on Application objects created for Web applications and Terminal Server applications because they do not apply

Distribute Always

By default, Application Launcher makes the distribution changes associated with the Application object at the following times:

- ◆ The first time the application is launched on a workstation.
- ◆ The first time the application is launched after the application's version number has been changed.

To force Application Launcher to redistribute the application each time it is launched, select *Distribute always*.

This option is useful to ensure that all application settings and files are updated every time the application runs. If the user has a NAL cache directory on his or her local machine, the files and settings are distributed from the NAL cache directory. If the user does not have a NAL cache directory (for example, the user is running Application Launcher through a terminal server client session) or if writing to the cache has been disabled for the user (User object > *ZENworks* tab > *Launcher Configuration* page > *Enable writing to cache* option), the application files and settings are updated from eDirectory. To force a distribution from eDirectory even if the user has a NAL cache directory on his or her local machine, you need to change the application's version number or have individual users right-click the Application object and click *Verify*.

If you need only specific files or settings to be distributed each time, you can update these on a case-by-case basis. For example, if you want to always distribute a particular registry key and value, you can set the *Distribute always* option on the Registry Settings page (*Distribution Options* tab) for that particular key and value.

Because this setting causes all application files and settings to be distributed each time, it overrides the *Distribute always* option on the Registry, INI Settings, Application Files, Icons/Shortcuts, and Text Files pages (**Distribution Options** tab).

Prompt Before Distribution

Select this option to prompt users to accept the distribution. Users are prompted the first time they click the application icon; all subsequent times they are not prompted. To better help users make a decision about installing the application, the prompt includes the text you've entered in the Description page (*Identification* tab).

Distribute in Workstation Security Space if Workstation Associated

The function of this option is conditional, based on whether you are configuring an MSI application or an AOT/Simple application.

If the application is an MSI: This check box is *not* selected by default. Workstation-associated MSI applications are normally distributed in the user security space, meaning that Application Launcher uses the user's credentials and file system access.

Select this option to instruct Application Launcher to distribute the application *in the workstation security space*. Application Launcher turns over distribution to the NAL Workstation Helper, which runs in the system space and uses the workstation's credentials. Using this option enables you to 1) do a lights-out distribution of the application and 2) better secure the application's source .msi files by giving the workstation, not the user, access to the source .msi files. Consider the following examples:

- ◆ You want to associate the application with a workstation and have it distributed before the user launches it. This is referred to as a lights-out distribution. To do so, you associate the application with the workstation on the Associations page (*Associations* tab), set the distribution schedule on the *Pre-Install Schedule* page (*Distribution Options* tab), and then enable this option. As long as the workstation is running at the scheduled distribution time, the NAL Workstation Helper distributes the application using the workstation security space rather than the user space normally used for installation of MSI applications.
- ◆ You want to distribute the application to a workstation but you don't want to give the user rights to the application's source files on the network. To do so, you associate the application with the workstation on the *Associations* page (*Associations* tab) and then select this option. When the user launches the application, Application Launcher calls NAL Workstation Helper. Workstation Helper distributes the application using the workstation security space.

It is important to remember that Application Launcher uses the workstation's credentials, not the user's credentials, to distribute the application. This means that you must assign the workstation the appropriate file system rights to access the network location where the source .msi files reside.

Not all MSI applications can be installed using this option. Some MSI applications have dependencies on a logged-in user (for example, to read and write to the HKCU hive in the Windows registry). In this situation, you must deselect this option in order to have the distribution occur in the user security space and not the workstation security space.

NOTE: If an application requires a reboot during installation, you must select *Reboot if needed* or *Reboot always* in the *Reboot* group box and *Don't prompt* in the *Prompt for reboot* group box.

If the application is an AOT or a simple application: This check box is selected by default. Workstation-associated AOT or simple applications are normally distributed in the workstation security space, meaning that Application Launcher uses the workstation's credentials and file system access.

Deselect this option to instruct Application Launcher to distribute the application *in the user security space*. Application Launcher then runs in the user space and uses the user credentials to distribute the files, even if the application is associated to the workstation.

Reboot

Select how a workstation reboot should occur. The available options are:

- ◆ **If needed:** Application Launcher reboots the workstation if changes need to be made that cannot occur while Windows is running (such as replacing open DLLs).
- ◆ **Always:** Application Launcher always reboots the workstation after distributing the application.

- ◆ **Never:** Application Launcher does not reboot the workstation. The changes take effect the next time the workstation reboots.

The NAL Service, which runs in the “system” space rather than the “user” space, distributes workstation-associated applications on Windows 2000/XP workstations. If you select the *Always* option, the NAL Service automatically reboots the workstation, even if you've set the *Prompt reboot* option to *Prompt* (see below); in other words, the NAL Service ignores the *Prompt reboot* setting. The same is true if you select the *If needed* option and a reboot is needed.

Prompt Reboot

Select whether or not the user is prompted to reboot the workstation. If you select *Prompt*, but deselect the *Show progress* option (*Identification* tab > *Icon* page), the user is not prompted (in other words, disabling the *Show progress* option overrides enabling the *Prompt reboot* option).

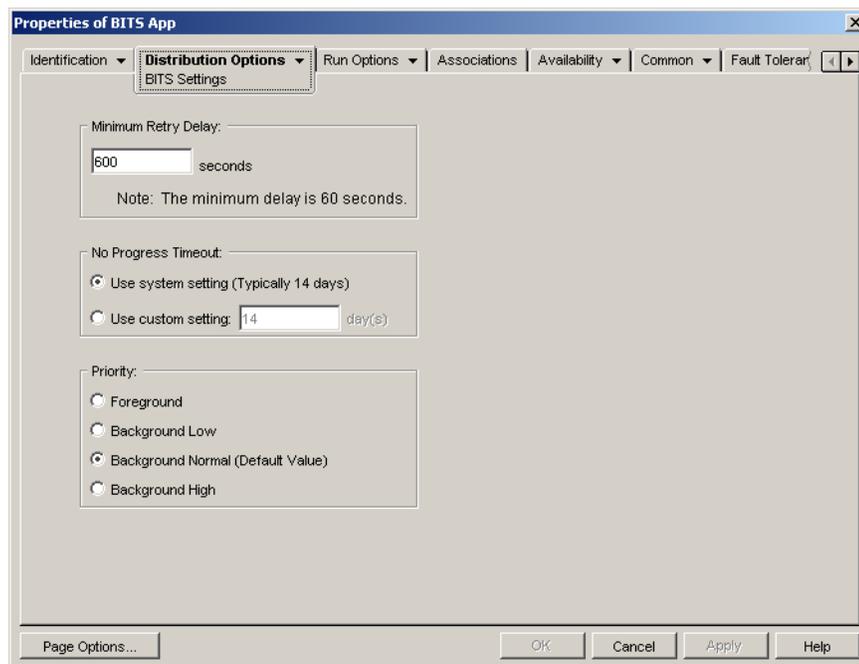
48.2.10 BITS Settings Page

This functionality is available beginning with Support Pack 1 of Novell ZENworks 7.

The BITS Settings property page is available only on Application objects created for simple applications, AOT/AXT applications, and MSI applications. It is not available on Application objects created for Web applications and terminal server applications.

The BITS Settings page lets you configure the settings used by the Microsoft Background Intelligent Transfer Service (BITS) when transferring the application to a workstation. BITS is used only if Novell Application Launcher and the application are configured to use BITS (see [Chapter 34](#), “Advanced Distribution: Transferring Applications Using BITS,” on page 365).

Figure 48-16 Application Object > Distribution Options Tab > BITS Setting Page



Minimum Retry Delay

If BITS encounters an error during transfer of the application, BITS classifies it as a fatal error or a transient error. BITS cannot recover from fatal errors; fatal errors require administrator intervention to fix the problem. BITS can possibly recover from transient errors.

Use this option to specify the minimum amount of time you want BITS to wait after a transient error occurs before trying to transfer the application again. The default is 600 seconds, or 10 minutes. The minimum setting is 60 seconds. The maximum setting is 2,147,483,647 seconds.

No Progress Timeout

Use this option to specify how many days you want BITS to continue to attempt to transfer the application after a transient error has occurred if no progress is being made.

- ♦ **Use System Setting (Typically 14 Days):** Select this option to use the Windows system setting. The Windows system setting comes from either 1) the BITS default setting, which is 14 days, or 2) the Timeout (Days) for Inactive Jobs setting in the Windows Group Policy, which is undefined by default. If you select this option, the BITS default setting (14 days) is used unless the Windows Group Policy setting has been assigned a value. You can use the Windows Group Policy Editor (gpedit.msc) to view and change the Windows Group Policy setting.
- ♦ **Use Custom Setting:** Select this option to manually enter a timeout period.
 - ♦ The minimum setting is 0 days. Enter 0 only if you do not want BITS to attempt to transfer the application again after it encounters a transient error; in this case, BITS immediately returns control of the transfer to Application Launcher.
 - ♦ The maximum setting is 24,855 days. However, BITS compares this number to the number in the Timeout (Days) for Inactive Jobs setting in the Windows Group Policy. If the Timeout (Days) for Inactive Jobs setting is less than this number, BITS uses the policy setting. For example, if you enter 45 days for this setting, but the policy is set to 30 days, BITS uses 30 days. If the Timeout (Days) for Inactive Jobs setting is undefined (the default state), the policy setting defaults to 90 days. In this case, for example, if you enter 91 days in this setting, BITS uses the policy setting (90 days).

If any transfer progress is made during the timeout period, the counter is reset. If BITS times out because no progress is being made, control of the transfer is returned to Application Launcher, which then transfers the application using its standard distribution process.

Priority

Use this option to assign a transfer priority level to the application. You can choose from one foreground priority and three background priorities (low, normal, high).

The foreground priority causes BITS to transfer the application in the foreground. Foreground transfers are the highest priority and are processed before any background transfers. Foreground transfers compete for network bandwidth with other applications, which can impede the user's network experience. Unless the timing of the transfer is critical or the user is actively waiting, you should use a background priority. In addition, BITS only supports foreground priority for files less than 2 GB.

For the three background priorities, the priority level determines when the transfer is processed relative to other transfers in the queue. Higher priority transfers preempt lower priority transfers. Transfers with the same priority level share transfer time, which prevents a large transfer from

blocking the transfer queue. Lower priority transfers do not receive transfer time until all higher priority transfers are completed or are in an error state.

48.3 Run Options Tab

The Run Options tab includes the following pages to help you configure how the application runs on the workstation:

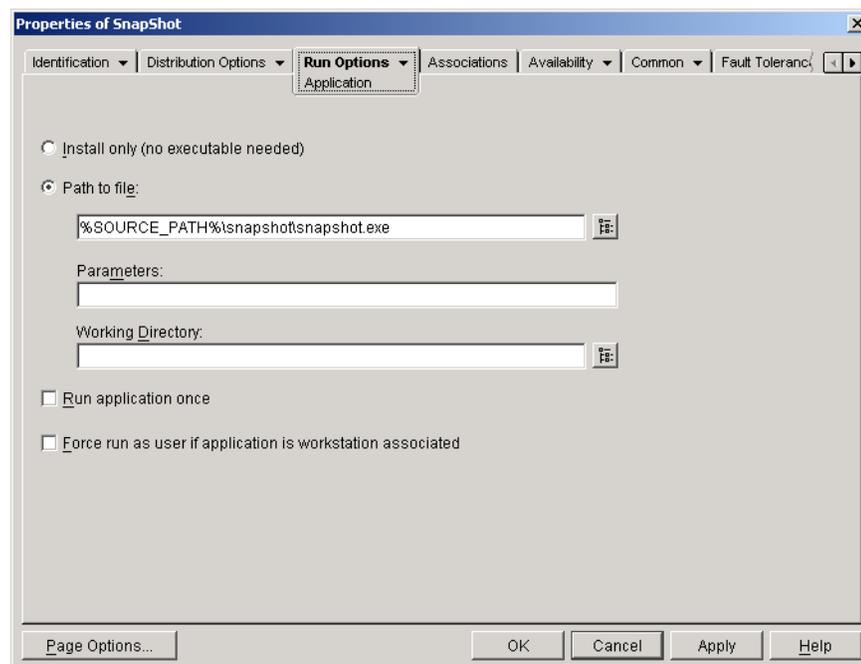
- ◆ Section 48.3.1, “Application Page,” on page 520
- ◆ Section 48.3.2, “Environment Page,” on page 523
- ◆ Section 48.3.3, “Launch Scripts Page,” on page 525
- ◆ Section 48.3.4, “Environment Variables Page,” on page 528
- ◆ Section 48.3.5, “Web URL,” on page 529
- ◆ Section 48.3.6, “License/Metering Page,” on page 530
- ◆ Section 48.3.7, “Application Dependencies Page,” on page 531

48.3.1 Application Page

The Application property page is available only on Application objects created for simple applications, AOT/AXT applications, and MSI applications. It is not available on Application objects created for Web applications and terminal server applications.

The Application property page, shown below, determines general options to be used by Application Launcher when running the application on a workstation.

Figure 48-17 Application Object > Run Options Tab > Application Page



Install Only (No Executable Needed)

Select this option if the distribution does not include an executable file for Application Launcher to launch after the distribution. For example, you would want to use this option if the Application object's only purpose is to update some files on the workstation, such as a driver or font, or to make modifications to some workstation settings, such as the registry settings.

Path to File

Enter, or browse for, the path to the executable that Application Launcher runs after the application is distributed. Typically, this path is defined when you create the Application object and should already contain the path to where the executable is located on the network or where it is located on the user's workstation after distribution. If you enter the path, make sure to include the `.exe` extension on the application executable filename (for example, enter `notepad.exe`, not `notepad`).

IMPORTANT: The ZENworks Desktop Management Agent and Middle Tier Server do not support launching of applications from a network location. If the executable is located on the network, you should only use this option if users have a network client (Novell Client or Microsoft Client for Networks) installed that enables launching from a NetWare[®], Linux, or Windows server.

You can also enter the path for a file type other than `.exe`. If you do so, when the user double-clicks the Application object, Application Launcher passes the file to the Windows system, which can then use the application associated with that file type to open the file. For example, if you entered a `.txt` file, the application associated with `.txt` files would be used to open the file.

You can use a mapped drive or a UNC path. The following path syntaxes are valid:

```
server\volume:path
\\server_name\volume\path
\\IP_address\volume\path
volume_object_name:path
directory_map_object_name:path
driveletter:\path
```

If you use a UNC path, applications running on Windows 2000 servers might launch slowly because of the way Windows resolves UNC paths. For more information, see [Microsoft Knowledge Base Article Q150807 \(http://support.microsoft.com/support/kb/articles/Q150/8/07.asp\)](http://support.microsoft.com/support/kb/articles/Q150/8/07.asp).

You can also use macros in this field. For information about macros, see [Chapter 49, "Reference: Macros,"](#) on page 593.

Parameters

Specify any command line parameters that need to be passed to the executable specified in the *Path to file* field. The field limit is 250 characters.

Application Launcher simply passes the parameters exactly as they are entered. Therefore, the parameter syntax you use must exactly match the syntax the executable requires. For example, if `word.exe` has a `/f=filepath` parameter that requires paths with spaces to be enclosed in quotation marks, you would enter the following:

```
/f="c:\my docs\sample.doc"
```

Working Directory

Specify the working directory of the executable you specified in the *Path to file* field. Typically, this is the directory in which the executable file is located. However, if another directory contains additional files required by the application to run, specify that directory. See your Windows documentation for more information.

The following path syntaxes are valid:

```
server\volume:path  
\\server\volume\path  
volume_object_name:path  
directory_map_object_name:path  
driveletter:\path
```

You can also use macros in this field. For information about macros, see [Chapter 49, “Reference: Macros,” on page 593](#).

Run Application Once

Select this option to have Application Launcher run the application one time only and then remove the Application object’s icon from the workstation. If the Application object’s version number (*Distribution Options > Options* page) changes, the Application object icon appears again so that the new version can be distributed.

This option can be useful when an Application object’s only purpose is to install software to a workstation. Having the Application object’s icon remain after the application has been installed can be confusing to users, especially if you have an application shortcut or program group icon created during the distribution (see the *Distribution Options > Icons/Shortcuts* page).

This option can also be used in combination with other Application object settings. For example, if the application is specified as *Force run* on the Associations page (meaning that without any user involvement, the application runs when it becomes available to Application Launcher), the application is forced to run only one time.

Force Run As User If Application Is Workstation Associated

With workstation-associated applications that are marked as *Force run* (*Associations* tab > *Associations* page), the Workstation Helper launches the application as soon as the workstation is started, typically before the user has logged in to Windows. This means that the only workstation-associated applications you would want to force run would be applications that don’t require user interaction.

Select this option if you want to delay the force running of a workstation-associated application until the user logs in to Windows and Application Launcher starts. Distribution of the application occurs in the workstation system space prior to the user logging in to Windows. After the user is logged in and Application Launcher starts, Application Launcher runs the application using the user’s credentials.

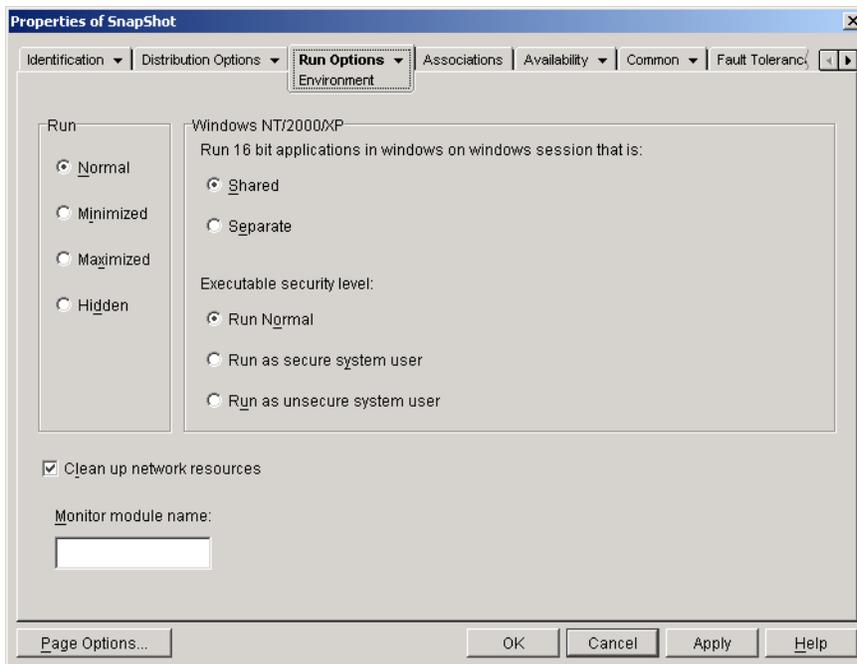
Because the distribution takes place in the workstation system space using the workstation’s credentials, you should not use any Application object distribution settings that are dependent on the user being logged in. For example, you would not want to have files copied to *c:\%USERNAME%* during the distribution, where *%USERNAME%* is the Windows environment USERNAME variable, because no user is logged in when the distribution occurs.

48.3.2 Environment Page

The Environment property page is available only on Application objects created for simple applications, AOT/AXT applications, and MSI applications. It is not available on Application objects created for Web applications and terminal server applications.

The Environment property page, shown below, defines the environment in which Application Launcher runs the application.

Figure 48-18 Application Object > Run Options Tab > Environment Page



Run

Select the application's initial window size: Normal, Minimized, Maximized, or Hidden. In Hidden mode, the application runs normally without a user interface available. This is useful if you want the application to process something then go away without user intervention. For example, you could use it for the Remote Control agent if you don't want it to appear in the taskbar.

NOTE: Some applications cannot be run in Minimized, Maximized, or Hidden mode. Depending on the application, the initial window size setting might be ignored. For example, `calc.exe` cannot be run in Hidden mode.

Windows NT/2000/XP

Use these options to specify how the application runs on Windows 2000/XP workstations.

Run 16-Bit Applications in Windows on Windows Session That Is

If you are setting up a 16-bit application to run on Windows 2000/XP, specify either a shared or separate Windows on Windows (WOW) session. To run on Windows 2000/XP, 16-bit applications must run in a Virtual DOS Machine (VDM) that emulates Windows 3.1 functionality. 16-bit

Windows applications can run in their own separate WOW sessions or they can share a WOW session with other 16-bit applications. Sharing a WOW session saves memory and allows applications to use DDE or OLE to communicate with other applications in the same WOW session. Running an application in its own separate WOW session allows for more robust operation because the application is isolated from other 16-bit applications. Because some applications do not run properly in a separate WOW session, the default option is to run the 16-bit Windows application in a shared WOW session.

Executable Security Level

On Windows 2000/XP, the application executable (as defined in the *Path to file* field on the **Application Page**) can run in either the “user” space or the “system” space. By default, the *Run normal* option is selected, which causes the application to run in the “user” space and inherit the same workstation security level as the logged-in user.

If the logged-in user's security level does not provide sufficient rights and file access to run the application, you can configure the application to run in the “system” space. The two “system” space options, as well as the *Run normal* (“user” space) option are described below:

- ◆ **Run normal:** The application inherits the logged-in user’s credentials. For example, the application has the same rights to the registry and the file system as the logged-in user.
- ◆ **Run as secure system user:** The application is run under the Local System user and inherits Administrator-level credentials. For example, the application has full rights to the registry and the file system. Because the security level is set to *Secure*, the application's interface is not displayed to the user and the application is only visible in the Task Manager. This option is useful when running applications that require full access to the workstation but require no user intervention, such as a service pack or update.
- ◆ **Run as unsecure system user:** This option is the same as *Run as secure system user*, with the exception that the application's interface is displayed to the user. This means that the application can display dialog boxes and prompt the user for information if necessary. In return, the user has the ability to do whatever that application offers on the workstation. For example, if you run REGEDIT as an unsecure system user, the workstation’s user is able to edit any registry values.

In a terminal server environment, you should not use the *Run as secure system user* and *Run as unsecure system user* options if users are running Application Launcher in a terminal server client session. Doing so causes the NAL Service (running on the terminal server) to launch the application. Because the NAL Service runs as a System user, the launched application displays only in the terminal server's Task Manager (for *Run as secure system user*) or on the terminal server's desktop (*Run as unsecure system user*). It does not display in the users' terminal server client sessions.

Clean Up Network Resources

Select this option to remove network connections, drive mappings, and port captures that were established for running the application. If this option is not selected, the network resources remain in effect. If the resource is already in use when Application Launcher starts the application, it is not cleaned up until all other distributed applications are finished using it. If distributed applications are still running when Application Launcher is terminated, the allocated resources remain intact.

Monitor Module Name

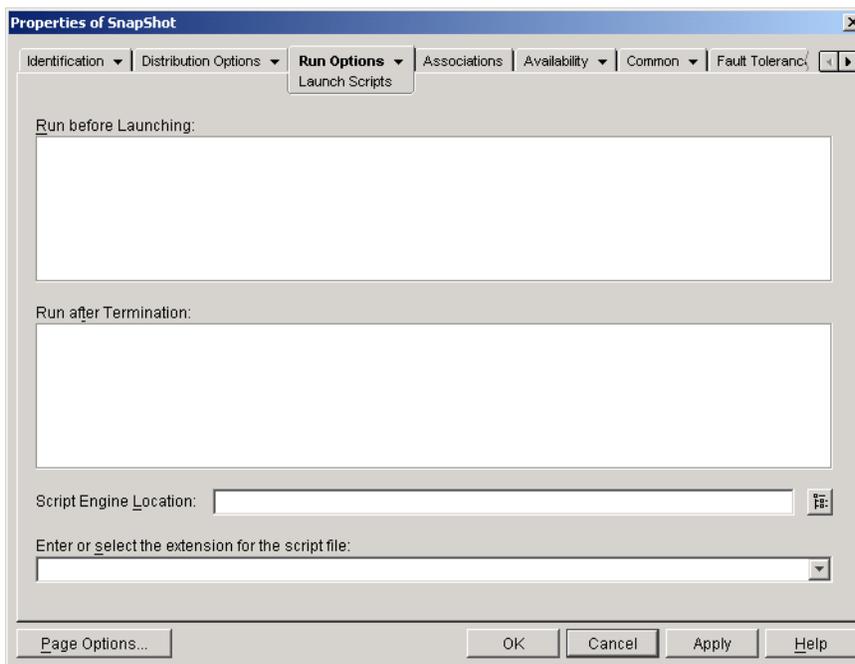
When an application is launched, Application Launcher monitors the executable of the application. When the executable terminates, the process of cleaning up network resources begins. However, it's possible that the executable filename is actually a “wrapper” that sets up environments, runs other executables, and then terminates. If Application Launcher monitors the wrapper executable, it might prematurely start cleaning up network resources before the application has terminated. Consult your application documentation about whether the application uses a wrapper executable. If it does, find out the name of the module that remains running. Type this name (without the extension) in the text box provided.

48.3.3 Launch Scripts Page

The Launch Scripts property page is available on Application objects created for simple applications, AOT/AXT applications, and MSI applications only. It is not available on Application objects created for Web applications and terminal server applications.

As part of the process of launching the application, Application Launcher can launch a script engine to execute a “before launch” script and an “after termination” script (for details about the order of script execution, see [“Script Execution Order” on page 527](#)). The Launch Scripts property page, shown below, defines the script engine that you want Application Launcher to use and the scripts you want executed.

Figure 48-19 Application Object > Run Options Tab > Launch Scripts Page



On Windows 2000/XP, launch scripts always run in the user space, meaning that they inherit the file system rights assigned to the logged-in user. This is true even if the application's executable is configured to run in the system space (*Run Options* tab > *Environment* page > *Executable security level* option). This behavior is different than the behavior for distribution scripts (*Distribution Options* tab > *Distribution Scripts* page), which run in the secure system space.

Run Before Launching

Use this text window to enter any script commands you want executed before launching the application. Do not use extended characters in the script; extended characters are not supported.

For a script example, see [“Script Example” on page 527](#).

Run After Termination

Use this text window to enter any script commands you want executed after the user closes the application. Do not use extended characters in the script; extended characters are not supported.

For a script example, see [“Script Example” on page 527](#).

The *Run after termination* script is executed before the network resources are cleaned up (*Run Options > Environment* page). However, the *Clean up network resources* option does not clean up drive mappings or port captures created by the *Run before launching* script. To clean up these resources, enter the appropriate commands in the *Run after termination* script.

Script Engine Location

The script engine determines the script commands and scripting language you need to use. If you do not define a script engine in the *Script engine location* field, Application Launcher uses the Novell Client as the script engine (if the workstation has the Novell Client installed), which means that you can use most Novell Client login script commands (see [“Supported Novell Client Login Script Commands” on page 509](#)).

If you want to use a script engine other than the Novell Client, specify the script engine. The script engine must reside in a location that is always available to users, such as their local drives. The script engine can reside on a network server only if users can map a drive to the server (for example, through the Novell Client or the Client for Microsoft Networks). If Application Launcher cannot find the script engine, it displays an error to the user and fails to launch the application.

If you use the Windows command interpreter as the script engine, you must include the `/c` switch, as shown in the following examples:

- ◆ Windows 2000/XP: `%*winsysdir%\cmd.exe /c`
- ◆ Windows 98: `%*windir%\command.com /c`

The `%*winsysdir%` and `%*windir%` variables specify the Windows system directory (for example, `c:\winnt\system32`), and the `/c` switch instructs the command interpreter to execute the script and then stop. If the `/c` switch is not used, the script does not complete.

For a script example, see [“Script Example” on page 527](#).

Script File Extension

This applies only if you specified a script engine in the *Script engine location* field.

When the application is launched, Application Launcher creates temporary script files for the *Run before distribution* scripts and *Run after distribution* scripts. These files are passed to the script engine, which then executes the script. You need to specify the file extension that the script engine requires for its script files.

For a script example, see [“Script Example” on page 527](#).

Script Example

The following script uses the Windows 2000/XP command interpreter as the script engine. Before the launch occurs, a listing of the `c:\` directory is saved to a text file and the `autoexec.bat` file is backed up.

Run Before Distribution Field

```
dir c:\ >c:\1.txt
copy autoexec.bat autoexec.bak /y
```

Script Engine Location Field

```
cmd.exe /c
```

Enter or Select the Extension of the Script File Field

```
.bat
```

Script Execution Order

Application Launcher can execute up to four different scripts when distributing and launching an application:

- ◆ **Distribution scripts:** *Run before distribution* and *Run after distribution* (*Distribution Options* tab > *Distribution Scripts* page)
- ◆ **Launch scripts:** *Run before launching* and *Run after termination* (*Run Options* tab > *Launch Scripts* page)

Application Launcher executes the scripts in the following order:

1. Run Before Launching script executed
2. Run Before Distribution script executed
3. Application distributed (files copied, settings modified, etc.)
4. Run After Distribution script executed
5. Application launched
6. Application closed (by user)
7. Run After Termination script executed

Supported Novell Client Login Script Commands

When using the Novell Client as the script engine, you can use all but the following script commands:

Table 48-2 Supported Novell Client Login Script Commands

CLS	INCLUDE	PCOMPATIBLE
DISPLAY	LASTLOGINTIME	SCRIPT_SERVER
EXIT	NO_DEFAULT	SET_TIME
FDISPLAY	NOSWAP	SWAP

Application Launcher does not output anything to the screen or display script errors.

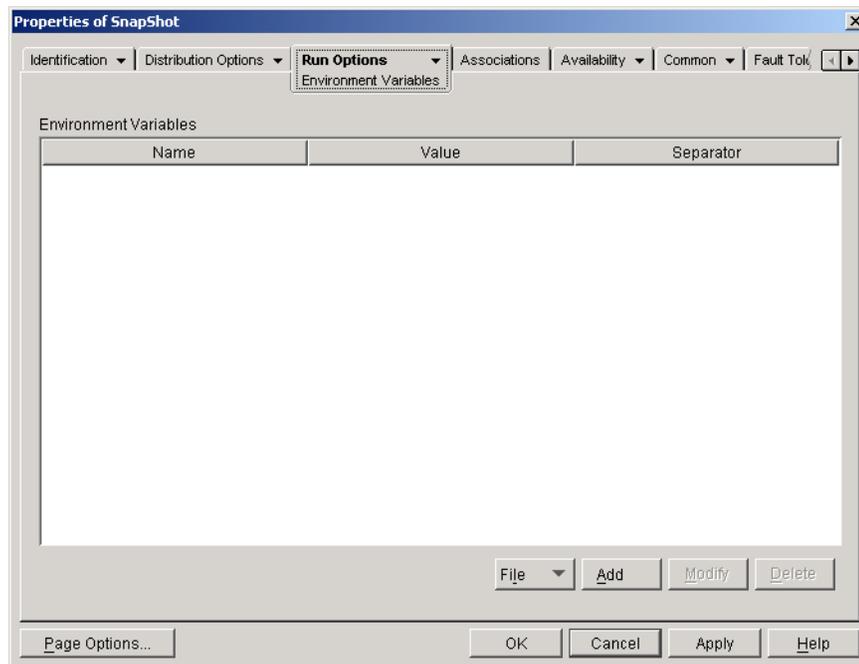
For script commands, syntax, and examples, see the [Novell Client documentation \(http://www.novell.com/documentation/lg/noclienu/docui/index.html\)](http://www.novell.com/documentation/lg/noclienu/docui/index.html) on the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

48.3.4 Environment Variables Page

The Environment Variables property page is available only on Application objects created for simple applications, AOT/AXT applications, and MSI applications. It is not available on Application objects created for Web applications and terminal server applications.

The Environment Variables property page, shown below, specifies the environment variables that Application Launcher sets before running the application. For example, you can specify the application's working directory in the PATH environment variable.

Figure 48-20 Application Object > Run Options Tab > Environment Variables Page



IMPORTANT: On Windows 98 workstations, Application Launcher cannot set environment variables if the Novell Client is not installed. This is not an issue on Windows 2000/XP workstations.

Environment Variables

The Environment Variables list displays all variables that are modified before the application is run. Application Launcher modifies only the variables in this list.

File

This option lets you search for variables or variable data in the *Environment variables* list and import variables from another Application object into the tree.

Click *File*, then choose one of the following options:

- ◆ **Find:** Searches for variables or variable data in the *Environment variables* list.
- ◆ **Find next:** Finds the next occurrence of the item specified by the initial search.
- ◆ **Import:** Imports variables from another Application object's *.aot* or *.axt* file. The Open dialog box defaults to **.axt* for its file type display. If you are importing from an *.aot* file, you must change the file type display to **.aot* or *All files* in order to select the *.aot* file.

Add

To add a variable to the *Environment variables* list, click *Add* to display the Edit Environment Variables dialog box, specify the variable name and data, select *Append to existing data using separator character* (only if you want to append the data to the workstation's existing variable) and enter a separator character, then click *OK*.

Modify

In the Environment Variables list, select the variable you want to modify, then click *Modify*.

Delete

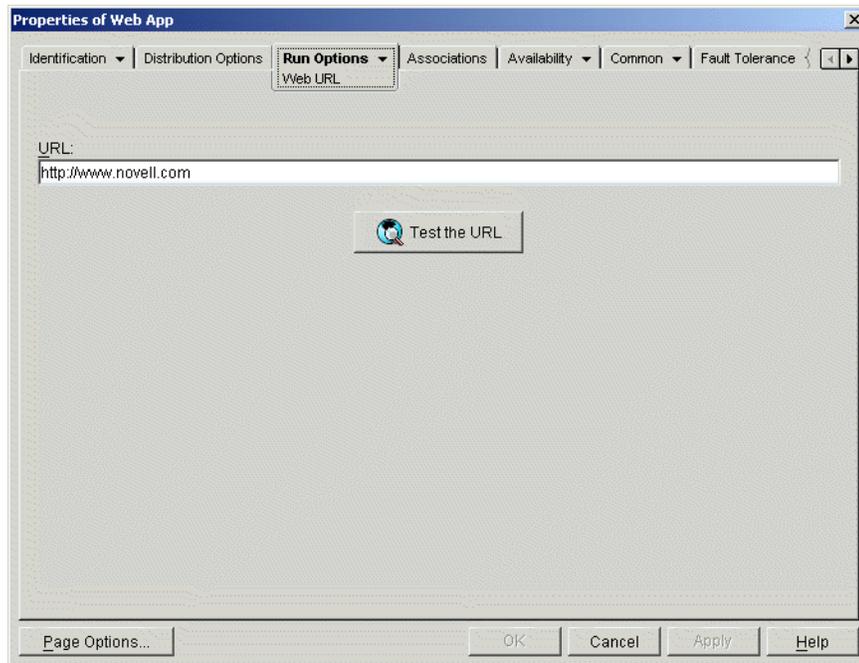
In the Environment Variables list, select the variable you want to delete, then click *Delete* to remove it from the list.

48.3.5 Web URL

The Web URL property page is available on Application objects created for Web applications only. It is not available on Application objects created for simple, AOT/AXT, MSI, and terminal server applications.

The Web URL property page, shown below, specifies the location (URL) of the Web application.

Figure 48-21 Application Object > Run Options Tab > Web URL Page



URL

This field defaults to the URL you entered when creating the Application object. The URL should point to the primary file for the Web application or to a Web page that allows access to the Web application. If necessary, you can modify the URL.

Test the URL

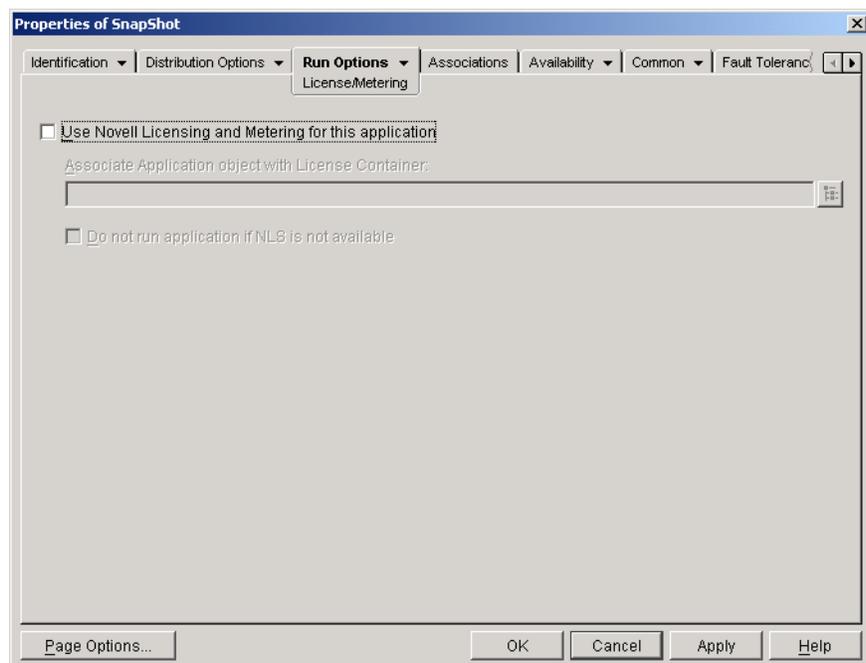
Click this button to launch the Web application. The current workstation must have a Web browser installed for this functionality to work.

48.3.6 License/Metering Page

The License/Metering property page is available on Application objects created for all application types (simple, AOT/AXT, MSI, Web, and terminal server).

The License/Metering property page, shown below, configures Application Launcher to use Novell Licensing Services to track application usage and comply with the application's licensing agreement. Novell Licensing Services must already be installed, you must have created a License Container and a Metered Certificate for the application, and users' workstations must be running the Novell Client. For detailed information about setting up license metering, see [Chapter 45, "Metering Software Licenses,"](#) on page 469.

Figure 48-22 Application Object > Run Options Tab > License/metering Page



Use Novell Licensing and Metering for this Application

Select this option to enable metering of the application.

Associate Application Object with License Container

After metering is enabled, select the License Container in which you've created the application's Metered Certificate.

Do Not Run Application If NLS Is Not Available

Select this option to prevent users from running the application if Novell Licensing Services is not running.

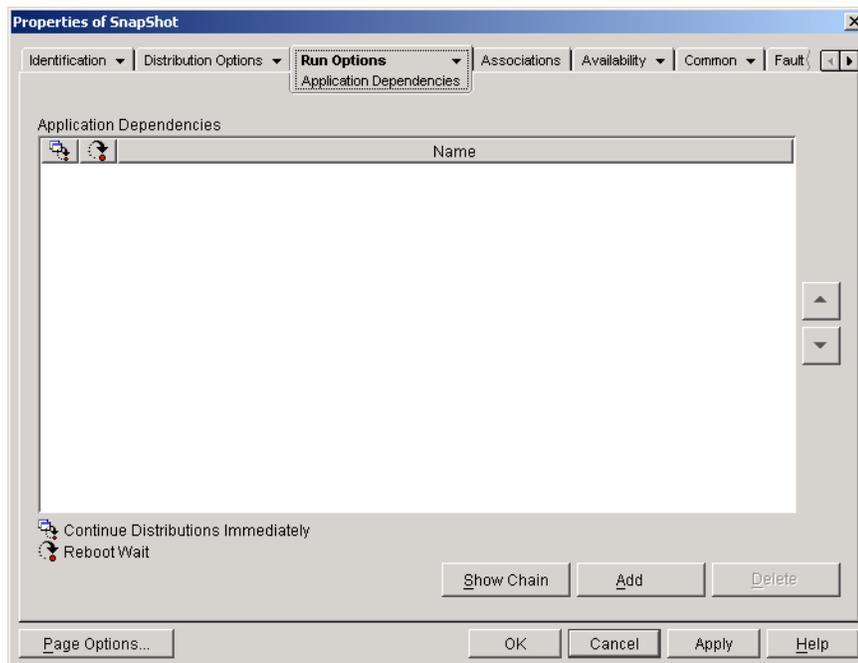
48.3.7 Application Dependencies Page

The Application Dependencies property page is available on Application objects created for all application types (simple, AOT/AXT, MSI, Web, and terminal server).

This section provides information about the fields located on the Application Dependencies property page. For complete instructions on setting up application dependencies, see [“Advanced Distribution: Configuring Application Dependencies and Chains” on page 399](#).

The Application Dependencies property page, shown below, lets you create a list of applications (referred to as dependent applications) that must be distributed and launched before this application (referred to as the primary application) is distributed and launched.

Figure 48-23 Application Object > Run Options Tab > Application Dependencies Page



You can specify one or more dependent applications. For example, assume that the primary application (Application A) is dependent on three other applications (Application B, Application C, and Application D). By adding all three applications to the Application Dependencies list on this page, Application Launcher ensures that all three dependent applications are distributed and launched before running Application A.

A dependent application can be a single application or it can be an application chain. An application chain exists when a dependent application is dependent on additional applications. Continuing the above example, if Application C is dependent on Application C1 and Application C2, then Application B, C, C1, C2, and D must all be distributed and launched before Application A is distributed and launched.

When setting up application dependencies, you should be aware of the following:

- ◆ **Distribution:** Novell Application Launcher only distributes a dependent application the first time the user launches the primary application, unless the dependent application's version is updated (Application object > *Distribution Options* tab > *Options* page) or the distribution is unsuccessful. If Application Launcher is unable to distribute a dependent application (for example, the user's workstation does not meet the dependent application's system requirements), the primary application is not distributed or launched.
- ◆ **Distribution through removable media:** If you distribute the primary application through removable media such as a CD (see [“Distributing Applications Via Removable Media” on page 421](#)) or through an image add-on (see [“Imaging Page” on page 568](#)), its dependent applications must be included on the removable media or as another image add-on.
- ◆ **Distribution to disconnected workstations:** For workstations running in disconnected mode, the applications must be force cached to the workstation before it becomes disconnected (or they must be distributed through removable media). If changes are made to dependent applications after the applications have been cached to workstations, the version number of the primary application must be updated in order to force a re-cache of the applications.

- ♦ **Launch:** Application Launcher launches a dependent application each time the primary application is launched. During launching, Application Launcher executes the dependent application's pre-launch and post-launch scripts, and then launches the dependent application. If a dependent application, such as a setup program, should be run only one time, you should configure the dependent application to run once (Application object > *Run Options* tab > *Application* page).
- ♦ **Uninstall:** If you uninstall the primary application (*Common* tab > *Uninstall* page or *Associations* tab > *Associations* page), all dependent applications that are not used by another application are also uninstalled.

Application Dependencies List

The Application Dependencies list displays the applications that the primary application has a dependency on. The order in which the dependent applications are listed, going from bottom to top, is the order in which they are distributed and launched on the user's workstation when the primary application is launched.

Add

To add an application to the list, click *Add*, browse for and select the Application object, then click *OK*. After the application is added to the list, you can modify the following two attributes:

- ♦ **Continue distributions immediately:** Select this option if you want Application Launcher to distribute and run the application but not wait for the application to finish running before continuing with the next application distribution.

If this option is not selected (the default), Application Launcher waits until it has distributed and run the application before continuing with the distribution of the next application in the list. Application Launcher only continues after all files, registry settings, and so forth have been distributed and the application (or file) listed in the *Path to file* field (*Run* tab > *Application* page) has been run and completed.

- ♦ **Reboot wait:** If distribution of the application requires the workstation to be rebooted, select this option to have Application Launcher postpone the reboot until 1) it distributes another application in the list that requires a reboot (and doesn't have this option selected) or 2) it has distributed all applications in the list. This option also applies when uninstalling the application.

Delete

To remove a dependent application from the list, select the application in the list, then click *Delete*.

Show Chain

To see whether or not a dependent application consists of an application chain, select the dependent application in the list, then click *Show chain*.

Up-Arrow and Down-Arrow

The order in which the dependent application are listed is the order in which they are distributed and launched, going from bottom to top. You can use the up and down arrows to change the order.

48.4 Associations Tab

The Associations tab includes only one page. This page lets you associate the Application object with users and workstations.

- ◆ [Section 48.4.1, “Associations Page,” on page 534](#)

48.4.1 Associations Page

The Associations property page is available on Application objects created for all application types (simple, AOT/AXT, MSI, Web, and terminal server).

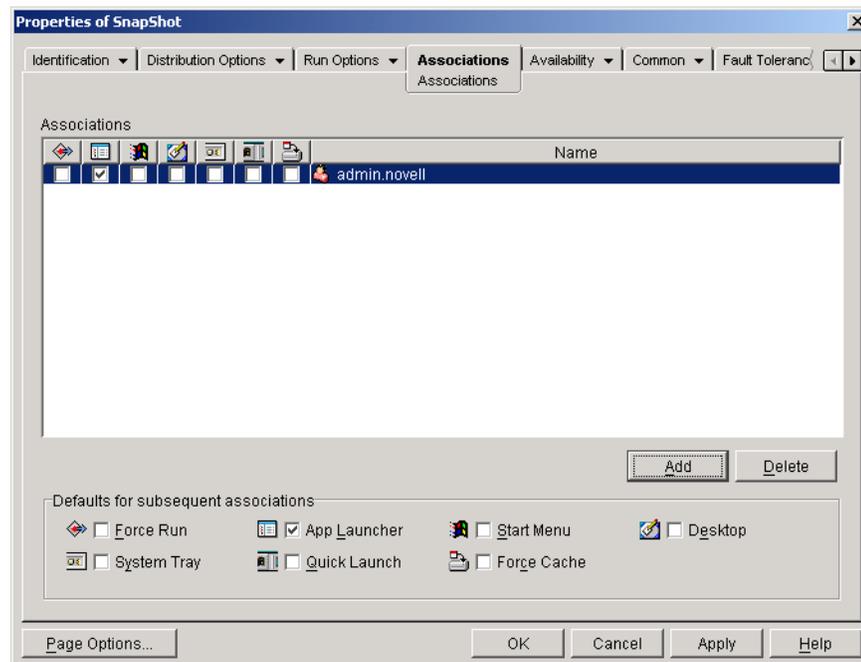
The Associations property page, shown below, lets you associate the Application object with the users and workstations to which you want the application distributed.

You can associate the Application object with individual users or workstations by selecting the appropriate User or Workstation objects. You can also associate it with groups of users or workstations by selecting Group, Workstation Group, Organizational Unit, Organization, and Country objects.

NOTE: Associating an Application object with a Group, Workstation Group, or other container object is the preferred method of associating the Application object in eDirectory. Associating the application to a large number of User or Workstation objects (for example, more than 250) might cause increased server utilization.

Although you are not restricted from associating Application objects with Alias objects, do not do so; Alias objects are not a supported object for associations.

Figure 48-24 Application Object > Associations Tab > Associations Page



Associations

The *Associations* list displays the objects associated with the Application object and lets you add or delete objects.

Add

To associate an object with the Application object, click *Add*, browse and select the object, then click *OK*. All objects receive the default characteristics (*Force run*, *App Launcher*, *Start menu*, *Desktop*, *System tray*, *Quick launch*, *Force cache*) selected in the *Defaults for subsequent associations* list. After you add an object to the *Associations* list, you can modify its characteristics by selecting or deselecting the appropriate check boxes.

Delete

To disassociate an object from the Application object, select the object in the *Associations* list, then click *Delete*.

Defaults for Subsequent Associations

When you add an object to the Associations list, the object receives the default characteristics selected in this box. The characteristics determine the locations (App Launcher, Start Menu, Desktop, System Tray, and Quick Launch) where the Application object's icon are displayed and the actions (*Force run* and *Force cache*) that are performed on the application. If you change the characteristics that are selected, any objects you add after that receive the new characteristics.

Force Run

Automatically runs the application. With a user-associated application, the application is run immediately after Application Launcher starts. With a workstation-associated application, the application is run immediately after the workstation starts up (initial startup or reboot).

You can use the *Force run* option in conjunction with several other settings to achieve unique behaviors. For example, if you use the *Force run* option with the *Run application once* option (*Run Options* tab > *Applications* page), as soon as the Application object is distributed it is run one time and then removed from the workstation. Or, suppose that you want to run the Application object immediately one time at a pre-determined time. If so, select *Force run*, select the *Run application once* option on the *Application* page (*Run Options* tab), and define a schedule using the *Schedule* page (*Availability* tab).

If you want to force run several applications in a specific order, mark each of them as *Force run*. Then give each Application object a numeric order by using the *Order icon display* option on the *Icon* page (*Identification* tab).

IMPORTANT: When using the *Force run* option for an MSI application associated with a workstation, you must also turn on the *Distribute in workstation security space if workstation associated* option (*Distribution Options* tab > *Options* page) and assign the workstation the appropriate file system rights to access the network location where the source `.msi` files reside. Otherwise, the distribution and subsequent forced running of the application fails.

If an application requires a reboot during installation, you must select *Reboot if needed* or *Reboot always* in the *Reboot* group box and *Don't prompt* in the *Prompt for reboot* group box.

App Launcher

Adds the Application object's icon to the Application Window, the Application Explorer window, and the Application Browser window.

Start Menu

If the workstation is running Application Explorer, this option adds the Application object to the Windows Start menu. The Application object is added to the top of the Start menu, unless you assign the Application object to a folder and use the folder structure on the Start menu. See the Application object's Folders page (*Identification* tab).

Desktop

If the workstation is running Application Explorer, this option displays the Application object's icon on the Windows desktop.

System Tray

If the workstation is running Application Explorer, this option displays the Application object's icon in the Windows system tray.

Quick Launch

If the workstation is running Application Explorer, this option displays the Application object's icon on the Windows Quick Launch bar.

Force Cache

This option applies only if the Application object is marked as disconnectable (*Identification* tab > *Icon* page).

Without *Force cache* enabled, users can launch the application while in disconnected mode as long as the application has already been distributed (installed) to the workstation. With *Force cache* enabled, Application Launcher automatically copies the application source files and other files required for distribution to the workstations cache directory. The user can then install or repair the application while disconnected from Novell eDirectory. The files are compressed to save space on the workstation's local drive. For more information about caching applications, see [Chapter 39, "Users: Supporting Disconnected Users," on page 411](#).

When distributing an MSI application, you must use the *Force cache* option if users or workstations do not have network client access to the source `.msi` files. The Microsoft Windows Installer requires file access that is provided by a network client but not by the ZENworks Desktop Management Agent. Consider the following examples:

- ◆ Users outside your firewall need an MSI application but have no network client access to the source `.msi` files on a server inside your firewall. They log in to the ZENworks Middle Tier Server and Application Launcher displays the MSI application. For successful distribution to occur, the MSI application must be marked as *Force cache* so that the source `.msi` files are copied to the user's cache directory (through the Middle Tier Server) and then distributed from the cache directory.
- ◆ Users inside your firewall need an MSI application. They don't have the Novell Client installed, so they log in to the ZENworks Middle Tier Server to authenticate to eDirectory. The users are part of an Active Directory domain, and the source `.msi` files are located on a Windows share

that they have rights to. The distribution succeeds without force caching the application because the Microsoft network client provides file access to the source .msi files.

48.5 Availability Tab

The Availability tab includes the following pages to help you define the rules for distributing the application to workstations, the days and times the application is available to users, and the messages and warnings Application Launcher displays to users when terminating an application that has become unavailable:

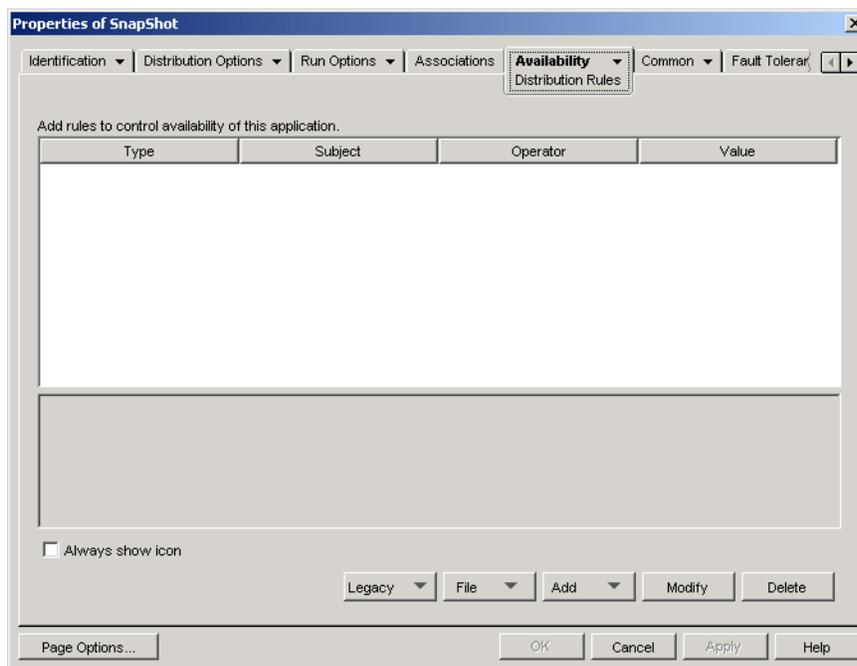
- ◆ [Section 48.5.1, “Distribution Rules Page,” on page 537](#)
- ◆ [Section 48.5.2, “Schedule Page,” on page 552](#)
- ◆ [Section 48.5.3, “Termination Page,” on page 555](#)

48.5.1 Distribution Rules Page

The Distribution property page is available on Application objects created for all application types (simple, AOT/AXT, MSI, Web, and terminal server).

The Distribution Rules property page, shown below, lets you define the rules applied to a workstation to determine whether or not Application Launcher distributes the application to it.

Figure 48-25 *Application Object > Availability Tab > Distribution Rules Page*



For example, you can define a disk space rule that instructs Application Launcher to only distribute the application on a workstation if the available disk space is greater than 2 GB. Or, you can define a rule that requires at least a Pentium III processor.

When creating rules, you can use AND/OR Boolean logic. For example, you can define a rule that instructs Application Launcher to only distribute the application if the available disk space is greater

than 2 GB AND the processor is at least a Pentium III. You can also create groups of rules. For examples of distribution rules that use Boolean logic and groups, see “[Distribution Rules Examples](#)” on page 540.

The distribution rules are used only by the ZENworks 7 (or later) versions of Application Launcher. Pre-ZENworks 7 versions of Application Launcher do not use distribution rules and require the use of legacy system requirements. If you have pre-ZENworks 7 versions of Application Launcher in your environment, see [Legacy](#) below for information about using legacy system requirements.

Add Rules to Control Availability of This Application

This list displays all the rules associated with the distribution of the application. For each rule, the following information is displayed:

- ◆ **Type:** You can define different types of rules, such as operating system rules, memory rules, and disk space rules. This column shows the rule’s type. The types are explained under [Add](#) below.
- ◆ **Subject:** This column shows the subject of the rule. For example, if the type is OS Version, the subject could be Windows 98 or Windows NT/2000/XP.
- ◆ **Operator:** This column shows if a condition (greater than, less than, equal to, etc.) must be met for the assigned value.
- ◆ **Value:** This column shows if a value is associated with the subject. For example, if Windows NT/2000/XP is the subject, the rule might require that the version be greater than or equal to 4.00.950 in order for the application to be distributed.

Always Show Icon

Select this option if you want Application Launcher to always display the Application object’s icon regardless of whether or not the workstation meets the requirements defined by the distribution rules.

If you select this option and one or more rules is not met, the icon is displayed in a disabled state. Users can right-click the disabled icon, select *Details*, then select *Requirements* to view a list of all of the distribution requirements that were defined for the application.

Legacy

The Legacy options let you manage system requirements in a mixed environment consisting of ZENworks 7 (or later) versions of Application Launcher and pre-ZENworks 7 Application Launcher versions.

- ◆ **Edit legacy settings:** Only ZENworks 7 (or later) versions of Application Launcher have the ability to process the new grouping and operator (AND/OR) logic used with the distribution rules on this page. To maintain backward compatibility with the pre-ZENworks 7 versions, you must define legacy system requirements.

Click *Legacy > Edit Legacy Settings* to display the Legacy System Requirements dialog box. If you created the Application object in a pre-ZENworks 7 version, the dialog box displays any previously-defined system requirements. If you created the Application object in ZENworks 7, the dialog box is empty. You can add, modify, or remove requirements using the options in the dialog box.

- ◆ **Import legacy settings:** This option copies the legacy system requirements into the *Add rules to control availability of this application* list. If you’ve already added rules to the list, the

legacy settings replace any rules you've added. To import legacy system requirements, click *Legacy > Import Legacy Settings*.

In a mixed-version environment, you need to decide how you want to use the new distribution rules and legacy system requirements together. In general, you have the following options:

- ◆ Define distribution rules only. The ZENworks 7 (or later) version of Application Launcher can read the distribution rules. Pre-ZENworks 7 versions cannot read the distribution rules and therefore are unable to distribute the application.
- ◆ Define distribution rules and legacy system requirements. ZENworks 7 (or later) versions use the distribution rules and pre-ZENworks 7 versions use the legacy system requirements. This method enables you to maintain the legacy system requirements for pre-ZENworks 7 Application Launcher while using the grouping, operator logic, and new rules available beginning with ZENworks 7. If you use this method, you can define the legacy system requirements first and then use the Import Legacy Settings option to populate the distribution rules. After populating the distribution rules, you can modify them to meet your needs.
- ◆ Define legacy system requirements only. Both ZENworks 7 (or later) and pre-ZENworks 7 versions use the legacy system requirements. This solution reduces administrative maintenance of distribution rules and system requirements, but excludes you from using the new grouping, operators, and rules. Be aware that if you define any distribution rules, ZENworks 7 (or later) versions of Application Launcher use the distribution rules, not the legacy system requirements. The ZENworks 7 Application Launcher checks first for the existence of distribution rules; only if no distribution rules exist does it then check for legacy system requirements.

File

This option lets you search for requirements in the list and import requirements into the `.aot` or `.axt` file.

Click *File* > choose one of the following options:

- ◆ **Find:** Searches for requirements that include certain information. For example, you could search for requirements that include “Windows,” “>=,” or “system memory.”
- ◆ **Find next:** Finds the next occurrence of the requirement.
- ◆ **Import:** Imports requirements from another Application object's `.aot` or `.axt` file. The Open dialog box defaults to `*.axt` for its file type display. If you are importing from a `.aot` file, you must change the file type display to `*.aot` or *All files* in order to select the `.aot` file.

Add

Click *Add*, then select one of the following distribution rules:

- ◆ **Applications:** Lets you require a specific application to be present or not be present on the workstation. The application must have been distributed through an Application object. For more information, see [“Applications” on page 541](#).
- ◆ **Client:** Lets you require the workstation to be running (or not running) the Novell Client. For more information, see [“Client” on page 542](#).
- ◆ **Connection speed:** Lets you require a specific network connection speed. For more information, see [“Connection Speed” on page 542](#).

- ◆ **Disk space:** Lets you require a specific amount of free disk space on the workstation. For more information, see [“Disk Space” on page 543](#).
- ◆ **Environment variables:** Lets you require specific environment variables to be present or not present. For more information, see [“Environment Variables” on page 544](#).
- ◆ **File date:** Lets you require a file to be a specific date. For more information, see [“File Date” on page 545](#).
- ◆ **File existence:** Lets you require a file to be present or not present. For more information, see [“File Existence” on page 545](#).
- ◆ **File size:** Lets you require a file to be a specific size. For more information, see [“File Size” on page 546](#).
- ◆ **File version:** Lets you require a file to be a specific version. For more information, see [“File Version” on page 547](#).
- ◆ **New group:** Lets you add a new rules group. For examples of how to use groups, see [“Distribution Rules Examples” on page 540](#).
- ◆ **Memory:** Lets you require a specific amount of memory on the workstation. For more information, see [“Memory” on page 548](#).
- ◆ **Operating system:** Lets you require a specific operating system. For more information, see [“Operating System” on page 548](#).
- ◆ **Processor:** Lets you require a specific processor type. For more information, see [“Processor” on page 549](#).
- ◆ **Registry:** Lets you require specific registry values to be present or not present. For more information, see [“Registry” on page 550](#).
- ◆ **Remote access:** Lets you require that Application Launcher be running in a specific mode, either remote mode or local (LAN) mode. For more information, see [“Remote Access” on page 551](#).
- ◆ **Terminal server:** Lets you require the workstation to be (or not be) a terminal server. For more information, see [“Terminal Server” on page 551](#).

Modify

Select a rule in the list, then click *Modify* to edit the information.

Delete

Select a rule in the list, then click *Delete* to remove the requirement.

Distribution Rules Examples

You can use AND and OR Boolean operators when defining the rules that control the distribution of an application. You can also use groups of rules to further control the distribution. Examples of using AND/OR Boolean operators and groups are provided below.

Simple Rule Using an OR Operator

You want to require a workstation to have either a Pentium III processor (or better) or the Windows 2000 operating system. To do so, use the following rule:

```
Processor type(Processor) >= Pentium III OR OS version(Windows NT/2000/XP) >= 5.00.2195
```

To create this rule:

- 1 On the Distribution Rules page, click the *Add* button, then select *Processor*.
- 2 In the Processor Rule dialog box, select *Greater than or equal to (>=)*, select *Pentium III*, then click *OK* to add the rule to the *Add rules to control availability of this application* list.
- 3 On the Distribution Rules page, click the *Add* button, then select *Operating system*.
- 4 In the Operating System Rule dialog box, select *Windows NT/2000/XP*, select *Greater than or equal to (>=)*, enter 5.00.2195, then click *OK* to add the rule to the *Add rules to control availability of this application* list.

The rule is added to the list with an AND operator.

- 5 In the list, change the AND operator, located in front of the OS Version rule, to an OR operator.

Simple Group

You want to require a workstation to have either a Pentium III processor (or better) or the Windows 2000 operating system with at least 512 MB of memory. To do so, use the following rule:

```
Processor type(Processor) >= Pentium III OR (OS version(Windows NT/2000/XP) >= 5.00.2195 AND System memory(RAM) >= 512)
```

To create this rule:

- 1 On the Distribution Rules page, click the *Add* button, then select *Processor*.
- 2 In the Processor Rule dialog box, select *Greater than or equal to (>=)*, select *Pentium III*, then click *OK* to add the rule to the *Add rules to control availability of this application* list.
- 3 On the Distribution Rules page, click the *Add* button, then select *New group* to add a new group to the *Add rules to control availability of this application* list.

The new group is labeled Group 1, unless you've previously added groups to the list. In addition, it is assigned the AND operator by default.

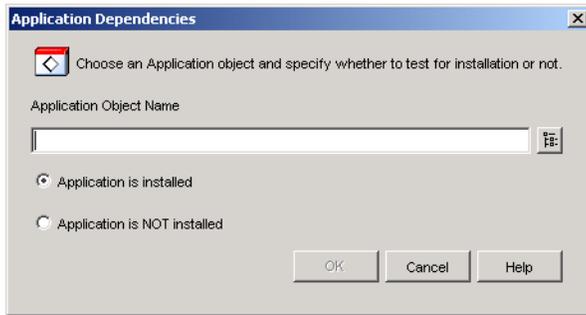
- 4 Locate Group 1 in the list and change the AND operator to an OR operator.
- 5 Next to Group 1 in the list, click the *Add to Group* button, then select *Operating System*.
- 6 In the Operating System Rule dialog box, select *Windows NT/2000/XP*, select *Greater than or equal to (>=)*, enter 5.00.2195, then click *OK* to add the rule under Group 1 in the list.
- 7 Next to Group 1 in the list, click the *Add to Group* button, then select *Memory*.
- 8 In the Memory Rule dialog box, select *Greater than or equal to (>=)*, enter 512, then click *OK* to add the rule under Group 1 in the list.

Applications

The Application Dependencies dialog box (*Distributions* page > *Add* > *Applications*), shown below, bases distribution on whether or not another application is installed on the workstation. The other application must be one that is represented in eDirectory as an Application object.

If you require an application that is not present, the distribution fails. To ensure that the distribution occurs, you can have Application Launcher install the dependent application. For information, see [“Application Dependencies Page” on page 531](#).

Figure 48-26 *Application Dependencies Dialog Box*



Application Object Name

Browse to and select the application whose existence on the workstation Application Launcher will check for.

Application Is Installed

Select this option to indicate that the selected application must already be installed on the workstation.

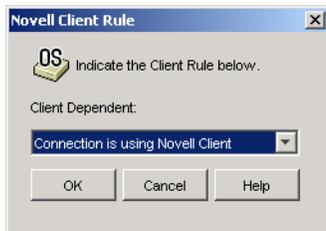
Application Is Not Installed

Select this option to indicate that the selected application must not be installed on the workstation.

Client

The Novell Client Rule dialog box (*Distribution Rules* page > *Add* > *Client*), shown below, lets you base distribution on whether or not the Novell Client is installed on the workstation.

Figure 48-27 *Novell Client Rule Dialog Box*



Client Dependent

Select *Connection is using Novell Client* if you want the application to be available only on workstations that are running the Novell Client.

Select *Connection is not using Novell Client* if you want the application to be available only on workstations that are not running the Novell Client.

Connection Speed

The Connection Speed Rule dialog box (*Distribution Rules* page > *Add* > *Connection speed*) lets you base distribution on the speed of the workstation's network connection.

For example, if you don't want the application to be available on remote workstations whose connection speed is less than 56 Kbps, you could require a connection speed of 56 Kbps or greater.

Figure 48-28 Connection Speed Rule Dialog Box



Speed Is (Conditional Statement)

Select whether the connection speed must be less than (<), less than or equal to (<=), greater than (>), or greater than or equal to (>=) the speed entered in the Kilobits per Second field.

Kilobits per Second (Condition)

Specify the connection speed in kilobits per second.

Disk Space

The Disk Space Rule dialog box (*Distribution Rules* page > *Add* > *Disk space*), shown below, lets you base distribution on the workstation's free disk space. This ensures that Application Launcher does not attempt to distribute an application to a workstation that does not have sufficient free disk space.

Figure 48-29 Disk Space Rule Dialog Box



Disk Space On

Select the disk where you are requiring free space. The list includes the following options:

- ◆ **A through Z:** Allows you to select the disk that is mapped to drive A, B, C, and so on.
- ◆ **Windows system directory drive:** Allows you to select the disk where the Windows system directory (for example, `winnt\system32`) resides.
- ◆ **Windows directory drive:** Allows you to select the disk where the Windows directory (for example, `winnt`) resides.

- ◆ **Temp directory drive:** Allows you to select the disk where the Windows temporary directory (for example, `winnt\temp`) resides.

If you create multiple disk space requirements for the application, Application Launcher adds the requirements together to determine the total disk space requirement. For example, if you specify 20 MB free on the Windows directory drive (which happens to be the C: drive), 10 MB free on the temp directory drive (which is also the C: drive), and 50 MB free on the D: drive, Application Launcher distributes the Application object only if there is 30 MB available on the C: drive and 50 MB free on the D: drive.

Is (Conditional Statement)

Select whether the available disk space must be less than, less than or equal to, equal to, greater than, greater than or equal to the amount listed in the MB field. The corresponding operators are as follows:

Less than: <

Less than or equal to: <=

Equal to: =

Greater than or equal to: >=

Greater than: >

For example, if you want to require at least 50 MB of free disk space, select >= (greater than or equal to) and then enter 50 in the MB field.

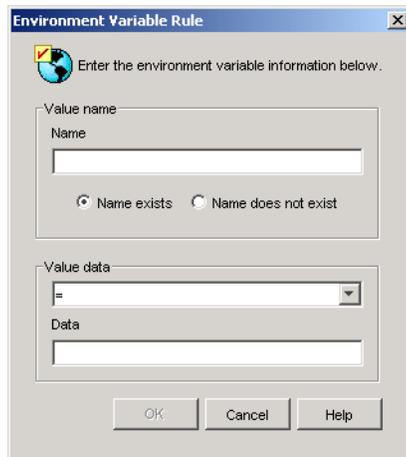
MB (Condition)

Specify the disk space requirement.

Environment Variables

The Environment Variable Rule dialog box (*Distribution Rules page > Add > Environment variables*), shown below, lets you base distribution on whether or not an environment variable containing specific data exists.

Figure 48-30 Environment Variable Rule Dialog Box



Value Name

Specify the name of the environment variable. Select the *Name exists* option to require the variable to already exist on the workstation, or select the *Name does not exist* option to require the variable to not exist on the workstation.

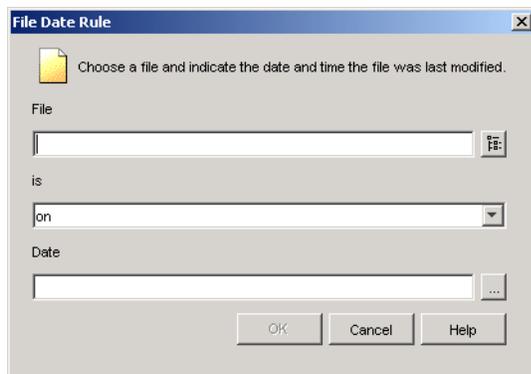
Value Data

Specify whether the variable data must be equal to, not equal to, contain, or not contain the data you enter in the Data field.

File Date

The File Date Rule dialog box (*Distribution Rules* page > *Add* > *File date*), shown below, lets you base distribution on the date of a specific file located on the workstation.

Figure 48-31 File Date Rule Dialog Box



File

Browse to select the file whose date Application Launcher will check.

Is (Conditional Statement)

Select whether the file date must be before, on or before, on, on or after, or after the date entered in the *Date* field. For example, if you want to make sure the application is distributed only if the file is dated 15 January 2000 or older, you would select *On or before* and then select the date.

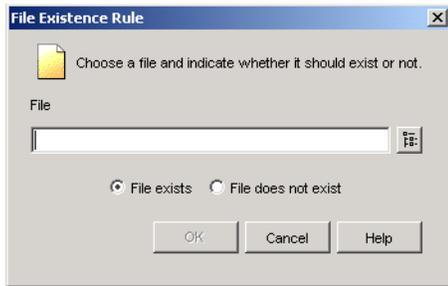
Date (Condition)

Use the Calendar to select the appropriate date.

File Existence

The File Existence Rule dialog box (*Distribution Rules* page > *Add* > *File existence*), shown below, lets you base distribution on whether or not a specific file exists on the workstation.

Figure 48-32 File Existence Rule Dialog Box



File

Browse and select the file that Application Launcher will check for.

File Exists

Select this option to require the file to exist on the workstation.

File Does Not Exist

Select this option to require the file to not exist on the workstation.

File Size

The File Size Rule dialog box (*Distribution Rules* page > *Add* > *File size*), shown below, lets you base distribution on the size of a specific file located on the workstation.

Figure 48-33 File Size Rule Dialog Box



File

Browse to select the file whose size Application Launcher will check.

Is (Conditional Statement)

Select whether the size must be less than, less than or equal to, equal to, greater than, or greater than or equal to the size entered in the Size field. The corresponding operators are as follows:

Less than: <

Less than or equal to: <=

Equal to: =

Greater than or equal to: >=

Greater than: >

For example, if you want to require the file to be at least 200 KB, select >= (greater than or equal to) and then enter 200000 in the Size field.

Size

Specify the file size in kilobytes.

File Version

The File Version Rule dialog box (*Distribution Rules* page > *Add* > *File version*), shown below, lets you base distribution on the version of a specific file located on the workstation.

Figure 48-34 File Version Rule Dialog Box



File

Browse to select the file whose version you want Application Launcher to check.

Is (Conditional Statement)

Select whether the version must be less than, less than or equal to, equal to, greater than, greater than or equal to the version entered in the Version field. The corresponding operators are as follows:

Less than: <

Less than or equal to: <=

Equal to: =

Greater than or equal to: >=

Greater than: >

For example, if you want to require at least version 1.12.2.234, select >= (greater than or equal to) and then enter 1.12.2.234 in the Version field.

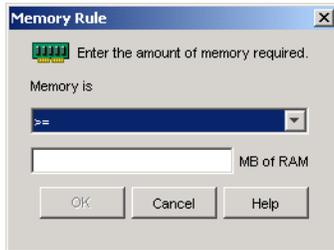
Version (Condition)

Specify the file version.

Memory

The Memory Rule dialog box (*Distribution Rules* page > *Add* > *Memory*), shown below, lets you base distribution on the amount of memory (RAM) required on the workstation.

Figure 48-35 *Memory Rule Dialog Box*



Memory Is (Conditional Statement)

Select whether the total memory must be less than, less than or equal to, equal to, greater than, greater than or equal to the memory entered in the MB of RAM field. The corresponding operators are as follows:

Less than: <

Less than or equal to: <=

Equal to: =

Greater than or equal to: >=

Greater than: >

For example, if you want to require at least 32 MB of free RAM, select >= (greater than or equal to) and then enter 32 in the MB of RAM field.

MB of RAM (Condition)

Specify the required memory amount.

Operating System

The Operating System Rule dialog box (*Distribution Rules* page > *Add* > *Operating system*), shown below, lets you base distribution on the workstation's operating system and version.

Figure 48-36 *Operating System Rule Dialog Box*



Platform Is

Select the operating system (Windows 98 or Windows NT/2000/XP) the workstation must be running.

NOTE: The requirement for an operating system to be defined before an application is available has been removed.

In previous ZENworks versions, an OS platform had to be defined in the System Requirements before an application was available for distribution and launching. This requirement has been removed.

The new behavior uses the following logic: If an application runs only on a specific operating system, define an operating system distribution rule. If an application does not require a specific operating system, there is no need to define a distribution rule. By default, applications without a defined operating system distribution rule are available on all supported platforms (Windows 98, Windows 2000, and Windows XP).

Version

Select whether the version number must be less than, less than or equal to, equal to, greater than, greater than or equal to the number entered in the Version field.

Less than: <

Less than or equal to: <=

Equal to: =

Greater than or equal to: >=

Greater than: >

For example, if you want to require at least version 4.00.950, select >= (greater than or equal to) and then enter 4.00.950 in the Version field.

Processor

The Processor Rule dialog box (*Distribution Rules* page > *Add* > *Processor*), shown below, lets you base distribution on the workstation's processor.

Figure 48-37 Processor Rule Dialog Box



Processor Is

Select whether the processor must be less than, less than or equal to, equal to, greater than, or greater than or equal to the processor type (386, 486, Pentium, Pentium Pro, Pentium II, Pentium III) selected in the processor list.

Less than: <
Less than or equal to: <=
Equal to: =
Greater than or equal to: >=
Greater than: >

For example, if you want to require at least a Pentium II processor, select >= (greater than or equal to) and then select Pentium II.

Registry

The Registry Rule dialog box (*Distribution Rules* page > *Add* > *Registry*), shown below, lets you base distribution on whether or not a specific registry key or value exists.

Figure 48-38 Registry Rule Dialog Box

Registry Rule

Enter a registry key or optionally enter a value and data to test.

Key

Key

Key exists Key does not exist

Value name

Name

Value exists Value does not exist

Value

is

=

Value

String DWORD

OK Cancel Help

Key

Browse and select the registry key you want to check before distribution. Select *Key exists* if the key must exist for the distribution to take place, or select *Key does not exist* if the key cannot exist for the distribution to take place.

Name

If the key has a specific value you want to check for, browse and select the value. Select *Value exists* if the value must exist for the distribution to take place, or select *Value does not exist* if the value cannot exist for the distribution to take place.

Value

If you specified a value in the *Name* field, select an operator (<, <=, =, !=, >=, >) for the value's data, enter the data in the *Value* field, then select whether the value data is a String or DWORD type.

The operator definitions are as follows:

Less than: <

Less than or equal to: <=

Equal to: =

Greater than or equal to: >=

Greater than: >

Remote Access

The Remote Access Rule dialog box (*Distribution Rules* page > *Add* > *Remote access*), shown below, lets you base distribution on whether Application Launcher is in remote mode or local (LAN) mode. The method Application Launcher uses to detect which mode it is in is determined by the Remote Access Detection setting (User/Workstation/container object > *ZENworks* tab > *Launcher Configuration* page > *Add* button > *User* tab).

Figure 48-39 Remote Access Rule Dialog Box



Remote Access Connection Is

Select whether the connection must be a remote access connection or a LAN connection. For example, if you select *Remote access connection*, the application is distributed to a user only if Application Launcher is running in remote mode.

Terminal Server

The Terminal Server Rule dialog box (*Distribution Rules* page > *Add* > *Terminal server*), shown below, lets you base distribution on whether or not the machine is a terminal server (in other words, Windows Terminal Services is active on the machine).

Figure 48-40 Terminal Server Rule Dialog Box



Select *Terminal server required* if you want the application to be distributed only to terminal servers. Select *Terminal server not required* if you want the application to be distributed only to non-terminal servers.

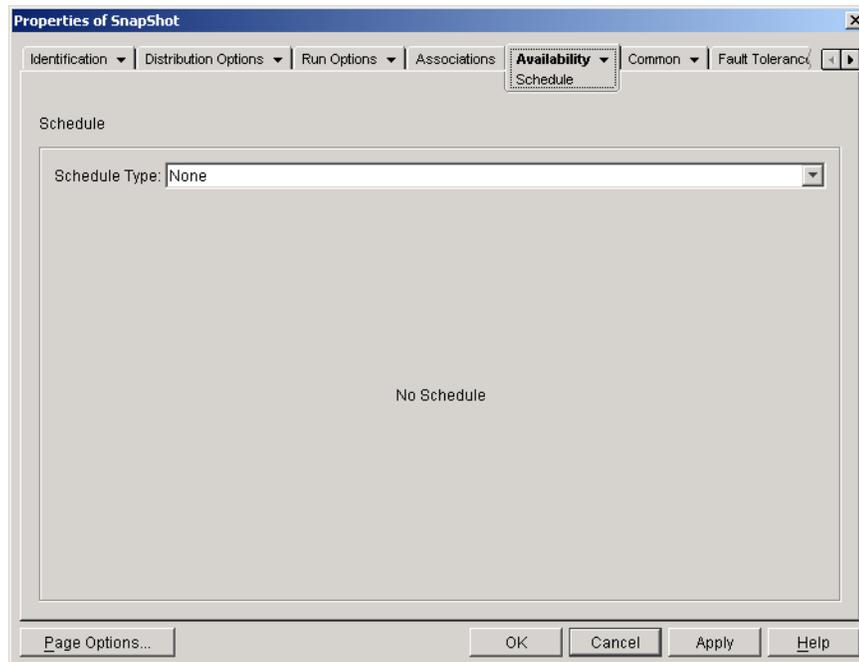
48.5.2 Schedule Page

The Schedule property page is available on Application objects created for all application types (simple, AOT/AXT, MSI, Web, and terminal server).

The Schedule property page, shown below, defines the dates and times when the application is available to users. Application Launcher displays the application icon only during the times defined by the schedule.

The schedule applies regardless of whether or not the application is installed. For example, if a user has not yet installed the application, it is only available for installation at the times specified by the schedule. Likewise, if a user has already installed the application, it is only available for running during the scheduled times.

Figure 48-41 Application Object > Availability Tab > Schedule Page



Schedule Type

Select the type of schedule you want to use. You can choose *None*, *Specified days*, or *Range of days*.

None

Use this option to indicate no schedule. The Application object becomes available to an object as soon as the application's system requirements have been established (*Availability > System Requirements* page) and the application has been associated with the object (*Associations* page).

Specified Days

Use this option to select specific dates when you want the Application object to be available. After selecting this option, fill in the following fields:

- ♦ **Date range:** The Date Range list displays all dates when the Application object is available. To add a date, click *Add* > select the date you want > click *OK* to display it in the list. You cannot select more than 350 specific dates.
- ♦ **Time for selected dates:** Select the availability start time and end time. The times apply to all dates in the Date Range list.

NOTE: The time increments in 5 minute intervals, with the earliest available start time being 00:00 (12:00 a.m.) and the latest end time being 23:55 (11:55 p.m.). This means there is always a 5-minute time period from 11:55 p.m. to 12:00 midnight when the application is unavailable. If you want the application to be available the entire day, you need to use the *Range of days* schedule type. For more information, see [“Range of Days” on page 554](#).

- ♦ **Spread from start time (in minutes):** The *Spread from start time* option spreads out user access times over the number of minutes specified so the application doesn't become available to all users at the same time. If you anticipate all users launching the application as soon as it becomes available and the application is being distributed or run from the network, you can use this option to avoid possible network overload.

For example, if you have a moderate number of users to whom the application is to be distributed (say about 100), you might specify a one-hour (60 minute) block of time (starting at the scheduled start time) to randomly distribute the application: thus all users will gain access to the application some time during the first sixty minutes after the scheduled start time.

If you want to substantially ease the load on your servers caused by the application distribution, or if you have bandwidth concerns, you might want to make the application distribute randomly throughout the time of availability. To spread out access times across the entire time (*Specified days* and *Time for selected dates*) that the application is available, use the total availability time specified for that application in terms of minutes. This will require that you make the maximum time available for each day you specify. For example, if an application is configured for a typical business day in the United States (9 hours per day: 8:00 a.m. to 5:00 p.m.), you calculate the total time of availability for that application like this:

Number of specified hours x 60 minutes per hour = Total availability time per day

Using this equation, the example above would be calculated like this:

9 x 60 (minutes per hour) = 540 minutes of availability

In this example, when you enter 540 minutes in the *Spread from start time* field, the application is distributed randomly for the entire 540 minutes that you have made it available on that scheduled day. Note that this might not be suitable for applications that must be distributed in a timely fashion, such as anti-virus updates. Note also that this is an example only: you can schedule the distribution for any specified amount of time for any day of the week.

Remember that the *Time for selected dates* setting makes the last five minutes of a day unscheduleable, so you need to consider these five minutes if the application schedule ends at 11:55 p.m. for that day.

Range of Days

Use this option to select a range of dates to make the application available. You can also use this option to make applications available only on certain days of the week within a given range of dates. After selecting this option, fill in the following fields:

- ♦ **Date range:** To define the range of days, select a start date and an end date, then select the days (Sunday through Saturday) within the established date range that the application is available. By default, all days are selected; a day is selected when the button appears to be pressed in.
- ♦ **Time for selected range:** Select the availability start time and end time. This option works differently depending on whether the date range includes one day, multiple days, or all seven days. If the date range includes one to six days (but not all seven days), the application is available between the start and end times on those days. For example, if you make the application available on Monday between 8:00 and 5:00, it is available during those hours. However, if the date range includes all seven days, the times are ignored and the application is available every day, 24 hours a day.
- ♦ **Spread from start time (in minutes):** The *Spread from start time* option spreads out user access times over the number of minutes specified so the application doesn't become available to all users at the same time. If you anticipate all users launching the application as soon as it becomes available and the application is being distributed or run from the network, you can use this option to avoid possible network overload.

For example, if you have a moderate number of users to whom the application is to be distributed (say about 100), you might specify a one-hour (60 minute) block of time (starting at the scheduled start time) to randomly distribute the application: thus all users will gain access to the application some time during the first sixty minutes after the scheduled start time.

If you want to ease the load of the application distribution on your servers or if you have bandwidth concerns, you might want to make the application distribute randomly throughout the time of availability. To spread out access times across the entire time (*Date range* and *Time for selected dates*) that the application is available, use the total availability time specified for that application in terms of minutes. For example, if a workstation-associated application is configured for an entire 24-hour, three-shift day, you can calculate the total time of availability for that application like this:

Number of days in date range x *Time of availability per day* = Total availability time

Using this equation, and making sure to convert hours to minutes, the example above would be calculated like this:

7 (days) x 24 (hours) = 168 hours of availability

168 x 60 (minutes per hour) = 10,080 minutes of availability

When you enter 10800 minutes in the *Spread from start time* field, the application is distributed randomly for the entire 10800 minutes that you have made it available. Note that this is not suitable for applications that must be distributed in a timely fashion, such as anti-virus updates.

- ♦ **Use this schedule in GMT for all clients:** The schedule is based on the workstation's time zone. If your network spans different time zones and you schedule an application to run at 1:00 p.m., it runs at 1:00 p.m. in each time zone. You can select this option to have workstations run applications at the same time regardless of their time zones (for example, 1:00 p.m. Rome time and 4:00 a.m. Los Angeles time).

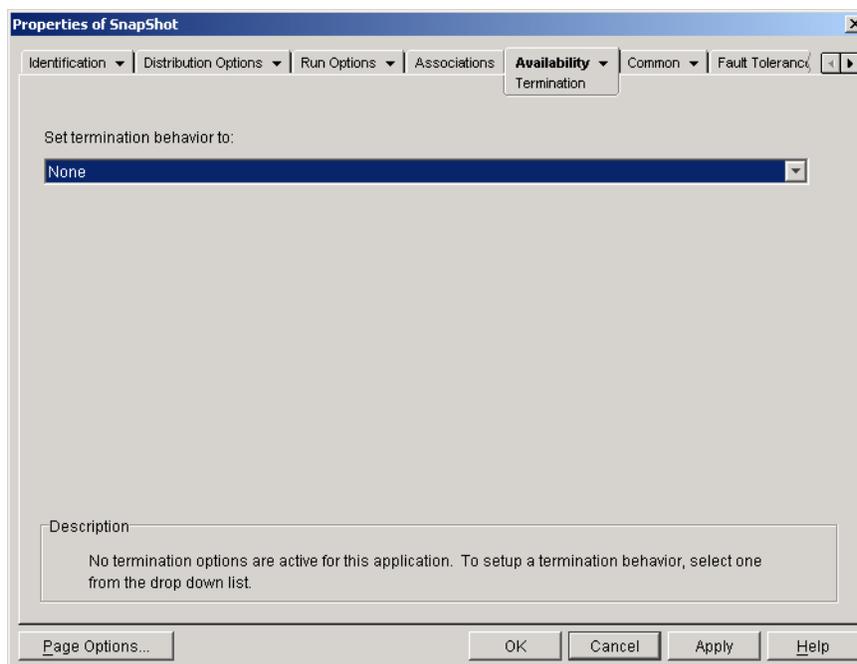
48.5.3 Termination Page

The Termination property page is available on Application objects created for all application types (simple, AOT/AXT, MSI, Web, and terminal server).

The Termination property page, shown below, determines how Application Launcher terminates the application if it becomes unavailable to a user while the user is running the application. An application becomes unavailable to a user when:

- ◆ You disassociate it from the user.
- ◆ The availability schedule expires.
- ◆ The application's distribution rules change and the user's workstation no longer complies with the requirements.

Figure 48-42 Application Object > Availability Tab > Termination Page



Set Termination Behavior To

Select the behavior you want used when the application terminates.

None

Select this option to disable any termination behaviors.

Send Message to Close Application

This option causes Application Launcher to display a message instructing the user to close the application. The message appears periodically until the user closes the application.

In the Basic Closing Message box, select how often you want Application Launcher to display the message. The following default message is used unless you click the Message button and define a custom message:

WARNING! Your access to this application has expired. Please close the application to avoid losing your work.

Send Message to Close Then Prompt to Save Data

This option causes Application Launcher to first display a message instructing the user to close the application. If the user does not close the application, Application Launcher sends a call to the application instructing it to close. If the user has no unsaved data, the application closes immediately. If the user has unsaved data, the application displays a Save dialog box to allow the user to save the data. After the application closes, users cannot launch it again.

IMPORTANT: Some applications display a Save dialog box that includes a *Cancel* button in addition to *Yes* and *No* buttons. The *Cancel* button aborts the close operation and the application remains open. If the application's Save dialog box includes a *Cancel* button and you want to ensure that the application is closed, you must use the *Send message to close, Prompt to save, Then Force to close* option.

After selecting this option, fill in the following fields:

- ◆ **Warning message (optional):** Select the *Send warning* option if you want to warn the user to close the application, then specify the number of times to display the warning message and the interval between messages. The following default message is used unless you click the *Message* button and define a custom message:

WARNING! Your access to this application has expired. Please close the application to avoid losing your work.

- ◆ **Prompt closing message:** This message is displayed if the user does not close the application after receiving the warning message the specified number of times, or if the warning message option is not enabled.

In the Prompt Closing Message box, specify how often you want to prompt the user to close the application. If, after receiving the prompt, the user does not close the application, Application Launcher attempts to close the application. The following default message is used unless you click the *Message* button and define a custom message:

WARNING! Your access to this application has expired and the application will close shortly. Please save your work and exit the application.

Send Message to Close, Prompt to Save, Then Force Close

This option causes Application Launcher to first display a message instructing the user to close the application. If the user does not close the application, Application Launcher sends a call to the application instructing it to close. If the user has no unsaved data, the application closes immediately. If the user has unsaved data, the application displays a Save dialog box to allow the user to save the data. Regardless of the option (*Yes, No, Cancel*) the user selects, the application is closed.

After selecting this option, fill in the following fields:

- ◆ **Warning message (optional):** Select the *Send warning* option if you want to warn the user to close the application, then specify the number of times to display the warning and the interval between warnings. The following default message is used unless you click the *Message* button and define a custom message:

WARNING! Your access to this application has expired. Please close the application to avoid losing your work.

- ◆ **Prompt closing message:** This message is displayed if the user does not close the application after receiving the warning message the specified number of times, or if the warning message option is not enabled.

In the Prompt Closing Message box, specify how often you want to prompt the user to close the application. If, after receiving the prompt, the user does not close the application, Application Launcher attempts to close the application. The following default message is used unless you click the *Message* button and define a custom message:

WARNING! Your access to this application has expired and the application will close shortly. Please save your work and exit the application.

- ◆ **Application termination:** This message is displayed after Application Launcher closes the application. If you don't want to use the default termination message, click the *Message* button and define a custom message. The default message is:

The application you were running has expired. For additional usage, please contact your system administrator.

Send Message to Close Then Force Close with Explanation

This option prompts the user to close the application. If the user does not, Application Launcher closes the application automatically and displays a termination message.

After selecting this option, fill in the following fields:

- ◆ **Warning message (optional):** Select the *Send warning* option if you want to warn the user to close the application, then specify the number of times to display the warning and the interval between warnings. If the user does not close the application, Application Launcher closes the application automatically after the last warning message. The following default message is used unless you click the *Message* button and define a custom message:

WARNING! Your access to this application has expired. Please close the application to avoid losing your work.

- ◆ **Application termination:** This message is displayed after Application Launcher closes the application. If you don't want to use the default termination message, click the *Message* button and define a custom message. The default message is:

The application you were running has expired. For additional usage, please contact your system administrator.

48.6 Common Tab

The Common tab includes the following pages:

- ◆ [Section 48.6.1, "Macros Page," on page 558](#)
- ◆ [Section 48.6.2, "Drives/Ports Page," on page 560](#)
- ◆ [Section 48.6.3, "File Rights Page," on page 563](#)
- ◆ [Section 48.6.4, "Reporting Page," on page 565](#)
- ◆ [Section 48.6.5, "Imaging Page," on page 568](#)
- ◆ [Section 48.6.6, "Sources Page," on page 571](#)
- ◆ [Section 48.6.7, "Uninstall Page," on page 572](#)

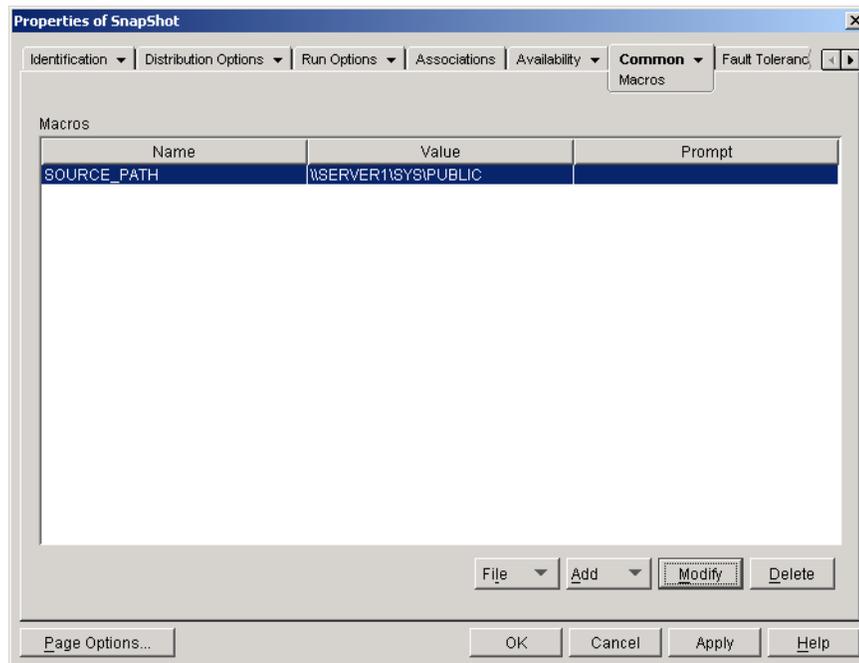
- ◆ [Section 48.6.8, “Uninstall Scripts Page,” on page 575](#)

48.6.1 Macros Page

The Macros property page is available on Application objects created for all application types (simple, AOT/AXT, MSI, Web, and terminal server).

The Macros property page, shown below, displays the macros that have been defined for use with the Application object. For information about the Application object fields that support macros, see [Section 49.8, “Application Object Properties Where Macros Can Be Used,” on page 605](#)

Figure 48-43 *Application Object > Common Tab > Macros Page*



These macros, referred to as Application object macros, are simply variable names that are associated with values, typically string values.

For example, when you create an Application object using an .aot or .axt file, a SOURCE_PATH macro is automatically added to the list. This macro defines the location of the source files from which the application is installed. Many times, you must enter the source path when defining other properties for the Application object. By creating a SOURCE_PATH macro, you only need to enter %SOURCE_PATH% rather than the actual source path.

IMPORTANT: When using a macro for any of these properties, you must enclose the macro name in % characters (for example, %SOURCE_PATH%).

Macros

The Macros list displays all macros that have been defined for use with the Application object.

File

This option lets you search for specific macro information and import macros from other Application objects.

Click *File*, then choose one of the following options:

- ◆ **Find:** Searches for specific information in the *Name*, *Value*, and *Prompt* fields.
- ◆ **Find next:** Finds the next occurrence of the information.
- ◆ **Import:** Imports macros from another Application object's `.aot` or `.axt` file. The Open dialog box defaults to `*.axt` for its file type display. If you are importing from an `.aot` file, you must change the file type display to `*.aot` or `All files` in order to select the `.aot` file.

Add

Use the Add button to define an Application object macro and add it to the Macros list. You can add either a String Value macro or a Prompted macro.

String Value

A String Value macro is used to create a variable that has a string value associated with it. This lets you use the variable in multiple locations while defining the value in one location.

To create a String Value macro, click *Add > String value* to display the Edit String Value dialog box. In the Value Name box, type a name for the variable, then type the value for the variable in the Value Data box. Click *OK* to add the macro to the Macros list.

You can place macros within macros. In the following example, a special Windows macro, `*WINDISK`, is used when defining the `TARGET_PATH` macro:

```
Value Name: TARGET_PATH
Value Data: %*WINDISK%\Program Files
```

When embedding another macro in the string, you must enclose the macro in `%` characters, as shown in the above example. You can embed special Windows macros, eDirectory attribute macros, environment variable macros, login script macros, language variable macros, and other Application object macros. For information about these types of macros, see [Chapter 49, "Reference: Macros," on page 593](#).

Prompted

A Prompted macro is used to create a macro that prompts the user to accept a default value (either a drive or string) or enter a new value. There are two types of Prompted macros: *Prompted drive* macros and *Prompted string* macros. Because Windows Installer, and not Application Launcher, controls the installation of MSI applications, *Prompted* macros do not work with MSI applications.

To create a *Prompted drive* macro, click *Add > Prompted > Drive* to display the Prompted Drive Macros dialog box. Fill in the fields as described below:

- ◆ **Macro name:** Give the macro any name you want. Do not use spaces.
- ◆ **Prompt text:** Specify the text you want presented to users. For example, "This application will be installed to your C: drive. Enter a different drive letter if desired:"
- ◆ **Default value:** Select the drive that is used if the user does not enter another drive letter.

- ◆ **Minimum disk space in MB:** Specify the minimum amount of free disk space required to install the application.

To create a *Prompted string* macro, click *Add > Prompted > String* to display the Prompted String Macros dialog box. Fill in the fields as described below:

- ◆ **Macro name:** Give the macro any name you want. Do not use spaces.
- ◆ **Prompt text:** Specify the text you want presented to users.
- ◆ **Default value:** Select the value that is used if the user does not enter another value in response to the prompt text.
- ◆ **Minimum disk space in MB:** Specify the minimum amount of free disk space required to install the application.
- ◆ **Maximum string length in chars:** Specify the maximum number of characters the user can enter as a response to the prompt.

Modify

Select the macro whose information you want to modify, then click *Modify*.

Delete

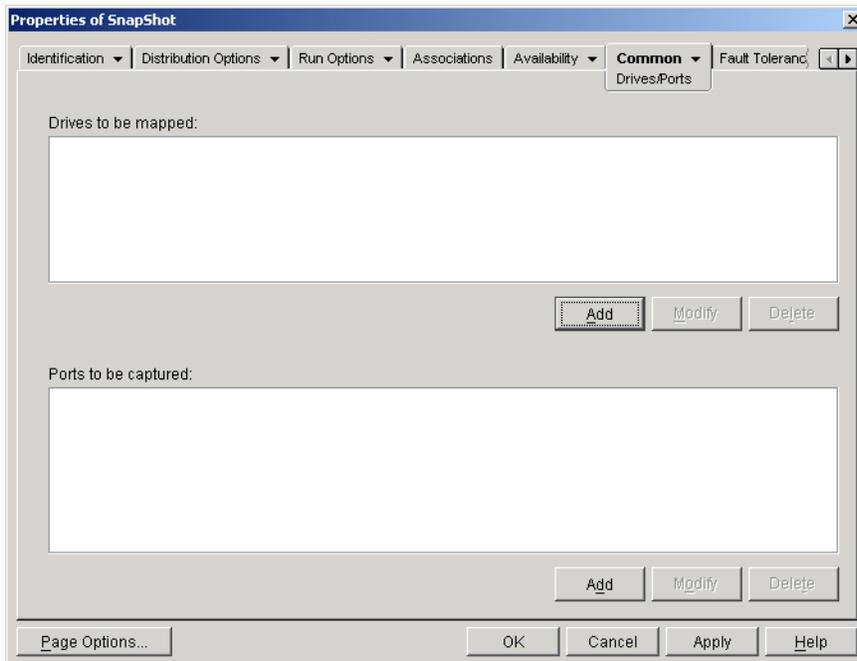
Select the macro you want to delete, then click *Delete*.

48.6.2 Drives/Ports Page

The Drives/Ports property page is available only on Application objects created for simple applications, AOT/AXT applications, and MSI applications. It is not available on Application objects created for Web applications and terminal server applications.

The Drives/Ports property page, shown below, determines the drive mappings and port captures for the application. Drive mappings and port captures are performed by the workstation's network clients, not by Application Launcher. This requires that the workstation have the network clients (for example, the Novell Client and the Microsoft Client) needed to perform the mappings and captures.

Figure 48-44 Application Object > Common Tab > Drives/ports Page



For example, suppose you've set up a database application to run from drive W:. To ensure that W is mapped to the location of the application, you could map drive W: to the server, volume, and directory where the application exists. When Application Launcher runs the application, it establishes the drive mapping according to the conditions defined when setting up the drive mapping.

Or, maybe users need access to a network location to store files created with a word processor. If you map a drive, the drive mapping becomes active as soon as the application runs.

By default, drive mappings and port captures are released when the user closes the application. If several launched applications use the same drive mappings or port captures, the drive mappings and port captures aren't released until the last application terminates. If you don't want drive mappings and port captures released, you must deselect the *Clean up network resources* option (*Run Options > Environment* page).

Drives to Be Mapped

The *Drives to be mapped* list displays all drives that are mapped before the application is launched.

Add

Click *Add* to display the Drive to Be Mapped dialog box. Fill in the following fields:

- ◆ **Root:** Select this option to treat the path as the root of the drive.
- ◆ **Option:** Select the type of drive to map. The available types are:
 - ◆ **DRIVE:** Select this option to map to a drive letter (A through Z).
 - ◆ **S1:** Select this option to assign the drive to the first available search drive.
 - ◆ **S16:** Select this option to assign the drive to the last search drive.

NOTE: Starting with ZENworks for Desktops 4.x, selecting the S1 or S16 option does not affect the system's global path. Only applications started by this NAL process uses this search path; applications external to the NAL process cannot use this search path.

- ◆ **Drive:** Assign a drive letter. The options are drives A through Z, or the next available drive.
- ◆ **Path:** Enter, or browse and select, the path you want mapped to the selected drive. If you enter the path, use the following syntax:

```
server\volume:path  
\\server\volume\path  
volume_object_name:path  
directory_map_object_name:path  
driveletter:\path
```

You can also use a macro in this field. For information about macros, see [Chapter 49, "Reference: Macros,"](#) on page 593.

- ◆ **Map if drive mapping doesn't exist or same as existing:** Select this option if you want Application Launcher to map the drive if 1) it doesn't exist already or 2) it is the same as an existing drive mapping.
- ◆ **Overwrite any existing drive mapping:** Select this option if you want Application Launcher to replace any mapping that uses the requested drive letter. For example, if S: is mapped to \\server1\sys\public and you've specified the new S: drive mapping as \\server1\apps\timecard, the end result is that drive S: is mapped to \\server1\apps\timecard.

To identify this option in the *Drives to be mapped* list, the word FORCE is added to the mapping information.
- ◆ **Use the existing drive mapping if already mapped:** Select this option to use the existing drive mapping even if it is not the same as the path requested. For example, if S: is mapped to \\server1\sys\public and you've specified the new S: drive mapping as \\server1\apps\timecard, the end result is that drive S: is still mapped to \\server1\sys\public.

To identify this option in the *Drives to be mapped* list, the word REUSE is added to the mapping information.

Modify

Select a drive mapping in the *Drives to be mapped* list, then click *Modify* to display the Drive to be Mapped dialog box. Change the drive mapping information as needed. See the field descriptions above or click the *Help* button in the dialog box.

Delete

Select a drive mapping in the *Drives to be mapped* list, then click *Delete* to remove the mapped drive.

Ports to Be Captured

The *Ports to be captured* list displays all ports that are captured before the application is launched.

Add

Click *Add* to display the Ports to Be Captured dialog box. Fill in the following fields:

- ♦ **Port:** Select an LPT port to assign to the printer or queue.
- ♦ **Printer or queue:** Browse to select a printer or queue.
You can also use a macro in this field. For information about macros, see [Chapter 49, “Reference: Macros,”](#) on page 593.
- ♦ **Capture flags:** You can use this option to override notification, banner, and form feed settings defined on the user’s workstation. Whenever a user prints from the application, the settings you defined are used.

In the *Capture flags* box, select the *Override workstation setting* box for the corresponding *Capture* setting you want to override. After you do this, the corresponding *Capture* setting becomes active. Select the setting to turn it on. Leave the setting deselected to turn it off.

For example, if you select the *Override workstation setting* box for *Notify* and also select the *Capture setting* box for *Notify*, when the user prints a document, the Novell Client displays a Notify alert panel to indicate the completion of the print job, no matter which Capture settings are on the user’s workstation. However, if you select the *Override workstation setting* box for *Notify* but do not select the *Capture setting* box for *Notify*, when the user prints a document, the Novell Client does not display a Notify alert panel to indicate the completion of the print job, no matter which Capture settings are used on the user’s workstation.

Modify

Select a port capture in the *Ports to be captured* list, then click *Modify* to display the Ports to be Captured dialog box. Change the capture information as needed. See the field descriptions above or click the *Help* button in the dialog box.

Delete

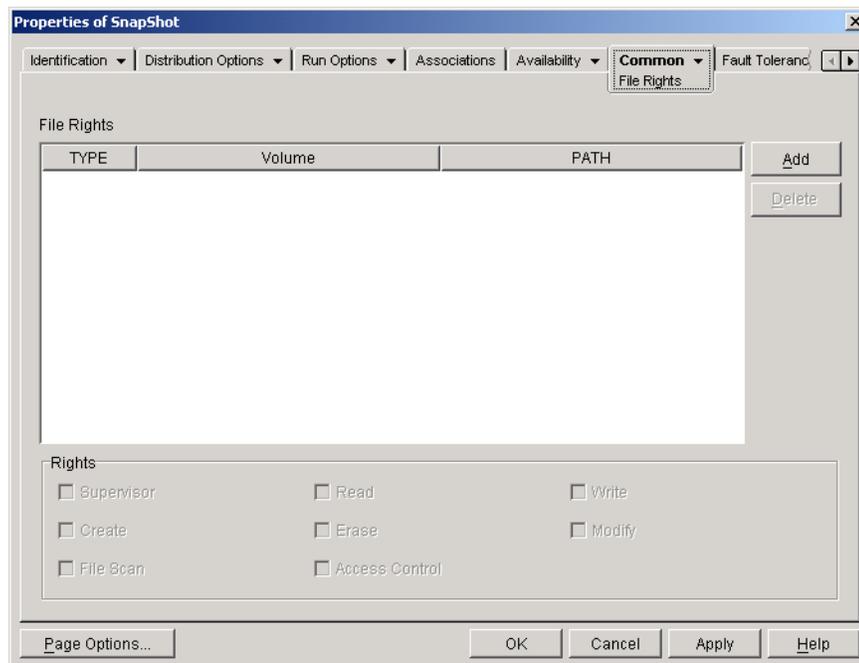
Select a port capture in the *Ports to be captured* list, then click *Delete* to remove the captured port.

48.6.3 File Rights Page

The File Rights property page is available on Application objects created for simple applications, AOT/AXT applications, and MSI applications only. It is not available on Application objects created for Web applications and terminal server applications.

The File Rights property page, shown below, specifies the file, directory, and volume rights a user must have to run the application. This page applies only to NetWare file system rights. Microsoft Windows file system rights cannot be assigned through eDirectory.

Figure 48-45 Application Object > Common Tab > File Rights Page



A user receives these NetWare rights when his or her User object is associated with the Application object, or when a Group, Organizational Unit, Organization, or Country object of which the user is a member is associated with the Application object. The rights are removed from the user when the object is no longer associated with the Application object.

The file rights you set up are not dependent on whether the user is actually using the application. The user has continual file rights from the time the Application object is associated with the user until the time the association is removed.

If two separate Application objects give file rights to the same file, directory, or volume and the user is unassociated with only one of them, then the user loses all rights even though the user might still be associated with the other Application object.

File Rights

The *File rights* list displays all files, directories, and volumes to which rights are given. When you select a file, directory, or volume, the rights that have been granted are displayed in the Rights box. You can grant or remove rights by selecting or deselecting the appropriate check boxes.

Add

Click *Add*, browse for and select the file, directory, or volume you want to add, then click *OK* to display it in the list.

Delete

Select the file, directory, or volume you want to delete from the list, then click *Delete*. Any users associated with the Application object lose rights to the file, directory, or volume.

Rights

Select the file, directory, or volume (in the *File rights* list) for which you are establishing users' rights, then select the appropriate check boxes.

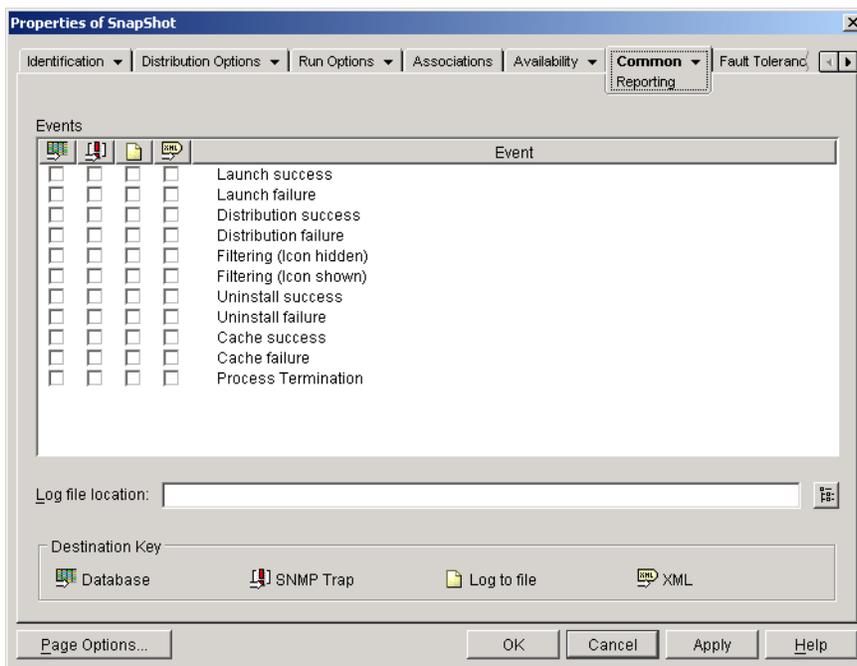
48.6.4 Reporting Page

The Reporting property page is available on Application objects created for all application types (simple, AOT/AXT, MSI, Web, and terminal server).

This section provides information about the fields located on the Reporting property page. For complete instructions on setting up reporting, see [“Reporting Application Events” on page 441](#).

Use the Reporting property page, shown below, to enable Application Launcher to report event information for the application. You can define the events you want Application Launcher to report and also define the destination where you want to send the event reports.

Figure 48-46 Application Object > Common Tab > Reporting Page



Reports contain the following information: event type, date and time, user distinguished name, workstation distinguished name, workstation address, application distinguished name, application global unique identifier (GUID), application version string, and event strings.

Events

Select the events you want reported and the destinations where you want them reported. If necessary, you can send an event report to multiple destinations. All events are described in the following table:

Table 48-3 Reporting Events

Event	Description
Launch Success	Occurs when a user double-clicks the Application object and Application Launcher successfully starts the application.
Launch Failure	Occurs when a user double-clicks the Application object and Application Launcher cannot start the application.
Distribution Success	Occurs when Application Launcher successfully modifies the workstation to support the application. Modifications include installation of files, changing of settings (registry, INI, etc.), creation of shortcuts, and so forth.
Distribution Failure	Occurs when Application Launcher cannot modify the workstation to support the application. Modifications include installation of files, changing of settings (registry, INI, etc.), creation of shortcuts, and so forth.
Uninstall Success	Occurs when Application Launcher successfully uninstalls the application from the workstation.
Uninstall Failure	Occurs when Application Launcher cannot uninstall the application from the workstation.
Cache Success	Occurs when Application Launcher successfully caches the application to the workstation.
Cache Failure	Occurs when Application Launcher cannot cache the application to the workstation.
Filter (Icon Hidden)	Occurs when Application Launcher cannot display an Application object on a workstation because the workstation does not meet one or more of the Application object's system requirements (Application object > Availability tab > System Requirements page) and the system requirement's Show Application Icon Even If Criteria Are Not Met option is set to False. The Application object's icon is hidden, or not shown, on the workstation.
Filter (Icon Shown)	Occurs when Application Launcher can only display a disabled (dimmed) Application object on a workstation. This occurs because the workstation does not meet one or more of the system requirements (Application object > Availability tab > System Requirements page) and the system requirement's Show Application Icon Even If Criteria Are Not Met option is set to True. Users can right-click the disabled icon and click Details to see what system requirements are not being met.
Process Termination	Occurs when a user or Application Launcher terminates the application.

All destinations are described in the following table:

Table 48-4 Reporting Destinations

Destination	Description
 Database	<p>Application Launcher can write events to any ODBC-compliant database (for example, the Sybase database included with ZENworks). To use a database, you must also create a ZENworks Database object to use for Application Management reporting. This cannot be the same database object used for Inventory reporting.</p> <ul style="list-style-type: none">◆ If necessary, create a Service Location Package. Associate the Service Location Package with the containers where the users or workstations reside whose reports you want sent to the database.◆ Enable the ZENworks Database policy in the Service Location Package.◆ Associate the ZENworks Database policy with the ZENworks Database object.◆ Make sure users have the appropriate ODBC database driver installed and configured.◆ After you've set up database reporting, you can use one of the predefined reports to see information about specific application events. To access the reports, right-click the ZENworks Database object you created for Application Management reporting > click <i>Reporting</i>.
 SNMP Traps	<p>Application Launcher can send SNMP traps to any SNMP management console. To use SNMP traps, you must also enable an SNMP Trap Target Policy in a Service Location Package. It might be necessary to first create the Service Location Package.</p> <ul style="list-style-type: none">◆ Associate the Service Location Package with the containers where the users or workstations reside that you want the SNMP Trap Target Policy applied to.◆ Add the SNMP Trap Targets (IP addresses) for the locations you want the traps sent.◆ Have a management console that displays the SNMP traps.
 Text Log File	<p>Application Launcher can write events to a text log file. You use the Log File Location field to specify the location of the log file.</p>
 XML	<p>Application Launcher can send events, as XML data, to a URL using the standard HTTP or HTTPS protocol. XML reporting is the recommended method if you want Application Launcher to report events for users located outside of your firewall.</p> <p>To use XML, you must also:</p> <ul style="list-style-type: none">◆ Enable an XML URL policy in a Service Location Package. It might be necessary to first create the Service Location Package.◆ Associate the Service Location Package with the containers where the users or workstations reside whose reports you want sent as XML data.◆ Make sure you've set up the XML processing mechanism and data storage mechanism. For example, if you are using the Application Management Reporting servlet and the Sybase database, make sure that each are set up and configured properly.

Log File Location

If you selected a log file as the destination for any of the event reports, enter (or browse to and select) the location and name for the log file. You can specify a local drive or a network drive. If you

enter a location that does not exist, Application Launcher creates it. Do not use extended characters in the path; extended characters are not supported.

If you want Application Launcher to log events to a file in a common network location, you need to establish the network directory and provide users with Read and Write rights to files in the directory. Because log file names are established on a per-application basis, you can have individual log files for each application (by specifying a different log file name for each Application object) or one log file for all applications (by specifying the same log file name for all Application objects).

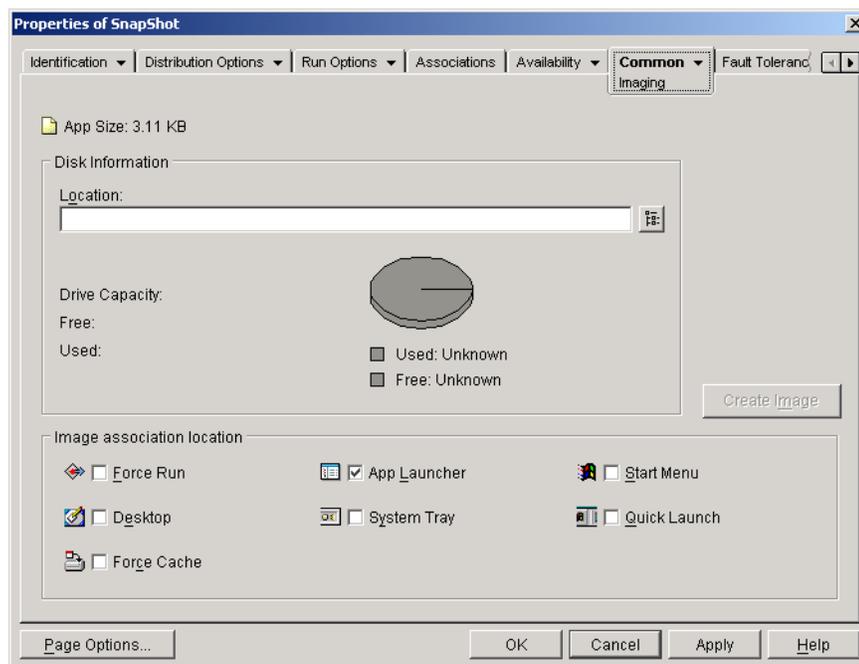
IMPORTANT: You should use log file reporting only if users have a network client such as the Novell Client or Microsoft Client that enables writing to a NetWare, Linux, or Windows server, or if you specify a local drive as the log file location.

48.6.5 Imaging Page

The Imaging property page is available only on Application objects created for simple applications, AOT/AXT applications, and MSI applications. It is not available on Application objects created for Web applications and terminal server applications.

The Imaging property page, shown below, lets you create an image file for the Application object. You can use the image file as an add-on image to a base image file.

Figure 48-47 *Application Object > Common Tab > Imaging Page*



If the application includes any dependent applications (*Application object > Run Options tab > Application Dependencies page*), the dependent applications must also be included as add-on images if they have not already been distributed to the user's workstation.

App Size

This field displays the size of the of the Application object. This includes all Application object information and application source files (as listed on the Application Files page). This is the size of the image file if you do not use a compression option.

When you image an MSI Application object, ConsoleOne is unaware of the files referenced or included in the `.msi` file. Therefore, it includes the `.msi` file and all files and folders at the same level or below. This ensures that all related files are included with the `.msi` file, but might also result in non-related files being included. To avoid this situation, make sure the folder where the `.msi` file is located contains only the `.msi` file and its related files and folders.

Disk Information

These fields let you specify a location and filename for the image and ensure that the disk to which you are saving the image has sufficient free space.

Location

Specify the location and filename for the image. You might want to save it in the same location as the base image file. The filename must have a `.zmg` file extension.

Drive Capacity, Free, and Used

After you've entered a location for the file, these fields display the amount of space on the disk, the amount of available free disk space, and the amount of used disk space. Verify that the Application object fits in the amount of free disk space.

In addition to the image location, the NAL cache directory located on the workstation's Windows drive (typically `c:\nalcache`) is used temporarily to store the application's source files (`.fil` files) and the Application object's settings (`.aot/.axt` file). The size of the source files varies; the size of the settings file is typically less than 200 KB. Make sure that the workstation's Windows drive has sufficient disk space for these files.

Image Association Location

Specify the association characteristics for the Application object. These are enforced when the image is applied to the workstation.

Force Run

Automatically runs the application. With a user-associated application, the application is run immediately after Application Launcher starts. With a workstation-associated application, the application is run immediately after the workstation starts up (initial startup or reboot).

You can use the *Force run* option in conjunction with several other settings to achieve unique behaviors. For example, if you use the *Force run* option with the *Run application once* option (*Run Options* tab > *Applications* page), as soon as the Application object is distributed it runs one time and is removed from the workstation. Or, suppose that you want to run the Application object immediately one time at a predetermined time. If so, select *Force run*, select the *Run application once* option on the Application page (Run Options tab), and define a schedule using the Schedule page (Availability tab).

If you want to force run several applications in a specific order, mark each of them as *Force run*. Then give each Application object a numeric order by using the *Determine force run order* option on the Icon page (Identification tab).

App Launcher

Adds the Application object's icon to the Application Window, the Application Explorer window, and the Application Browser window.

Start Menu

If the workstation is running Application Explorer, this option adds the Application object to the Windows Start menu. The Application object is added to the top of the Start menu, unless you assign the Application object to a folder and use the folder structure on the Start menu. See the Application object's Folders page (Identification tab).

Desktop

If the workstation is running Application Explorer, this option displays the Application object's icon on the Windows desktop.

System Tray

If the workstation is running Application Explorer, this option displays the Application object's icon in the Windows system tray.

Quick Launch

Displays the Application object's icon on the Windows Quick Launch bar.

Force Cache

Forces the application source files and other files required for installation to be copied to the workstation's cache directory. The user can then install or repair the application while disconnected from eDirectory. The files are compressed to save space on the workstation's local drive.

Create Image

After you've defined the image location and selected the image associations, click the *Create Image* button to create the image. A dialog box appears that lets you select from the following compression options:

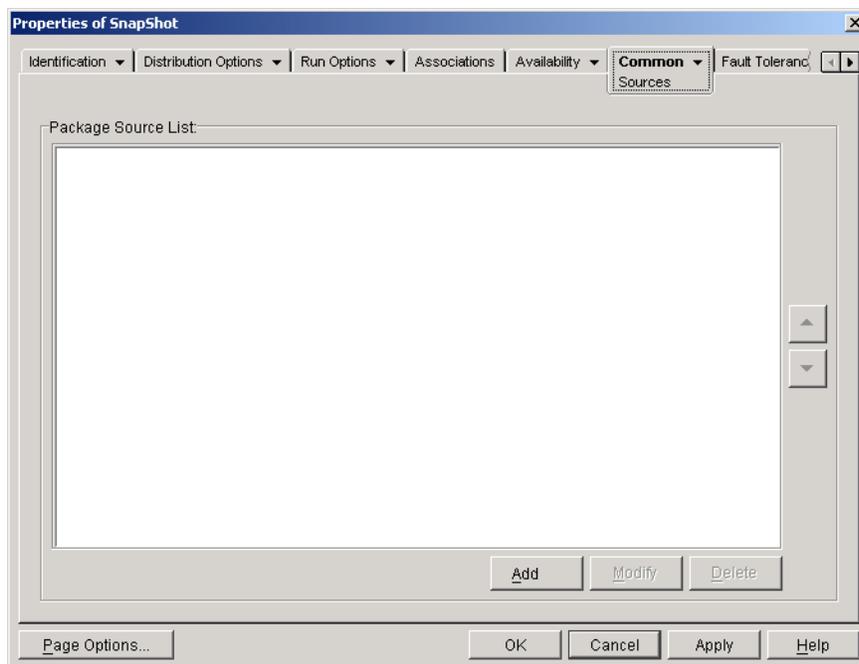
- ◆ **None:** Creates the image file but does not compress it. This results in the largest image size produced by any of the compression options.
- ◆ **Optimize compression time:** Applies the lowest level of compression. This minimizes the time required to compress the image file, but results in a larger file size than could be achieved by using the *Balance compression time and image size* option or *Optimize image size* option.
- ◆ **Balance compression time and image size:** Applies a medium level of compression in order to balance the amount of time required to compress the file against the size of the image file.
- ◆ **Optimize image size:** Applies the highest level of compression. This decreases the image file size but increases the time required to compress the image file as compared to the *Optimize compression time* option and *Balance compression time and image size* option.

48.6.6 Sources Page

The Sources property page is available on Application objects created for simple applications, AOT/AXT applications, and MSI applications only. It is not available on Application objects created for Web applications and terminal server applications.

The Sources property page, shown below, specifies additional network locations that contain installation packages from which Application Launcher can distribute the application. If the Application object was created from an .aot or .axt file, the installation packages must be snAppShot packages (.aot, .axt, and .fil files). If the Application object was created from an .msi file, the installation packages must be Microsoft Windows Installer packages (.msi files). The source list is used only if the original source location is unavailable, in which case Application Launcher begins with the first location in the list and continues down the list until an accessible package is found.

Figure 48-48 Application Object > Common Tab > Sources Page



Before you can define additional source locations, you need to make sure to copy the original package to the new source locations. For snAppShot packages, copy the .aot, .axt, .fil, and .txt files. For Windows Installer packages, copy the administrative installation folder and all subfolders.

With Application objects that use an .msi file rather than an .aot or .axt file, this source list can be extremely important. Because Windows Installer supports on-demand installations (installation of files as the user requests certain features or components), the source list can provide source resiliency for installations that occur after the initial installation. If the original installation source is no longer available and no source list has been established, the on-demand installation fails. However, if you create additional source locations and define them in the source list, the on-demand installation has a better chance of succeeding.

IMPORTANT: If you have used the SOURCE_PATH macro (Common > Macros page) in any of the Application object's property fields, you should verify that you used all uppercase letters and no lowercase letters. If the original source location is referenced through the SOURCE_PATH macro and Application Launcher cannot access the location, it uses the package sources in this list only if the SOURCE_PATH macro is all uppercase.

Package Source List

The *Package source* list displays the additional package source locations that have been defined for the application. When the application is distributed to a workstation, Application Launcher uses these locations, in the order listed, if the original source location is unavailable.

When distributing applications that use the Windows Installer, Application Launcher passes the source list to the Windows Installer, which then uses the list for the locations from which to install the application files.

Add

To add a source location to the list, click *Add*, browse to and select the directory that contains the installation package, then click *OK*.

Delete

To remove a source location from the list, select the location > click *Delete*.

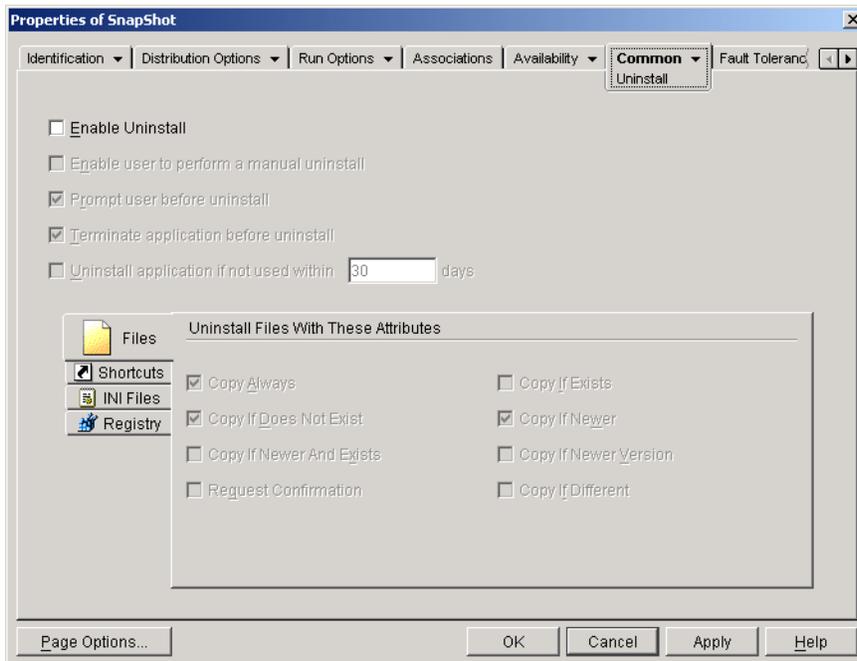
48.6.7 Uninstall Page

The Uninstall property page is available on Application objects created for all application types (simple, AOT/AXT, MSI, Web, and terminal server).

This section provides information about the fields located on the Uninstall property page. For complete instructions on uninstalling applications, see [“Uninstalling Applications” on page 435](#).

The Uninstall property page, shown below, enables the application to be uninstalled and determines the behaviors associated with uninstalling the application.

Figure 48-49 Application Object > Common Tab > Uninstall Page



When Application Launcher distributes an application to a workstation, it creates a cache directory on the workstation that contains the information required to successfully uninstall the application. However, when Application Launcher distributes an application to a terminal server, it does not create a cache directory, which means that it does not have the information to uninstall the application from the terminal server. Therefore, the *Uninstall* options on this page do not apply when the application is installed on a terminal server. If you no longer want the application on the terminal server, you must manually uninstall it.

Enable Uninstall

Select this option to enable the application to be removed from the workstation. The other uninstall options are available only if this option is enabled.

Enable User to Perform a Manual Uninstall

Select this option to enable users to remove the application from their workstations. If this option is not enabled, only you or other administrators can remove applications.

Prompt User Before Uninstall

Select this option to prompt users before removing the application from their workstations. If users answer “No” to the prompt, the application is not removed.

Terminate Application Before Uninstall

Select this option to ensure that Application Launcher terminates the application (if the application is running) before it begins uninstalling the application files.

Uninstall Applications If Not Used Within XX Days

Select this option to automatically remove the application if the user has not run it within the specified number of days (the default is 30).

Files

Click *Files*, then select the attributes you want to use to determine whether or not a file is removed. The attributes correspond to the attributes listed on the Application Files page (Distribution Options tab). The default selections (*Copy always*, *Copy if does not exist*, *Copy if newer*) ensure that a file is removed only if it was installed when the application was installed and not at an earlier time by another application that also needs the file.

Application Launcher does not uninstall files that are copied to the workstation using wildcards. For example, assume you copy `\\server\vol\public\zenworks\sysapps*.*` to `c:\winnt\system32`. When you uninstall the application, its files are not removed from `c:\winnt\system32`. To perform an uninstall, Application Launcher simply reverses the install process, which means that if it uninstalled the files in the previous example, it would do so by doing a wildcard delete (delete `c:\winnt\system32*.*`). Because this could have undesirable results, Application Launcher has been designed to not uninstall files that are copied using wildcards.

Shortcuts

Click *Shortcuts*, then select the attributes you want to use to determine whether or not the application's shortcuts are removed. The attributes correspond to the attributes listed on the Shortcuts/Icons page (Distribution Options tab).

INI Files

Click INI Files, then select the attributes you want to use to determine whether or not an INI setting is removed. The attributes correspond to the attributes listed on the INI Settings page (Distribution Options tab). The default selections (*Create always*, *Create if exists*, *Create or add to existing section*) ensure that a setting is removed only if it was created when the application was installed and not at an earlier time by another application that also needs the setting.

Registry

Click *Registry*, then select the attributes you want to use to determine whether or not a registry setting is removed. The attributes correspond to the attributes listed on the Registry Settings page (Distribution Options tab). The default selections (*Create always*, *Create if exists*) ensure that a setting is removed only if it was created when the application was installed and not at an earlier time by another application that still needs the setting.

Uninstall Keys/Values From These Registry Hives

This option allows you to select two specific registry hives, `HKEY_LOCAL_MACHINE` and `HKEY_CURRENT_USER`, to include or exclude when removing the registry settings that were added by Application Launcher during distribution of the application to the workstation. The default settings cause Application Launcher to remove settings from the `HKEY_CURRENT_USER` hive but not from the `HKEY_LOCAL_MACHINE` hive.

The purpose of this option is to help you ensure that no Windows system settings are removed during the uninstall. This is particularly important if you used snAppShot to capture the

application's installation. When snAppShot captures an application's installation, it includes all settings that are changed during the installation. If the application's installation program causes the workstation to reboot during the installation process, these changes can include not only settings that apply to the application but also settings that apply to the Windows system. Removing Windows system settings might cause the workstation to encounter problems when starting up. We strongly recommended that you remove these settings only if you are positive that they do not affect the workstation's ability to function properly.

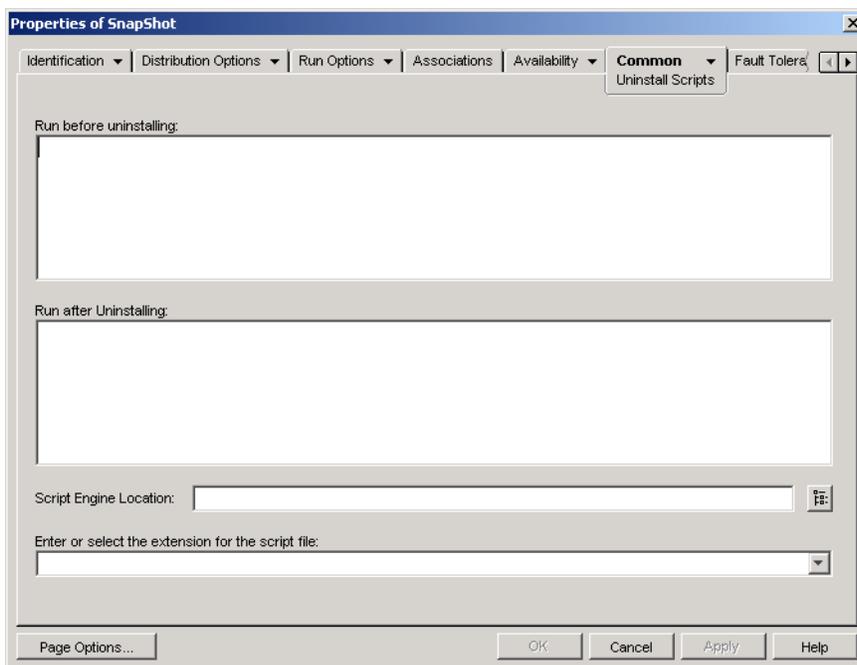
If you want to use this option, you should examine the registry settings that will be removed from the HKEY_LOCAL_MACHINE and HKEY_CURRENT_USER hives and delete any settings that apply to anything other than the application. The registry settings are displayed on the Registry Settings page (Distribution Options tab).

48.6.8 Uninstall Scripts Page

The Uninstall Scripts property page is available only on Application objects created for simple applications, AOT/AXT applications, and MSI applications. It is not available on Application objects created for Web applications and terminal server applications.

As part of the process of uninstalling the application, Application Launcher can launch a script engine to execute a "before uninstall" script and an "after uninstall" script. The Uninstall Scripts page, shown below, defines the script engine that you want Application Launcher to use and the scripts you want executed.

Figure 48-50 Application Object > Common Tab > Uninstall Scripts



On Windows 2000/XP, uninstall scripts always run in the user space for both user-associated applications and workstation-associated applications. This is the same behavior as launch scripts (*Run Options* tab > *Launch Scripts*) but different than the behavior for distribution scripts (*Distribution Options* tab > *Distribution Scripts*), which run in the secure system space.

Run Before Uninstalling

Use this text box to specify any script commands you want executed before the application is uninstalled. Do not use extended characters in the script; extended characters are not supported. For script example, see [“Script Example” on page 577](#).

Run After Uninstalling

Use this text box to specify any script commands you want executed after the application is uninstalled. Do not use extended characters in the script; extended characters are not supported. For script example, see [“Script Example” on page 577](#).

Script Engine Location

The script engine determines the script commands and scripting language you need to use. If you do not define a script engine in the *Script engine location* field, Application Launcher uses the Novell Client as the script engine (if the workstation has the Novell Client installed), which means that you can use most Novell Client login script commands (see [“Supported Novell Client Login Script Commands” on page 577](#)).

If you want to use a script engine other than the Novell Client, specify the alternate script engine. The script engine must reside in a location that is always available to users, such as their local drives. The script engine can reside on a network server only if users can map a drive to the server (for example, through the Novell Client or the Client for Microsoft Networks). If Application Launcher cannot find the script engine, it displays an error to the user and fails to uninstall the application.

If you use the Windows command interpreter as the script engine, you must include the `/c` switch, as shown in the following examples:

- ◆ Windows 2000/XP: `%*winsysdir%\cmd.exe /c`
- ◆ Windows 98: `%*windir%\command.com /c`

The `%*winsysdir%` and `%*windir%` variables specify the Windows system directory (for example, `c:\winnt\system32`), and the `/c` switch instructs the command interpreter to execute the script and then stop. If the `/c` switch is not used, the script does not complete.

For a script example, see [“Script Example” on page 577](#).

Script File Extension

This applies only if you specified a script engine in the *Script engine location* field.

When the application is uninstalled, Application Launcher creates temporary script files for the *Run before uninstall* scripts and *Run after uninstall* scripts. These files are passed to the script engine, which then executes the script. You need to specify the file extension that the script engine requires for its script files.

For a script example, see [“Script Example” on page 577](#).

Script Example

The following script uses the Windows 2000/XP command interpreter as the script engine. Before the distribution occurs, a listing of the `c:\` directory is saved to a text file and the `autoexec.bat` file is backed up.

Run Before Uninstalling Field

```
dir c:\ >c:\1.txt  
copy autoexec.bat autoexec.bak /y
```

Script Engine Location Field

```
cmd.exe /c
```

Enter or Select the Extension of the Script File Field

```
.bat
```

Supported Novell Client Login Script Commands

When using the Novell Client as the script engine, you can use all but the following script commands:

Table 48-5 Supported Novell Client Login Script Commands

CLS	INCLUDE	PCOMPATIBLE
DISPLAY	LASTLOGINTIME	SCRIPT_SERVER
EXIT	NO_DEFAULT	SET_TIME
FDISPLAY	NOSWAP	SWAP
IF MEMBER OF	PAUSE	WRITE

Application Launcher does not output anything to the screen or display script errors.

For script commands, syntax, and examples, see the [Novell Client documentation \(http://www.novell.com/documentation/lg/noclienu/docui/index.html\)](http://www.novell.com/documentation/lg/noclienu/docui/index.html) on the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

48.7 MSI Tab

The MSI tab is available only on Application objects created for MSI applications.

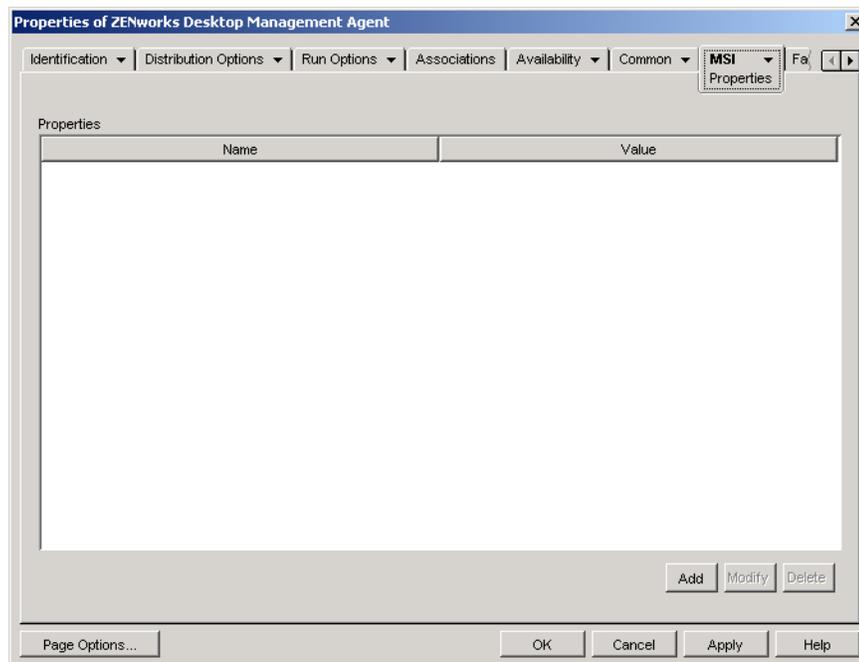
The MSI tab includes the following pages to help you configure and manage the distribution of MSI applications installed through the Microsoft Windows Installer:

- ◆ [Section 48.7.1, “Properties Page,” on page 578](#)
- ◆ [Section 48.7.2, “Transforms Page,” on page 579](#)
- ◆ [Section 48.7.3, “Verify Page,” on page 580](#)
- ◆ [Section 48.7.4, “Patches Page,” on page 582](#)

48.7.1 Properties Page

The Properties property page, shown below, lets you override the public property values contained in the MSI package and add new public properties.

Figure 48-51 Application Object > MSI Tab > Properties Page



The MSI package contains the property values that were defined during the administrative installation of the application. These property values determine the way the Microsoft Windows Installer installs the application to the user's workstation. In some cases, you might want to change one or more of the property values. For example, a property value might define the default location for a user's work files. By adding the property to the Properties list and changing the property's value, you can override the default location defined in the MSI package.

If necessary, you can add public properties that were not included in the MSI package. When doing so, you should be careful to add only those properties that are valid for the package.

Add

To override a property value, you change the property value and add the property to the *Properties* list so that Application Launcher knows to use that property value rather than the one defined in the MSI package. To do so, click *Add* to display the Select Property to Add dialog box. In the *Value name* field, select the property whose value you want to override, specify the new value in the *Value data* field, then click *OK* to add the property to the *Properties* list.

To define a new property that is not already included in the MSI package, click *Add* to display the Select Property to Add dialog box. In the *Value name* field, specify the new property's name, enter the property's value in the *Value data* field, then click *OK* to add the property to the *Properties* list.

For a list of properties and descriptions, see the [Microsoft Developer Network \(MSDN\) \(http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msi/setup/property_reference.asp\)](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msi/setup/property_reference.asp).

Although you can add and define new properties, the *SOURCELIST* property is reserved for Application Launcher use.

NOTE: When Application Management distributes and installs an MSI application, it uses the AlwaysInstallElevated policy, which installs the MSI application using elevated (system) privileges. Because of this, in order to set the *ALLUSERS* property value to null so that the application is installed on a per-user basis, you need to leave the *Value data* field empty. Using double quotation marks (“”) causes the distribution to fail.

Modify

To modify a property that is in the *Properties* list, select the property, click *Modify* to display the Edit String Value dialog box, modify the value data, then click *OK*.

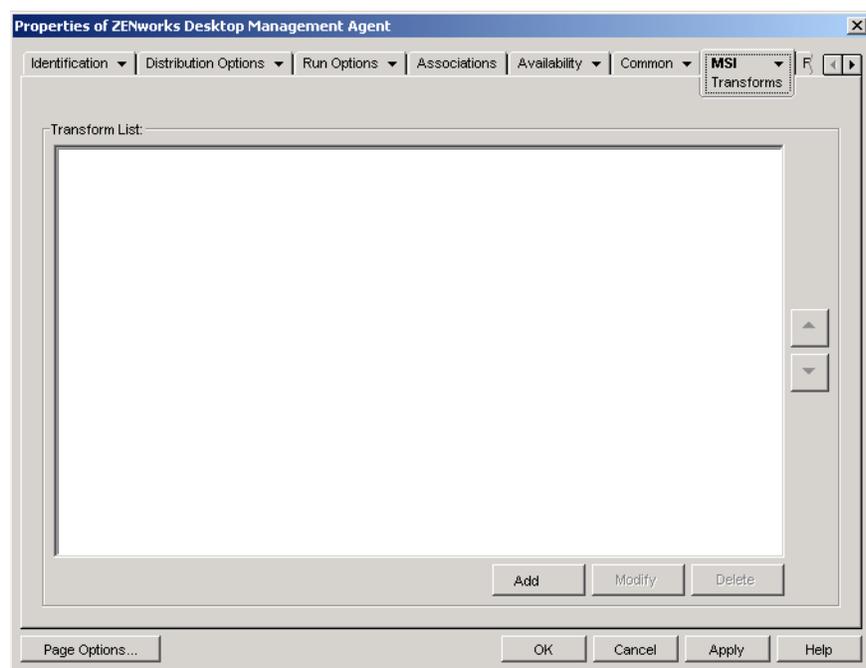
Delete

To delete a property from the Properties list, select the property, then click *Delete*. Deleting the property causes future installations of the application to use the property value defined in the MSI package.

48.7.2 Transforms Page

The Transforms property page, shown below, lists the transforms that Microsoft Windows Installer applies to the MSI package during distribution. A transform adds, deletes, or changes properties in the MSI package to enable customizing of the installation for different groups of users.

Figure 48-52 *Application Object > MSI Tab > Transforms Page*



Transforms are applied in the order they appear in the *Transform* List. If two transforms modify the same package property, the property retains the value applied by the last transform.

For information about creating transforms for your MSI package, see the documentation you received with the software application.

Add

Click *Add*, browse to and select the transform file (.mst file) you want added to the *Transform* list, then click *OK* to add it to the list. If the transform file is in the same directory as the .msi file, you can simply enter the filename in the *Transform path* field rather than browsing to and selecting it. If you select the file, make sure that the resulting path is either a UNC path or mapped path common to all users.

Delete

Select a transform from the *Transforms list*, then click *Delete* to remove it from the list.

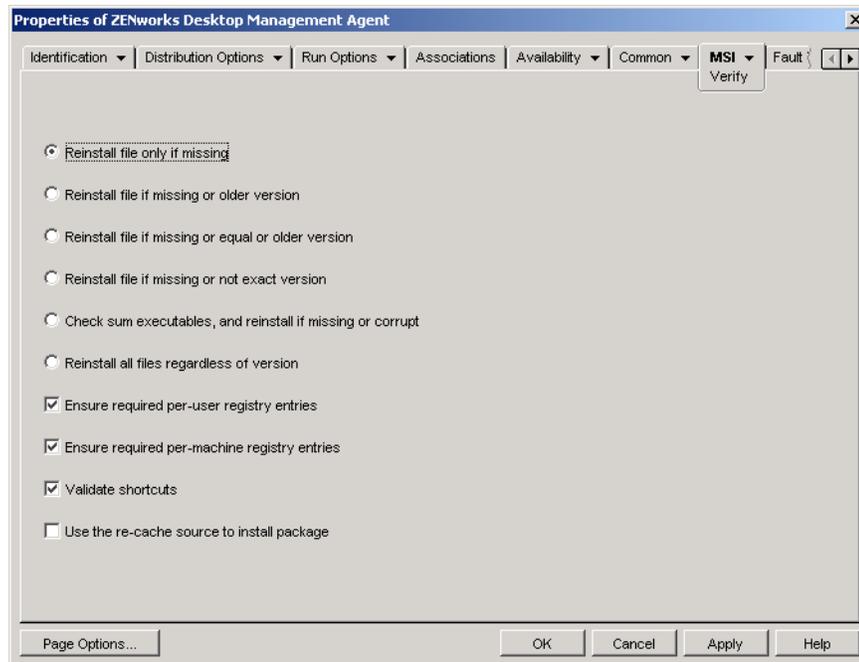
Up-Arrow and Down-Arrow

Select a transform from the *Transform list*, then click the up-arrow to move the transform up in the list or click the down-arrow to move it down in the list. Microsoft Windows Installer applies the transforms in the order they are listed, from top to bottom.

48.7.3 Verify Page

The Verify property page, shown below, determines the actions that Microsoft Windows Installer takes when instructed to verify the application.

Figure 48-53 Application Object > MSI Tab > Verify Page



When a user verifies the MSI application, Application Launcher launches Windows Installer. If no options are selected, Windows Installer performs a full verification, which results in all files,

settings, and so forth being reinstalled. If one or more options are selected, Windows Installer only performs the actions specified by the selected options.

The reinstall options are mutually exclusive, meaning that you can only select one of the options. The remaining four options are independent of one another: some, all, or none can be selected.

Reinstall File Only if Missing

Instructs Windows Installer to reinstall a file only if it is missing.

Reinstall File if Missing or Older Version

Instructs Windows Installer to reinstall a file if it is missing or if the installed file's version is older than the file in the MSI package.

Reinstall File if Missing or Equal or Older Version

Instructs Windows Installer to reinstall a file if it is missing or if the installed file's version is the same as or older than the file in the MSI package.

Reinstall File if Missing or Not Exact Version

Instructs Windows Installer to reinstall a file if it is missing or if the installed file's version is not exactly the same as the file in the MSI package.

Check Sum Executables, and Reinstall if Missing or Corrupt

Instructs Windows Installer to perform a checksum on all executable files and to reinstall a file if it is missing or if the checksum verifies that the file is corrupt. Only files that have `msidbFileAttributesChecksum` in the *Attributes* column of the MSI package's File Table are repaired.

Reinstall All Files Regardless of Version

Instructs Windows Installer to reinstall all files.

Ensure Required Per-User Registry Entries

Instructs Windows Installer to rewrite all per-user entries from the MSI package to the Windows system registry. Per-user entries are those entries contained in the `HKEY_CURRENT_USER` and `HKEY_USERS` registry hives.

Ensure Required Per-Machine Registry Entries

Instructs Windows Installer to rewrite all per-machine entries from the MSI package to the Windows system registry. Per-machine entries are those entries contained in the `HKEY_LOCAL_MACHINE` and `HKEY_CLASSES_ROOT` registry hives.

Validate Shortcuts

Instructs Windows Installer to reinstall the MSI application's shortcuts, overwriting any existing shortcuts and icons.

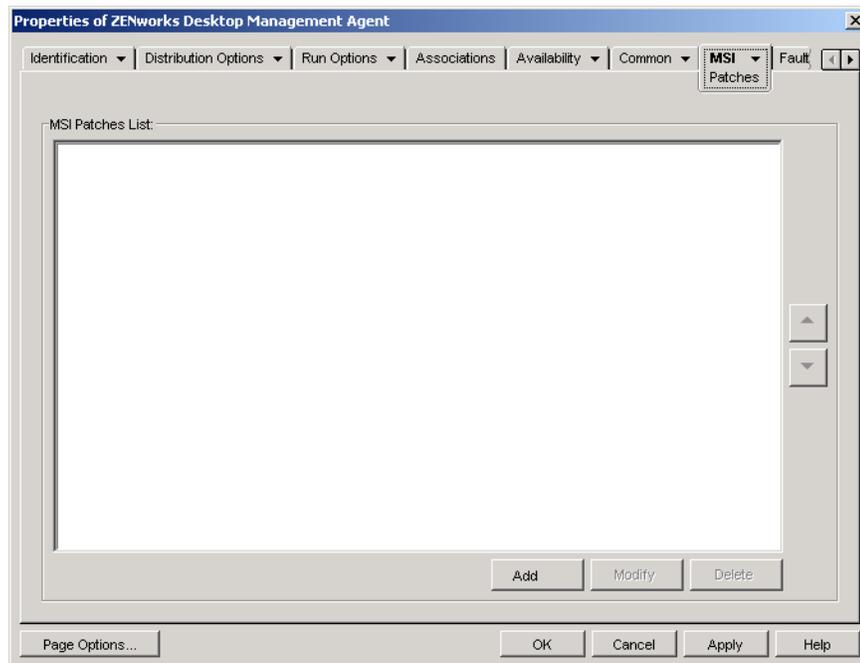
Use the Re-Cache Source to Install Package

Instructs Windows Installer to install files from the re-cache (local) source rather than the source package.

48.7.4 Patches Page

The Patches property page, shown below, lists the patch files that Microsoft Windows Installer applies to the MSI package during distribution. Each patch is applied in the order listed in the MSI Patches List.

Figure 48-54 *Application Object > MSI Tab > Patches Page*



Add

Click *Add*, browse to and select the patch file (.msp file or other MSI-based patch file) you want added to the MSI Patches List, then click *OK* to add it to the list. If the patch file is in the same directory as the .msi file, you can simply enter the filename in the MSI Patch File field rather than browsing to and selecting it. If you select the file, make sure that the resulting path is either a UNC path or mapped path that is common to all users.

Modify

Select a patch from the *MSI patches* list, then click *Modify* to edit the location of the patch.

Delete

Select a patch from the *MSI patches* list, then click *Delete* to remove it from the list.

Up-Arrow and Down-Arrow

Select a patch from the MSI Patches List, then click the up-arrow to move the patch up in the list or click the down-arrow to move it down in the list. Microsoft Windows Installer applies the patches in the order they are listed, from top to bottom.

48.8 Terminal Server Client Tab

The Terminal Server Client tab is available only on Application objects created for terminal server applications.

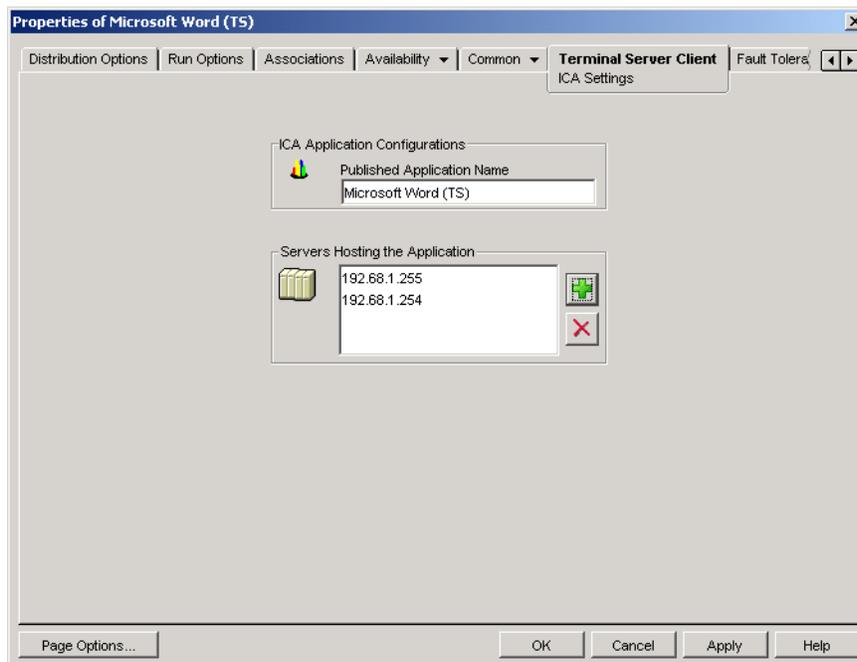
The Terminal Server Client tab includes one of the following pages, depending on whether the Application object's defined client protocol is ICA or RDP:

- ♦ “ICA Settings Page” on page 583
- ♦ “RDP Settings Page” on page 584

48.8.1 ICA Settings Page

The ICA Settings property page, shown below, lets configure the settings used by the Citrix ICA client when launching the application from a Citrix MetaFrame Server.

Figure 48-55 Application Object > Terminal Server Client Tab > ICA Settings Page



Published Application Name

Specify the application name exactly as it is defined in Citrix.

Servers Hosting This Application

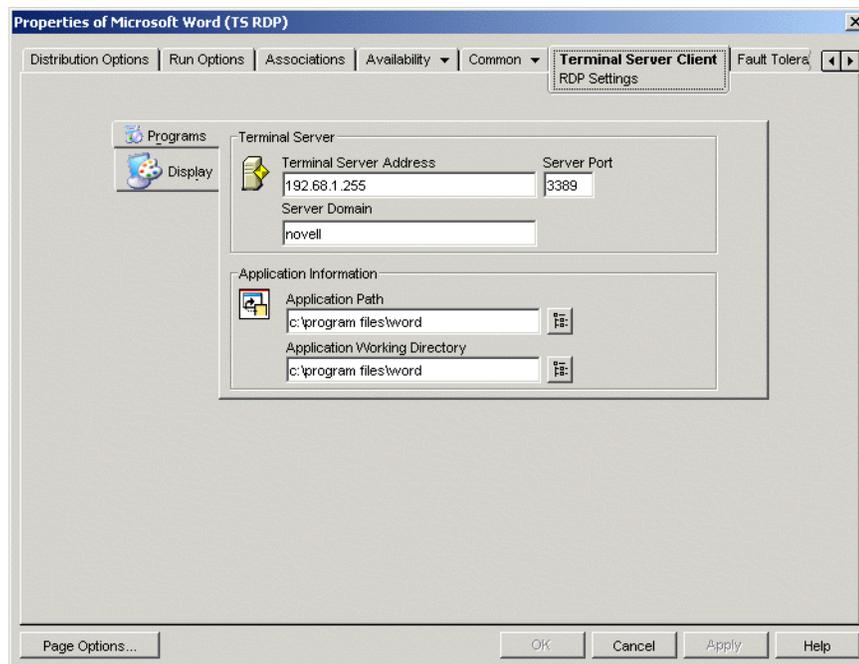
Add the Citrix servers that host the application. To add a server, click the *Add* button, specify the server's IP address or hostname, then click *OK*.

The server's you define here are used only when the application is launched from Application Launcher or the NAL plug-in. The ZENworks Launch gadget uses the servers defined in its configuration settings. For information about the ZENworks Launch gadget's configuration settings, see [Chapter 26, “ZENworks Launch Gadget: Configuring Settings,”](#) on page 319.

48.8.2 RDP Settings Page

The RDP Settings property page lets you configure the settings used by the RDP client when launching the application from a Windows Terminal Server.

Figure 48-56 *Application Object > Terminal Server Client Tab > Window Page*



Programs

The Program settings define the location from which the application runs. Click the *Programs* tab to show the following settings.

- ◆ **Terminal server address and server port:** Specify the terminal server's IP address or hostname. If the terminal server is not using default port 3389, edit the Server Port field to specify the correct port number.
- ◆ **Server domain:** If the terminal server is part of a Windows NT domain or an Active Directory domain, specify the domain name. If a user's name and password in the domain matches the user name and password in eDirectory, the user is not prompted to log in to the terminal server when launching the application.

- ◆ **Application path:** Specify the path to the application's executable file from the perspective of the terminal server. If you are running ConsoleOne on the terminal server, you can browse for and select the file.
- ◆ **Working directory:** Specify the path to the directory you want the application to use for its working files.

Display

The Display settings determine the window size and number of colors used for the RDP client session. Click the Display tab to show the following settings.

- ◆ **Remote desktop size:** Select the window size for the RDP client session. The window size determines the resolution at which information is displayed and the amount of screen area used for the window. The higher the resolution, the smaller the window size and information display.

By default, the user's RDP client setting is used. You can change the size to full screen or another standard screen resolution.
- ◆ **Colors:** Select the number of colors for the RDP client session. You can select 256 colors, high color (15 bits), high color (16 bits), or true color (32 bits). The default is 256 colors.

48.9 Fault Tolerance Tab

The Fault Tolerance tab includes the following pages to help you balance the workload required to host an application between multiple servers and provide access to backup applications in case of server failure:

- ◆ [Section 48.9.1, “Fault Tolerance Page,” on page 585](#)
- ◆ [Section 48.9.2, “Load Balancing Page,” on page 587](#)
- ◆ [Section 48.9.3, “Site List Page,” on page 589](#)
- ◆ [Section 48.9.4, “Remote Alternate App Page,” on page 591](#)

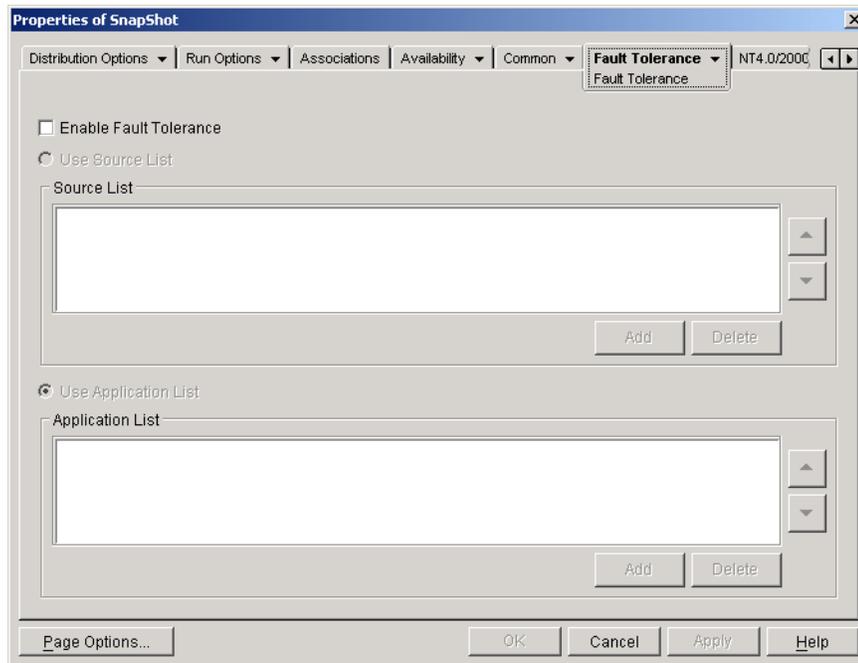
48.9.1 Fault Tolerance Page

The Fault Tolerance property page is available only on Application objects created for simple applications and AOT/AXT applications. It is not available on Application objects created for MSI applications, terminal server applications, and Web applications.

This section provides information about the fields located on the Fault Tolerance property page. For complete instructions on setting up fault tolerance, see [Section 36.1, “Setting Up Fault Tolerance,” on page 387](#).

The Fault Tolerance property page, shown below, specifies the installation package sources or Application objects that Application Launcher uses as backups if this Application object becomes unavailable. Application Launcher tries the package sources or Application objects in the order shown in the *Source List* or *Application List*, from top to bottom.

Figure 48-57 Application Object > Fault Tolerance Tab > Fault Tolerance Page



The Fault Tolerance page varies according to the type of application. For simple/AOT/AXT applications, the page looks like the one shown above. For Web applications, the page includes only the Application List because the Source List functionality is not supported.

Enable Fault Tolerance

Select this option to enable fault tolerance. The other fault tolerance options are available only if this option is enabled.

Use Source List

Select this option to have Application Launcher use a list of installation package sources as the backup. You must have already created at least one package source (*Common > Sources* page).

Add

Click *Add*, browse to and select a package source, then click *OK* to add it to the Source List.

Delete

Select a package source from the Source List, then click *Delete* to remove it from the Source List.

Up-Arrow and Down-Arrow

Select a package source from the *Source* list, then click the up-arrow to move the source up in the list or click the down-arrow to move it down in the list. Application Launcher tries the sources in the order they are listed, from top to bottom.

Use Application List

Select this option to have Application Launcher use a list of Application objects as the backup. You must have already created additional Application objects for the application, with the application files being stored on a different server or volume than this application. If the application is not available, then Application Launcher tries each Application object in the specified order.

Only one-level-deep fault tolerance is supported. If Application Launcher fails over to one of the backup Application objects in the list and the backup Application object fails, the distribution fails even if fault tolerance is enabled for the backup Application object.

NOTE: If Application Launcher has detected a remote connection and is running in remote mode, it does not use the Application List for fault tolerance. Application Launcher uses only the Source List.

Add

Click *Add*, browse to and select an Application object, then click *OK* to add it to the Application List.

Delete

Select an Application object from the *Application* list, then click *Delete* to remove it from the Application List.

Up-Arrow and Down-Arrow

Select an Application object from the *Source* list, then click the up-arrow to move the Application object up in the list or click the down-arrow to move it down in the list. Application Launcher tries the Application objects in the order they are listed, from top to bottom.

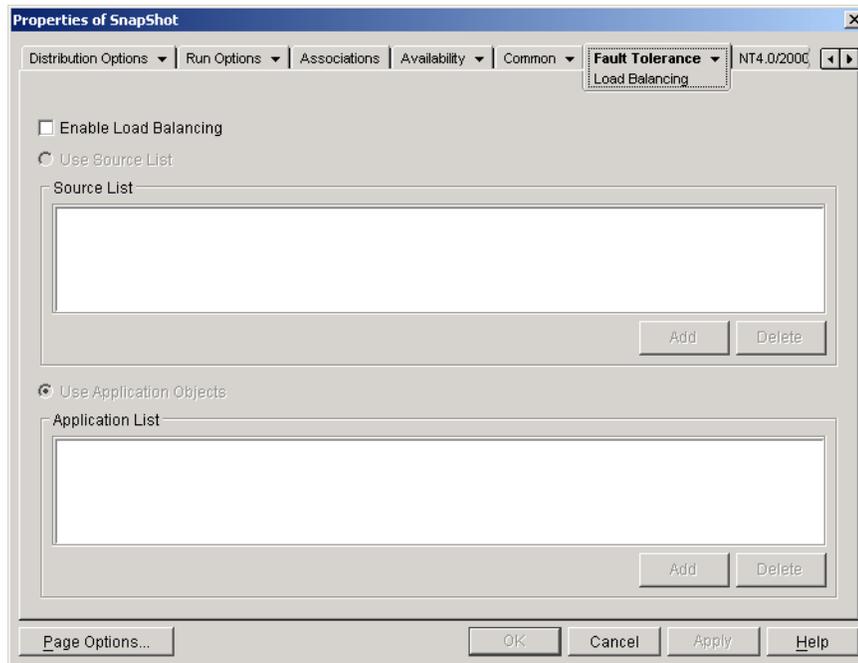
48.9.2 Load Balancing Page

The Load Balancing property page is available only on Application objects created for simple applications, AOT/AXT applications, and MSI applications. It is not available on Application objects created for terminal server applications and Web applications.

This section provides information about the fields located on the Load Balancing property page. For complete instructions on setting up load balancing, see [Section 36.2, “Setting Up Load Balancing,” on page 391](#).

The Load Balancing property page, shown below, lets you balance the work of distributing the application. With load balancing enabled, Application Launcher distributes the application from all servers offering the application, ensuring a more balanced workload across servers.

Figure 48-58 Application Object > Fault Tolerance Tab > Load Balancing Page



The Load Balancing page varies according to the type of application. For simple/AOT/AXT applications, the page looks like the one shown above. For MSI applications, it looks like the one above, but Enable Load Balancing is the only available option. For Web applications, the page includes only the Application List because the Source List functionality is not supported.

To use load balancing, you must first have multiple installation package sources or Application objects defined for the application, with the application source files located on different servers. When a user launches the application, Application Launcher selects one of the package sources or Application objects to use. If the package source or Application object is unavailable, Application Launcher selects another one.

Enable Load Balancing

Select this option to enable load balancing. The other load balancing options are available only if this option is enabled.

For Application objects that use Microsoft Windows Installer (MSI) packages, the *Enable load balancing* option is the only option that is enabled on this page; the *Use source list* and *Use application list* options are disabled. To enable load balancing for an MSI Application object, select this option and then use the Application object's Sources page (Common tab) to define the list of source locations where you've created additional installation packages for the application. Application Launcher randomly selects source locations from the list to balance the workload.

Use Source List

Select this option to have Application Launcher use a list of package sources for load balancing. You must have already created at least one package source (*Common > Sources* page).

Add

Click *Add*, browse to and select a package source, then click *OK* to add it to the Source List.

Delete

Select a package source from the Source List, then click *Delete* to remove it from the Source List.

Use Application List

Select this option to have Application Launcher use a list of Application objects for load balancing. You must have already created additional Application objects for the application, with the application files being stored on a different server or volume than this application.

Add

Click *Add*, browse to and select an Application object, then click *OK* to add it to the Application List.

Delete

Select an Application object from the *Application* list, then click *Delete* to remove it from the Application List.

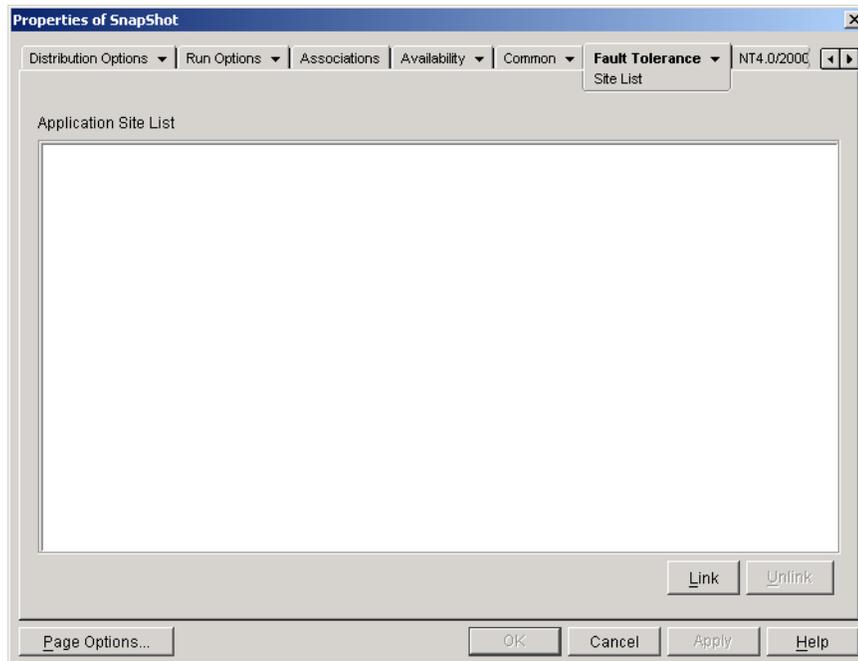
48.9.3 Site List Page

The Site List property page is available only on Application objects created for simple applications, AOT/AXT applications, MSI applications, and terminal server applications. It is not available on Application objects created for Web applications.

This section provides information about the fields located on the Site List property page. For complete instructions on setting up site lists, see [Section 36.3, “Setting Up Site Lists,” on page 395](#).

The Site List property page, shown below, lets you link an application at one site (Site1) to an application at another site (Site2). By linking applications whose source installation packages are located on servers at different sites, you can distribute the application from the server that is closest to the user.

Figure 48-59 Application Object > Fault Tolerance Tab > Site List Page



For example, you have two different sites, Site1 and Site2. Employees at both sites use the same spreadsheet program. Site1 has an Application object (App1) that distributes the spreadsheet program from a Site1 server, and Site2 has its own Application object (App2) to distribute the program from a Site2 server. If you link App1 to App2, any Site1 employee who travels to Site2 and launches App1 has App2 installed from the Site2 server. Likewise, any Site2 employee who travels to Site1 and launches App2 has App1 installed from the Site1 server.

The site list is a distribution mechanism that applies only to undistributed, non-cached applications. If an application is already distributed or cached on a user's workstation, that application is used regardless of the site list.

Novell Client vs. ZENworks Middle Tier Server

When a user logs in to Novell eDirectory through the ZENworks Middle Tier Server rather than the Novell Client, site lists work slightly differently. Rather than the Novell Client being used to determine the user's location, the Middle Tier Server determines the location. This means that the application closest to the Middle Tier Server is used, which is not necessarily the application closest to the user. In the above example, the Site1 user would continue to have applications distributed from the Site1 server rather than from a Site2 server.

How to Build a Site List

You can link to only one other Application object. However, when you do so, the first Application object is automatically linked to any other Application objects that the second Application object is linked to. For example, you have three identical Application objects (App1, App2, and App3) at three different sites. You link App1 to App2, which creates the following site lists for each of the Application objects:

App1 List	App2 List	App3 List
App2	App1	(none)

Because you can link an Application object to only one other Application object, you now need to link App3 to either App1 or App2. Doing so creates the following site list for each of the Application objects:

App1 List	App2 List	App3 List
App2	App1	App1
App3	App3	App2

Link

Click *Link*, browse for and select the Application object you want to link to, then click *OK* to add it to the *Application Site* list. Any other Application objects that the selected Application object is linked to are also added to the list.

Unlink

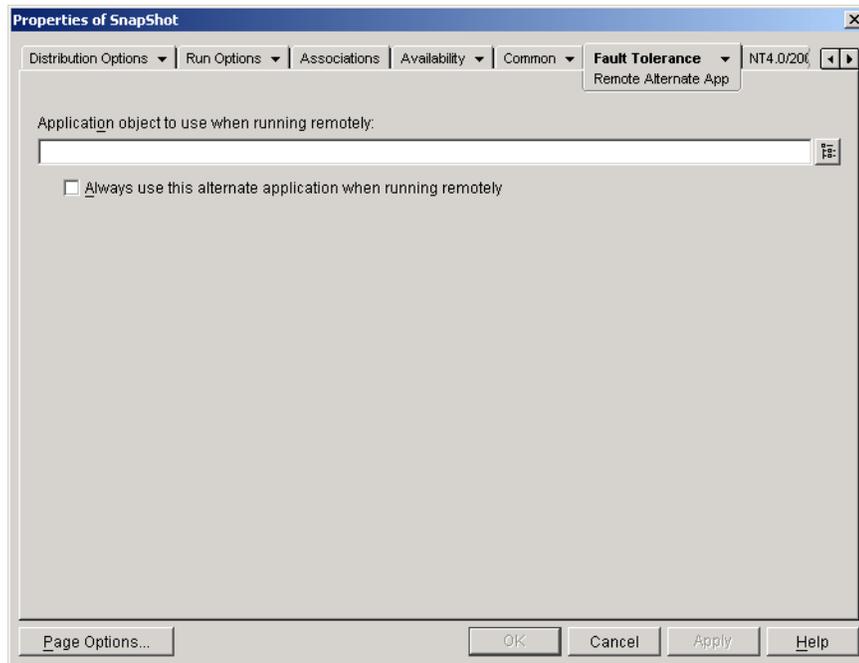
Click *Unlink* to remove the Application object's link to the Application objects displayed in the *Application Site* list.

48.9.4 Remote Alternate App Page

The Remote Alternate App property page is available on Application objects created for all application types (simple, AOT/AXT, MSI, Web, and terminal server).

The Remote Alternate App property page, shown below, lets you specify an alternate application that Application Launcher launches if the user's workstation is running in remote mode rather than local mode. Generally, this alternate application should be a terminal server application or Web application that is designed for use over slower remote connections.

Figure 48-60 Application Object > Fault Tolerance Tab > Remote Alternate App Page



Application Object to Use When Running Remotely

Select the Application object for the alternate application.

Always Use this Alternate Application When Running Remotely

By default, Application Launcher does not use the alternate application if the original application is installed on the user's workstation; instead, it uses the locally installed application. However, if the application requires access to a database or some other network resource that is only available if the alternate application is used, select this option to force Application Launcher to use the alternate application rather than the locally installed application.

Novell® Application Launcher™ supports the use of macros, or variables, with many of the properties available on the Application object pages in ConsoleOne®. You can use any of the six types of macros listed below.

- ◆ Application object macros
- ◆ Special Windows macros
- ◆ Login script macros
- ◆ Novell® eDirectory™ attribute macros
- ◆ Environment variable macros
- ◆ Language variable macros

Each macro type gives you various abilities. For example, Application object macros are basically variables to which you can assign values. By using an Application object macro, you have the flexibility to change the macro's value in one location and have it automatically changed anywhere you have used the macro. Special Windows macros, on the other hand, let you specify Windows locations, such as the Windows temporary directory, by specifying a variable name (for example, %*WINTEMP%) that exists on the workstation and defines the location on that workstation. This enables you to specify locations that might vary from workstation to workstation.

NOTE: Be aware that using macros causes increased network traffic because the ZENworks® macro lookup checks with eDirectory on the network before it checks the local environment.

The following sections provide information about the five types of macros supported by Application objects, the precedence order, and the Application object properties where macros can be used.

- ◆ [Section 49.1, “Application Object Macros,” on page 593](#)
- ◆ [Section 49.2, “Special Windows Macros,” on page 595](#)
- ◆ [Section 49.3, “Login Script Macros,” on page 598](#)
- ◆ [Section 49.4, “eDirectory Attribute Macros,” on page 600](#)
- ◆ [Section 49.5, “Environment Variable Macros,” on page 602](#)
- ◆ [Section 49.6, “Language Variable Macros,” on page 602](#)
- ◆ [Section 49.7, “Macro Precedence,” on page 605](#)
- ◆ [Section 49.8, “Application Object Properties Where Macros Can Be Used,” on page 605](#)
- ◆ [Section 49.9, “Refreshing Macro Information,” on page 607](#)

49.1 Application Object Macros

An Application object macro is defined on the Application object's Macros page and can only be used with that Application object. The macro is simply a variable that has a value associated with it, as shown in the following two examples:

Variable	Value
SOURCE_PATH	\\server1\sys\public\zen\word
TARGET_PATH	c:\program files\word

In the examples, SOURCE_PATH defines the network location from which Novell Application Launcher™ distributes the application to the user's workstation, and TARGET_PATH defines the workstation location to which Application Launcher copies the files.

You can also create a macro that prompts the user to accept a default value you've assigned or to enter a different value. For example, you could use a macro that prompts the user with a default installation drive (such as c:) but lets the user change it to another drive (such as d:).

The following sections provide information about nesting other macros in Application object macros and instructions to help you define Application object macros:

- ◆ [Section 49.1.1, “Nesting Macros,” on page 594](#)
- ◆ [Section 49.1.2, “Defining an Application Object Macro,” on page 594](#)

49.1.1 Nesting Macros

When defining an Application object macro, you can embed other macros in the definition. In the following example, a special Windows macro, *WINDISK, is used when defining the TARGET_PATH macro:

```
%*WINDISK%\Program Files
```

Any time you use the TARGET_PATH macro in an Application object property field, Application Launcher substitutes the current workstation's Windows drive letter for %*WINDISK% (for example, c:\temp).

When embedding another macro in the string, you must enclose the macro in % characters, as shown in the above example. You can embed special Windows macros, eDirectory attribute macros, environment variable macros, login script macros, and other Application object macros.

49.1.2 Defining an Application Object Macro

- 1 In ConsoleOne, right-click the Application object for which you want to define the macro, then click *Properties*.
- 2 Click the *Common* tab > *Macros* to display the Macros page.
- 3 To define a macro consisting of a variable name and a value (a String Value macro), continue with [“Defining a String Value \(Non-Prompted\) Macro” on page 594](#).
or
To define a macro that prompts the user to accept a default value or enter a different value, skip to [“Defining a Prompted Macro” on page 595](#).

Defining a String Value (Non-Prompted) Macro

- 1 On the Macros page, click *Add > String value* to display the Edit String Value dialog box.

- 2 In the Value Name box, type a name for the macro variable (for example, TARGET_PATH). Do not use spaces in the name.
- 3 In the Value Data box, type the value for the macro variable (for example, c:\program files\word).
- 4 Click *OK* to add the macro to the Macros list.

For information about using the macro in a valid Application object property, see [Section 49.8, “Application Object Properties Where Macros Can Be Used,”](#) on page 605.

Defining a Prompted Macro

Prompted macros do not apply to MSI applications. With all applications other than MSI applications, Application Launcher controls the installation process and can therefore supply the prompt. With MSI applications, Windows Installer, not Application Launcher, installs the applications; Application Launcher has no opportunity to supply the prompt.

- 1 On the Macros page, click *Add > Prompted > Drive* to create a macro that prompts the user for a drive or click *Add > Prompted > String* to create a macro that prompts the user for text input.
- 2 Fill in the following fields:

Macro name Specify any name that describes the macro. Do not use spaces in the name.

Prompt text: Specify the text you want presented to the user. For example, if you are prompting for an installation drive you could enter the following: “This application will be installed to your C: drive. Enter a different drive letter if desired.”

Default value: For a prompted drive macro, select the drive to display as the default drive. For a prompted string macro, specify the text to display as the default text.

Minimum disk space in MB: Specify the minimum amount of free disk space required. The user cannot install to any drive that does not meet the minimum disk space requirement.

Maximum string length in chars (prompted string macro only): Specify the maximum number of characters the user can enter as a response to a prompted string macro.

- 3 Click *OK* to add the macro to the Macros list.

For information about using the macro in a valid Application object property, see [Section 49.8, “Application Object Properties Where Macros Can Be Used,”](#) on page 605.

49.2 Special Windows Macros

A special Windows macro is one that defines Windows 98 and Windows 2000/XP directories. The typical paths listed below are based on default installations and might or might not match your specific setup. On Windows 98 workstations, macros behave differently if User Profiles are enabled.

Suppose that you have installed Windows to drive d: (for example, d:\winnt). However, an application installation expects Windows to be on drive c: (for example, c:\winnt). Using the *WinDisk macro, you can substitute drive d: with the macro for the files that require it.

The following two sections describe the special Windows macros you can use.

- ◆ [Section 49.2.1, “Windows 98 and Windows 2000/XP Macros,”](#) on page 596
- ◆ [Section 49.2.2, “Windows 2000/XP Macros,”](#) on page 597

For information about using the macros in a valid Application object property, see [Section 49.8, “Application Object Properties Where Macros Can Be Used,”](#) on page 605.

49.2.1 Windows 98 and Windows 2000/XP Macros

The following macros apply to Windows 98 and Windows 2000/XP unless otherwise noted.

IMPORTANT: The asterisk character (*) is a required syntax for these macros. Don’t confuse these asterisk characters with the Novell trademark asterisk. In addition, when using the macro in one of the Application object’s property fields, you must enclose it in % characters (for example, %*WinDisk%).

Table 49-1 Windows 98 and Windows 2000/XP Macros

Macro	Description
%*AppData%	File system directory that serves as a common repository for application specific data. Typically <code>c:\winnt\profiles\user name\application data</code> . Used only by Windows NT version 4.
%*CommonDesktop%	File system directory that contains files and folders that appear on the desktop for all users. Typically <code>c:\winnt\profiles\all users\desktop</code> . If not available, the *DESKTOP value is used.
%*CommonPrograms%	File system directory that contains the directories for the common program groups that appear on the Start menu for all users. Typically <code>c:\winnt\profiles\all users\start menu\programs</code> . If not available, the *PROGRAMS value is used.
%*CommonStartMenu%	File system directory that contains the programs and folders that appear on the Start menu for all users. Typically <code>c:\winnt\profiles\all users\start menu</code> . If not available, the *STARTMENU value is used.
%*CommonStartup%	File system directory that contains the programs that appear in the Startup folder for all users. The system starts these programs whenever any user logs on to Windows NT 4 or starts Windows 98. Typically this directory is <code>c:\winnt\profiles\all users\start menu\programs\startup</code> . If not available, the *STARTUP value is used.
%*CommonWinDesktop%	Windows NT 4 common desktop directory (<code>c:\winnt\profiles\all users\desktop</code>). This is an Application Launcher 2.0 macro that is allowed for backward compatibility.
%*Cookies%	Files system directory that contains the user’s cookies. Typically <code>c:\windows\cookies</code> or <code>c:\winnt\profiles\username\cookies</code> .
%*Desktop%	File system directory used to physically store file objects on the desktop (not to be confused with the desktop folder itself). Typically this directory is <code>c:\windows\desktop</code> or <code>c:\winnt\profiles\username\desktop</code> .
%*Favorites%	File system directory that serves as a common repository for the user’s favorite items. Typically this directory is <code>c:\windows\favorites</code> or <code>c:\winnt\profiles\username\favorites</code> .
%*Fonts%	Virtual folder containing fonts. Typically <code>c:\windows\fonts</code> or <code>c:\winnt\fonts</code> .

Macro	Description
%*History%	File system directory that contains the user's history of visited Internet addresses. Typically <code>c:\windows\history</code> or <code>c:\winnt\profiles\username\history</code> .
%*NetHood%	File system directory containing objects that appear in the network neighborhood. Typically <code>c:\windows\nethood</code> or <code>c:\winnt\profiles\username\nethood</code> .
%*Personal%	File system directory that serves as a common repository for documents. Typically <code>c:\myfiles</code> or <code>c:\winnt\profiles\username\personal</code> .
%*PrintHood%	File system directory that serves as a common repository for printer links. Typically <code>c:\winnt\profiles\username\printhood</code> . Only used by Windows NT 4.
%*Programs%	File system directory that contains the user's program groups (which are also file system directories). Typically <code>c:\windows\start menu\programs</code> or <code>c:\winnt\profiles\username\start menu\programs</code> .
%*Recent%	File system directory that contains the user's most recently used documents. Typically <code>c:\windows\recent</code> or <code>c:\winnt\profiles\username\recent</code> .
%*SendTo%	File system directory that contains Send To menu items. Typically <code>c:\windows\sendto</code> or <code>c:\winnt\profiles\username\sendto</code> .
%*StartMenu%	File system directory containing Start menu items. Typically <code>c:\windows\start menu</code> or <code>c:\winnt\profiles\username\start menu</code> .
%*Startup%	File system directory that corresponds to the user's Startup program group. Typically <code>c:\windows\start menu\programs\startup</code> or <code>c:\winnt\profiles\username\start menu\programs\startup</code> .
%*TempDir%	Windows temporary directory (<code>c:\windows\temp</code>).
%*Templates%	File system directory that serves as a common repository for document templates. Typically <code>c:\windows\shellnew</code> or <code>c:\winnt\shellnew</code> .
%*WinDesktop%	Windows desktop directory (<code>c:\windows\desktop</code> or <code>c:\winnt\profiles\username\desktop</code> for Windows NT 4). This is an Application Launcher 2.0 macro that is allowed for backward compatibility.
%*WinDir%	Windows directory. Typically <code>c:\windows</code> or <code>c:\winnt</code> .
%*WinDisk%	Drive letter (plus colon) for the Windows directory. Typically <code>c:.</code>
%*WinSysDir%	Windows system directory. Typically <code>c:\windows\system</code> or <code>c:\winnt\system32</code> .
%*WinSysDisk%	Drive letter (plus colon) for the Windows system directory. Typically <code>c:.</code>

49.2.2 Windows 2000/XP Macros

The following macros apply to Windows 2000/XP only.

IMPORTANT: The asterisk character (*) is a required syntax for these macros. Don't confuse these asterisk characters with the Novell trademark asterisk. In addition, when using the macro in one of

the Application object's property fields, you must enclose it in % characters (for example, %*WinDisk%).

Table 49-2 Windows 2000/XP Macros

Macro	Description
%*AdminTools%	File system directory that contains the administrative tools that appear in the Control Panel when a specific user logs in to Windows 2000/XP. Typically <code>c:\documents and settings\username\start menu\programs\administrative tools</code> .
%*CommonAdminTools%	File system directory that contains the administrative tools that appear in the Control Panel for all users who log in to Windows 2000/XP. Typically <code>c:\documents and settings\all users\start menu\programs\administrative tools</code> .
%*CommonAppData%	File system directory that contains the application-specific data for all users who log in to Windows 2000/XP. Typically <code>c:\documents and settings\all users\application data</code> .
%*CommonDocuments%	File system directory that contains the documents shared by all users who log in to Windows 2000/XP. Typically <code>c:\documents and settings\all users\documents</code> .
%*CommonTemplates%	File system directory that contains the document templates shared by all users who log in to Windows 2000/XP. Typically <code>c:\documents and settings\all users\templates</code> .
%*MyPictures%	File system directory that contains a specific user's graphics files. Typically <code>c:\documents and settings\username\my documents\my pictures</code> .
%*ProgramFiles%	File system directory that contains the program files. Typically <code>c:\program files</code> .
%*ProgramFilesCommon%	File system directory that contains the program files shared by multiple applications. Typically <code>c:\program files\common</code> .
%*UserProfile%	File system directory that contains the logged-in user's profile. Typically <code>c:\documents and settings\username</code> .

49.3 Login Script Macros

Novell Client login script commands allow you to use identifier variables to make login scripts more efficient and flexible. Application Launcher supports many, but not all, login script identifier variables for use in Application object macros.

- ◆ [Section 49.3.1, “Supported Login Script Identifier Variables,” on page 599](#)
- ◆ [Section 49.3.2, “Unsupported Login Script Identifier Variables,” on page 600](#)

For information about using the macros in a valid Application object property, see [Section 49.8, “Application Object Properties Where Macros Can Be Used,” on page 605](#).

49.3.1 Supported Login Script Identifier Variables

The following is a list of supported login script identifier variables and what they mean. Alternate variable names are shown in parentheses.

Table 49-3 *Supported Login Script Identifier Variables*

Macro	Description
%COMPUTER_NAME%	The name of the computer. For example: work_pc.
%DAY%	Numeric day of the month. For example: 01, 10, 15.
%FILESERVER% (%FILE_SERVER%)	Name of the NetWare file server or eDirectory monitored connection. For example: APPS_PROD.
%FULL_NAME%	Full name attribute of the User object. This is supported by virtue of Application Launcher's support for eDirectory attributes from the User object.
%HOUR24% (%24HOUR%)	Time of the day according to a 24-hour clock. For example: 02, 05, 14, 22.
%HOUR% (%HOURS%)	Hour of the day. For example: 0 = 12, 13 = 1.
%LAST_NAME%	Last name of the current user (also known as the user's eDirectory Surname attribute). For example: Jones.
%LOGIN_NAME%	First eight bytes of the user's eDirectory object name. For example: jsmith.
%MINUTE% (%MINUTES%)	Current minute. For example: 02, 59.
%MONTH%	Current month number. For example: 01 for January.
%NDAY_OF_WEEK%	Numeric day of the week. For example: 1 for Sunday, 2 for Monday.
%NETWORK% (%NETWORK_ADDRESS%)	Workstation network address. For example: 01010120.
%OS_VERSION%	Version of the OS. For example: v5.00 (Win3x shows the DOS version, Win 98 and Win 2000/XP shows the Windows version).
%OS	OS type. For example: MSDOS, WIN98, WINNT, WIN2000, WINXP. (Win3 shows MSDOS.)
%PLATFORM%	Platform running. For example: WIN, W98, WNT, W2000, WXP.
%PHYSICAL_STATION% (%P_STATION%)	MAC address. For example: 0000C04FD92ECA.
%REQUESTER_CONTEXT%	Context of the requester for the selected tree.
%SECOND% (%SECONDS%)	Number of seconds. For example: 03, 54.
%SHORT_YEAR%	Short year number. For example: 97, 00.
%WINVER%	Windows version. For example: v3.11, v4.00.
%WORKSTATION_ID%	The name of the workstation object.
%WORKSTATION_TREE%	The name of the tree in which the workstation is located.

Macro	Description
%WORKSTATION_DN%	The distinguished name and context of the workstation object.
%YEAR%	Full year number. For example: 2000, 2001.

49.3.2 Unsupported Login Script Identifier Variables

The following is a list of login script macros that Application Launcher does not support:

ACCESS_SERVER
 AM_PM
 DAY_OF_WEEK
 DIALUP
 ERROR_LEVEL
 GREETING_TIME
 LOGIN_ALIAS_CONTEXT
 LOGIN_CONTEXT
 MACHINE
 MEMBER OF
 NOT MEMBER OF
 MONTH_NAME
 NETWARE_REQUESTER
 OFFLINE
 PASSWORD_EXPIRES
 REQUESTER_VERSION
 SHELL_TYPE
 SHELL_VERSION
 SMACHINE
 STATION
 USER_ID

49.4 eDirectory Attribute Macros

Application Launcher supports macros that pull information from the attributes of the currently logged-in user, the current Application object, or from the attributes of other eDirectory objects.

The following sections explain the macro syntax and provide examples:

- ◆ [Section 49.4.1, “Syntax,” on page 600](#)
- ◆ [Section 49.4.2, “Examples,” on page 601](#)

For information about using the macros in a valid Application object property, see [Section 49.8, “Application Object Properties Where Macros Can Be Used,” on page 605](#).

49.4.1 Syntax

eDirectory attribute macros use the following syntax:

`%object_name.container;eDirectory_attribute%`

Table 49-4 Macro Syntax

Element	Description
<code>%</code>	Flags the text as a macro. The entire macro must be enclosed in% characters.
<code>.object_name.container;</code>	Indicates the eDirectory object from which to read the attribute. To use an attribute from the currently logged in user, leave this element out. For example, <code>%CN%</code> would return the common name of the currently logged in user. To use an attribute from the Application object, replace <code>.object_name.container</code> with an asterisk (*). For example, <code>%;DN%</code> would return the distinguished name of the Application object.
<code>eDirectory_attribute</code>	Defines the attribute to be read. You can use the ConsoleOne Schema Manager (available from the Tools menu) to view an eDirectory object's available attributes.

49.4.2 Examples

The following table provides examples of eDirectory attribute macros.

Table 49-5 Macro Examples

Macro	Description
<code>%CN%</code>	Returns the common name of the currently logged-in user.
<code>%DN%</code>	Returns the distinguished name of the currently logged-in user.
<code>%Full Name%</code>	Returns the full name of the currently logged-in user. This is the name defined in User object > General tab > Identification page > Full Name field.
<code>%Given Name%</code>	Returns the first name of the currently logged-in user. This is the name defined in User object > General tab > Identification page > Given Name field.
<code>%Surname%</code>	Returns the last name of the currently logged-in user. This is the name defined in the User object > General tab > Identification page > Last Name field.
<code>%;DN%</code>	Returns the distinguished name of the Application object (the one in which the macro is being used).
<code>%;App:Caption%</code>	Returns the application icon title given the Application object. This is the title defined in the Application object > Identification tab > Icon page > Application Icon Title field.

Macro	Description
*;App:Path%	Returns the path to the application's executable as defined for the Application object. This is the path defined in the Application object > Run Options tab > Application page > Path to File field.
%*;App:GUID%	Returns the Application object's global unique identifier (GUID). This is the GUID defined in the Application object > Distribution Options tab > Options page > GUID field.
%.JSmith.Novell;Description%	Returns the description for the JSmith User object located in the Novell container. This is the description defined in the User object > General tab > Identification page > Description field.

49.5 Environment Variable Macros

Application Launcher supports all valid Windows 98 and Windows 2000/XP environment variables. The value of the variable must not exceed the length of the Application object name; otherwise, the variable fails.

Here are a few examples:

- ♦ NWLANGUAGE
- ♦ TEMP
- ♦ PATH

When using an environment variable, you must enclose it in % characters (%PATH%).

For information about using the macros in a valid Application object property, see [Section 49.8, “Application Object Properties Where Macros Can Be Used,” on page 605](#).

49.6 Language Variable Macros

To minimize the number of Application objects required to distribute the same application in different languages, you can use language variables to represent language-related information in MSI Application objects. Language variables are not supported for simple, AOT/AXT, terminal server, or Web applications.

For example, assume you want to distribute language-appropriate versions of the ZENworks Desktop Management Agent, which has a separate .msi file for each language. By default, the Desktop Management Agent's .msi files are installed to the Desktop Management server's `sys:\public\zenworks\zfdagent\language` directory, where *language* represents the various supported languages. In the Desktop Management Agent's Application object, you could include a language variable in the source path for the *language* directory. For example:

```
\\server\sys\public\zenworks\zfdagent\%LOCALE_SYS_NATIVE_LANG%\zfdagent.msi
```

During distribution, Application Launcher retrieves the requested language information from the workstation's system and substitutes it for the language variable.

As another example, assume you have an .msi file such as `app1.msi`. When installing a specific language, you must apply a language-specific transform (.mst) file. To accommodate this need,

you could create language directories for each of the .mst files and then use a variable for the language directory when specifying the transform path.

```
\\server\sys\public\zenworks\app1\%LOCALE_SYS_NATIVE_LANG%\app1.mst
```

Language variables can be used in the same MSI Application object properties that other macro variables (special Windows macros, environment variables, and so forth) can be used. For a list of these locations, see [Section 49.8, “Application Object Properties Where Macros Can Be Used,”](#) on [page 605](#)

The following table describes the available language variables.

Table 49-6 *Language Variable Macros*

Language Variable	Description
%LOCALE_SYS_DEFAULT_ANSI_CP%	Retrieves the American National Standards Institute (ANSI) code page associated with the system locale. If the locale does not use an ANSI code page, the value is 0. Example: 1252
%LOCALE_SYS_DEFAULT_OEM_CP%	Retrieves the original equipment manufacturer (OEM) code page associated with the system locale. If the locale does not use an OEM code page, the value is 1. Example: 437
%LOCALE_SYS_LANGID%	Retrieves the language identifier for the system locale. The language identifier is a standard international numeric abbreviation for the language in a country or geographical region. Example: 0409
%LOCALE_SYS_ABBR_LANG%	Specifies the abbreviated name of the system language. In most cases, it is created by taking the two-letter language abbreviation from the International Organization for Standardization (ISO) Standard 639 and adding a third letter, as appropriate, to indicate the sublanguage. Example: ENU
%LOCALE_SYS_ENG_LANG%	Specifies the full English name of the system language from ISO Standard 639. This is always restricted to characters that can be mapped into the ASCII 127-character subset. Example: English
%LOCALE_SYS_LANG%	Specifies the full localized name of the system language. This name is based on the localization of the product and might vary for each localized version. Example: English (United States)

Language Variable	Description
%LOCALE_SYS_ISO639_LANG%	Specifies the abbreviated name of the system language based only on ISO Standard 639. Example: en
%LOCALE_SYS_NATIVE_LANG%	Specifies the native name of the system language. Example: English
%LOCALE_USER_DEFAULT_ANSI_CP%	Retrieves the American National Standards Institute (ANSI) code page associated with the user locale. If the locale does not use an ANSI code page, the value is 0. Example: 1252
%LOCALE_USER_DEFAULT_OEM_CP%	Retrieves the original equipment manufacturer (OEM) code page associated with the user locale. If the locale does not use an OEM code page, the value is 1. Example: 850
%LOCALE_USER_LANGID%	Retrieves the language identifier for the user locale. The language identifier is a standard international numeric abbreviation for the language in a country or geographical region. Example: 0c09
%LOCALE_USER_ABBR_LANG%	Specifies the abbreviated name of the user language. In most cases, it is created by taking the two-letter language abbreviation from the International Organization for Standardization (ISO) Standard 639 and adding a third letter, as appropriate, to indicate the sublanguage. Example: ENA
%LOCALE_USER_ENG_LANG%	Specifies the full English name of the user language from ISO Standard 639. This is always restricted to characters that can be mapped into the ASCII 127-character subset. Example: English
%LOCALE_USER_LANG%	Specifies the full localized name of the user language. This name is based on the localization of the product and might vary for each localized version. Example: English (Australia)
%LOCALE_USER_ISO639_LANG%	Specifies the abbreviated name of the user language based only on ISO Standard 639. Example: en
%LOCALE_USER_NATIVE_LANG%	Specifies the native name of the user language. Example: English

49.7 Macro Precedence

To resolve conflicts with macro names, Application Launcher uses the following order of precedence:

- ◆ Application object macros
- ◆ Special Windows macros
- ◆ Login script macros
- ◆ eDirectory attribute macros
- ◆ Environment variable macros

The macro types are listed in order of precedence. For example, Application object macros take precedence over special Windows macros, special Windows macros take precedence over login script macros, and so forth.

49.8 Application Object Properties Where Macros Can Be Used

You can use macros when defining the Application object properties.

When entering a macro in a property field, you must enclose the macros in % characters (for example, %TARGET_PATH% or %*WINDISK%). This enables Application Launcher to recognize and treat the macro as a macro.

The following sections describe the fields on each page on which you can use macros:

- ◆ [Section 49.8.1, “Common Tab,” on page 605](#)
- ◆ [Section 49.8.2, “Distribution Options Tab,” on page 606](#)
- ◆ [Section 49.8.3, “MSI Tab,” on page 606](#)
- ◆ [Section 49.8.4, “Run Options Tab,” on page 607](#)

49.8.1 Common Tab

You can use macros in the following fields on the Common tab:

Table 49-7 *Using Macros on the Common Tab*

Field Name	Access to Field
<i>Drive to be mapped/path</i>	<i>Common tab > Drives/Ports page > Add</i>
<i>Port to be captured/printer or queue</i>	
<i>Run before uninstalling</i>	<i>Common tab > Uninstall Scripts page</i>
<i>Run after uninstalling</i>	

NOTE: To use macros in these two fields, you must be using a script engine rather than the Novell Client, as specified in the Script Engine Location field.

Field Name	Access to Field
<i>Package source path</i>	<i>Common tab > Sources > Add</i>

49.8.2 Distribution Options Tab

You can use macros in the following fields on the Distribution Options tab:

Table 49-8 *Using Macros on the Distribution Options Tab*

Field Name	Access to Field
<i>Key, Value name, and Value data stings</i>	<i>Distribution Options tab > Registry page</i>
<i>Section, Value name, and Value data strings</i>	<i>Distribution Options tab > INI Settings page</i>
<i>Source file</i>	<i>Distribution Options tab > Application Files page > Add > File or Directory</i>
<i>Target file</i>	
<i>Target directory</i>	
<i>Add text to the file</i>	<i>Distribution Options tab > Text Files page > Add > Change</i>
<i>Delete text from the file</i>	
<i>Find file and replace text in the file</i>	
<i>All Icon/Shortcut properties</i>	<i>Distribution Options tab > Icons/Shortcuts page</i>
	NOTE: the Icons/Shortcuts page is not available for MSI applications.
<i>Run before distribution</i>	<i>Distribution Options tab > Distribution Scripts page</i>
<i>Run after distribution</i>	
NOTE: To use macros in these two fields, you must be using a script engine rather than the Novell Client, as specified in the Script Engine Location field.	

49.8.3 MSI Tab

You can use macros in the following fields on the MSI tab:

Table 49-9 *Using Macros on the MSI Tab*

Field Name	Access to Field
<i>Transform path</i>	<i>MSI tab > Transforms > Add</i>
<i>MSI Patch file</i>	<i>MSI tab > Patches > Add</i>

49.8.4 Run Options Tab

You can use macros in the following fields on the Run Options tab:

Table 49-10 *Using Macros on the Run Options Tab*

Field Name	Access to Field
<i>Path to file</i>	<i>Run Options tab > Application page</i>
<i>Parameter</i>	
<i>Working directory</i>	
<i>Run before launching</i>	<i>Run Options tab > Launch Scripts page</i>
<i>Run after termination</i>	

NOTE: To use macros in these two fields, you must be using a script engine rather than the Novell Client, as specified in the Script Engine Location field.

49.9 Refreshing Macro Information

When Application Launcher starts, it caches the macro information defined for each Application object associated with the logged in user or workstation. To refresh macro information, the user must restart Application Launcher.

The Novell Client also caches the macro information. If users are authenticated through the Novell Client, they must not only restart Application Launcher but must also log out of eDirectory and log in again.

For example, if you are using the %DN% (distinguished name) eDirectory attribute macro in a folder name and you move a User object from one eDirectory container to another, the change is not be reflected on the user's workstation until he or she restarts Application Launcher and, if authenticated through the Novell Client, uses the Novell Client to log out and log in again.

Reference: Novell Application Launcher Tools

50

Novell® ZENworks® Desktop Management provides several tools that can be helpful as you manage applications for your users. These tools, located on the Tools > ZENworks Utilities > Application Launcher Tools menu in ConsoleOne®, are explained in the following sections:

- ♦ Section 50.1, “Export Application Object,” on page 609
- ♦ Section 50.2, “Show Inherited Applications,” on page 609
- ♦ Section 50.3, “Search and Replace,” on page 609
- ♦ Section 50.4, “Manage Distribution GUIDs,” on page 610
- ♦ Section 50.5, “AOT/AXT File Tools,” on page 611
- ♦ Section 50.6, “Create Virtual CD,” on page 612

50.1 Export Application Object

This tool enables you to export an Application object's properties to a new .aot or .axt file. You can perform a default export, which exports all general properties, INI settings, registry settings, text file changes, shortcuts, environment variables, and application files, or you can perform a custom export, which enables you to choose which settings you want exported.

- 1 In ConsoleOne, select the Application object.
- 2 Click the *Tools* menu > *ZENworks Utilities* > *Application Launcher Tools* > *Export Application Object*.
- 3 Select the export option you want, click *Next*, then follow the prompts to complete the export.

50.2 Show Inherited Applications

This tool enables you to see the applications that have been associated with a user or workstation, including applications associated with groups or containers that the user or workstation belong to. The applications are listed by mode of delivery, such as *Force run*, *App Launcher*, *Desktop*, *Start menu*, and *System tray*.

- 1 In ConsoleOne, select a User object or Workstation object.
- 2 Click the *Tools* menu > *ZENworks Utilities* > *Application Launcher Tools* > *Show Inherited Applications*.

50.3 Search and Replace

This tool lets you search and replace text strings in an Application object's general settings, registry settings, INI settings, copy files settings, icon settings, and text file settings. As settings are found that match your search criteria, you are given the choice of whether or not to replace the text string.

- 1 In ConsoleOne, select the Application object.

- 2 Click the *Tools* menu > *ZENworks Utilities* > *Application Launcher Tools* > *Search and Replace*.
- 3 In the *Search for* box, type the text string you are searching for.
- 4 In the *Replace with* box, type the text string you want to use in place of the text string you are searching for.
- 5 Select the *Match case* box to require the search results to match the case (uppercase and lowercase) of the text string you entered in the *Search For* box.
- 6 If you want to limit the Application object settings that are searched, click *Options*, deselect any settings you don't want searched, then click *OK* to return to the *Search and Replace* dialog box.
- 7 To start the search, click *Find Next*.

The dialog box expands to display the search results. If a setting is found that matches the search criteria, the setting's text is displayed in the *Text found* box.
- 8 To replace the setting's text with the text listed in the *Replace with* box, click *Replace*.

or

To replace the text for all settings that match the criteria, click *Replace All*.

or

To skip replacing the setting's text, click *Find Next*.
- 9 Repeat **Step 8** until all settings have been searched.

50.4 Manage Distribution GUIDs

The GUID Manager lets you change an application's GUID (global unique identifier). You can randomly generate a new GUID, manually specify a new GUID, or change the GUID to match another application's GUID. You might want to change an application's GUID for the following reasons:

- ◆ The Application object is accidentally deleted from Novell eDirectory™. When you re-create the Application object, it is randomly assigned a new, unique GUID. Because the new GUID causes the application to be redistributed to all users and workstations associated with the application, you use the GUID Manager to change the new GUID to the old GUID.
- ◆ You have multiple Application objects for the same application to enable fault tolerance, site lists, and so forth. You want to make sure that all Application objects have the same GUID so that the application is distributed only one time regardless of the Application object that is used. You use the GUID Manager to synchronize the GUIDs.

Changing the GUID of an undistributed application has no redistribution consequences. However, before you change the GUID of a distributed application, be aware that doing so might result in Application Launcher redistributing the application. In particular, redistribution occurs in the following situations:

- ◆ You give the application a new, unique GUID (one that is different from all other applications' GUIDs). A new, unique GUID always results in redistribution.
- ◆ You give the application the same GUID as another application that is not distributed on the workstation. For example, App1 has GUID 1234 and App2 has GUID 5678. You synchronize

App1 with App2 so that both applications now have 5678 as their GUIDs. Any workstations that had App1 but not App2 now has App1 redistributed to it.

To use the GUID Manager:

- 1 In ConsoleOne, select the Application object whose GUID you want to modify.
- 2 Click the *Tools* menu > *ZENworks Utilities* > *Application Launcher Tools* > *Manage Distribution GUIDs*.
Alternately, you can access the GUID Manager by opening the Application object, clicking *Distribution Options* > *Options* to display the Options page and then clicking the *Modify* button next to the *GUID* field.
- 3 In the *Applications to receive new GUID* list, add the applications that will be assigned the new GUID.

The list already contains the application you selected before opening the GUID Manager.

- 4 In the *GUID source* box, select the method you want to use to assign the new GUID:

Randomly generate new GUID: Use this option to have the GUID Manager randomly generate a new GUID. If you've added multiple applications to the Applications to Receive New GUID list, choose from the following two options:

- ♦ **All selected apps share same GUID:** Assigns the same randomly-generated GUID to all of the applications.
- ♦ **All selected apps get new GUID:** Assigns different randomly-generated GUIDs to each application.

Select GUID from existing application: Use this option to synchronize the GUIDs of all listed applications with an existing application. Click the *Browse* button to browse for and select the Application object whose GUID you want to share. The source application's GUID (the one you select using this option) does not change. Only the GUIDs for the applications displayed in the Application to *Receive new GUID* list change.

Manually specify new GUID: Use this option to define a specific GUID for all listed applications. Click *Modify* to display the Edit GUID dialog box used to define the GUID.

- 5 After you've defined the target applications and the GUID source, click *Change GUIDs*.

50.5 AOT/AXT File Tools

This tool converts an `.aot` file to an `.axt` file or an `.axt` file to an `.aot` file.

- 1 In ConsoleOne, click the *Tools* menu > *ZENworks Utilities* > *Application Launcher Tools* > *AOT/AXT File Tools*.
- 2 To convert an `.aot` file to an `.axt` file, click *AOT -> AXT*.
or
To convert an `.axt` file to an `.aot` file, click *AXT -> AOT*.
- 3 In the Open dialog box, browse for and select the file you want to convert, then click *Open*.
- 4 In the Save As dialog box, type a name for the new file, then click *Save As*. The correct extension (`.aot` or `.axt`) is added automatically.

50.6 Create Virtual CD

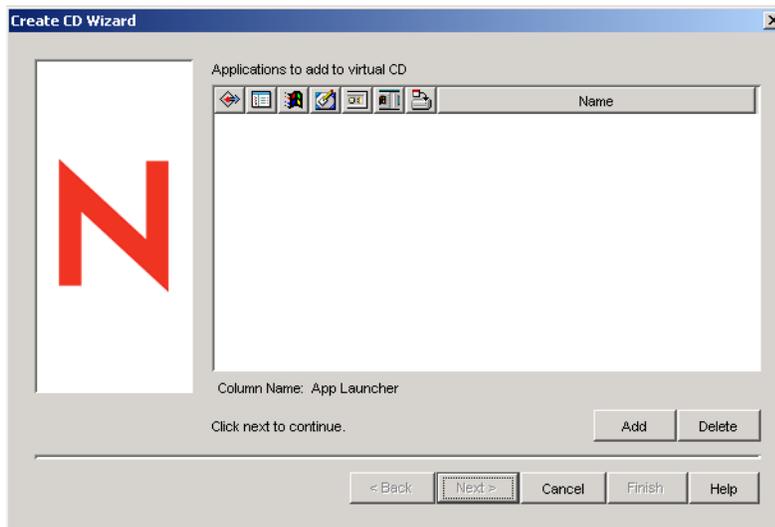
This tool lets you create a cache of applications that can be added to removable media (CD, Jaz drive, Zip drive, and so forth). The removable media functions as a second workstation cache, containing the Application object settings and application source files required to install and run the application. Any user whose workstation has Application Launcher installed can use the media to install and run the applications.

For example, you have a mobile user who seldom connects to the network but needs an application that you've distributed to other users. You burn a CD with the application and send it to the user. The user inserts the CD into his or her workstation's drive, Application Launcher reads the CD and displays the Application object's icon in the places you've configured (Application Launcher window, Start menu, desktop, and so forth). The user then launches the application, which is distributed to the workstation according to the Application object's configuration.

If the application includes any dependent applications (see [Chapter 37, "Advanced Distribution: Configuring Application Dependencies and Chains,"](#) on page 399), the dependent applications must be included on the removable media unless they have already been distributed to the user's workstation.

To create a cache on removable media:

- 1 In ConsoleOne, click the *Tools* menu > *ZENworks Utilities* > *Application Launcher Tools* > *Create Virtual CD* to start the Create CD Wizard and display the following page.



- 2 Click *Add*, then browse for and select the Application object for the application you want to distribute. Repeat this step to add additional applications.

If you add an MSI application, make sure the MSI source directory includes only MSI package files and subdirectories. The Create CD Wizard includes all files and subdirectories located in the source directory, causing Application Launcher to install all of the files and subdirectories regardless of whether or not they are part of the MSI package. The source directory is specified by the SOURCE_PATH variable on the Application object's Macros page (Common tab).

- 3 For each application you added to the list, select the locations where you want the Application object's icon to appear and select whether or not you want to force run and force cache the application. These settings are described below:

Force run:  Automatically runs the application. With a user-associated application, the application is run immediately after Novell Application Launcher™ starts or refreshes. With a workstation-associated application, the application is run immediately after the workstation starts up (initial startup or reboot) or after NAL Workstation Helper refreshes.

You can use the *Force run* option in conjunction with several other settings to achieve unique behaviors. For example, if you use the *Force run* option with the *Run application once* option (*Application* object > *Run Options* tab > *Applications* page), as soon as the Application object is distributed it runs one time and then is removed from the workstation. Or, suppose that you want to run the Application object immediately one time at a predetermined time. If so, select *Force run*, select the *Run application once* option on the Application page (*Application* object > *Run Options* tab), and define a schedule using the Schedule page (*Application* object > *Availability* tab).

App Launcher:  Adds the Application object's icon to the Application Window, the Application Explorer window, and the Application Browser window.

Start menu:  Causes Application Explorer, if running, to add the Application object to the Windows Start menu.

Desktop:  Causes Application Explorer, if running, to display the Application object's icon on the Windows desktop.

System Tray:  Causes Application Explorer, if running, to display the Application object's icon in the Windows system tray.

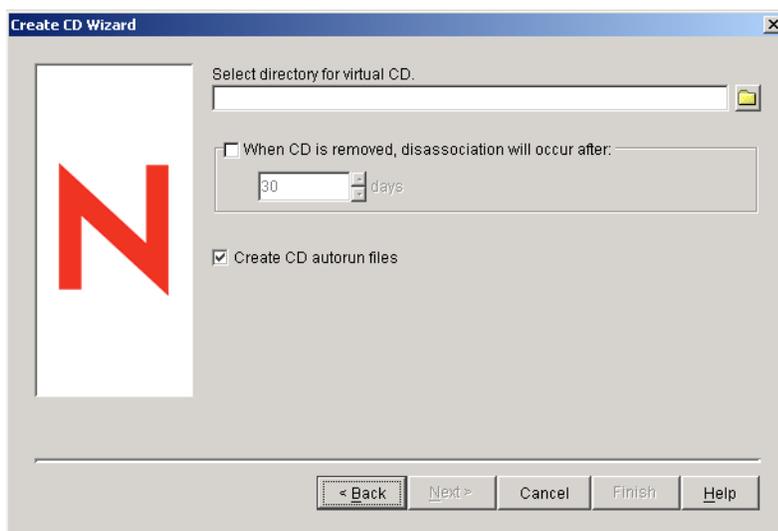
Quick Launch:  Causes Application Explorer, if running, to display the Application object's icon on the Windows Quick Launch bar.

Force cache:  Applies only if the Application object is marked as disconnectable (*Application* object > *Identification* tab > *Icon* page). With *Force cache* enabled, the first time the application is launched, Application Launcher copies the application source files and other files required for installation to the workstation's cache directory. The user can then install or verify the application while disconnected from eDirectory. The files are compressed to save space on the workstation's local drive.

The *Force cache* behavior for a removable media application (described in the previous paragraph) is different than the behavior for a network (eDirectory) application. With a network application, Application Launcher caches the application as soon as it becomes aware of the application, even if the application has not been launched yet. With a removable media application, Application Launcher does not cache the application until the application is launched the first time. This ensures that the user is not required to wait through the caching of multiple applications when Application Launcher initially reads the removable media.

NOTE: Do not use an asterisk character (*) in the name of the container that contains applications to be distributed using a virtual CD. Because Windows does not support the asterisk character in filenames, Windows changes the asterisk to an underscore character (_) and the force cache returns an error (D018).

- 4 After you've finished adding applications, click *Next* to display the following page.



- 5 Fill in the following fields:

Select directory for virtual CD: Select the location where you want to create the cache for the applications. Make sure the location you select has sufficient disk space for the cache. If enough disk space does not exist, you'll receive a message stating that an error has occurred.

You can also specify the path to the directory. If you specify a path to a network directory, you must use a mapped drive and not a UNC path. UNC paths are not supported.

When CD is removed disassociation will occur after: If you want to restrict the amount of time the user can continue to use an application after the CD or other media have been removed from the workstation, select this option, then select the number of days after which the disassociation occurs.

The time is measured from when the CD (or removable media) is removed from the user's workstation. The default time is 30 days. The valid range is 0 to 65535 days. Entering 0 instructs Application Launcher to disassociate the application the first time an Application Launcher refresh occurs after the CD (or removable media) is removed.

When the disassociation occurs, Application Launcher removes the application icon and the application files from the workstation.

Create CD autorun files: Select this option to create an `autorun.inf` file. The `autorun.inf` file initiates an Application Launcher refresh so that the Application objects included on the virtual CD are immediately displayed. If the user's workstation is not configured to autorun CDs, or if you use another media type, the user must manually refresh Application Launcher. This is done by right-clicking the Application Launcher icon and clicking *Refresh*.

In ZENworks 7, the method that the `autorun.exe` file uses to trigger an Application Launcher refresh changed from previous ZENworks versions. As a result, ZENworks 7 Application Launcher does not automatically refresh for virtual CDs created with previous ZENworks versions. Likewise, pre-ZENworks 7 versions of Application Launcher do not automatically refresh for ZENworks 7 virtual CDs.

When you use ZENworks 7 SP1 to create Virtual CDs, `autorun.exe` is automatically created in the directory where you are creating the application.

6 Click *Next*, review the settings, then click *Finish* to create the cache.

7 If necessary, copy the cache to the removable media (for example, burn the CD).

NOTE: Some CD-burning software does not allow for leading periods in filenames. When you create a virtual CD, ZENworks creates the virtual CD using a leading period in the application directory file pathname. If your CD-burning software does not allow for leading periods in filenames, and converts the period to another character such as an underscore, the virtual CD is not usable.

Currently, ZENworks does not support virtual CDs that are burned using the K3B CD-burning software.

Reference: Application Object Location

51

Each application you distribute is represented by an Application object that you create in Novell® eDirectory™. The Application object lets you configure distribution settings as well as other settings that determine how Novell Application Launcher™ manages the application.

Before you create an Application object, you need to decide where to locate the object in the eDirectory tree. The primary principle to follow is that an Application object should be placed in a container at the same site as the application’s users. The following two sections provide examples:

- ♦ [Section 51.1, “Single Site,” on page 617](#)
- ♦ [Section 51.2, “Multiple Sites,” on page 618](#)

51.1 Single Site

If your eDirectory tree encompasses only one site, you can place Application objects in any container. For example, if you have a small site consisting of one or two organizations, you might want to create a common APPS container.

Figure 51-1 Common APPS Container Under an Organizational Unit



If your site is divided into many organizations, you might want to create a general APPS container for your corporate-wide Application objects and then create APPS containers within each organization container for the organization-specific applications.

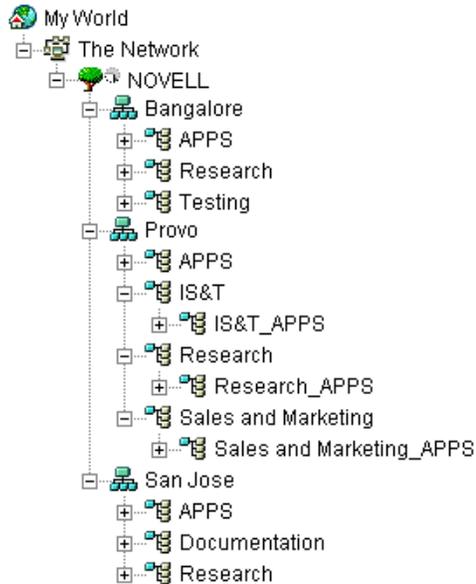
Figure 51-2 Common APPS Container and Organization-Specific APPS Containers



51.2 Multiple Sites

If your eDirectory tree encompasses several sites, we recommend that you place your Application objects in the tree at the same site as the users who access them, and ensure that the eDirectory partition for that site is replicated to one or more servers at that site. Typically, this means that you have APPS containers at multiple sites, as shown below.

Figure 51-3 APPS Containers at Three Different Sites



In the above example, the eDirectory tree has been established geographically, with each Organization container comprising a different site. Ideally, this is the most efficient way to organize your tree. If you have not organized your tree by geographical location, you can still place Application objects in the same location as the users who access them, but you need to discover these locations.

Undoubtedly, you have an application that you need to distribute to users at all your sites. In this case, you should create multiple Application objects (at least one at each site) for the application.

When giving users access to the application, you would associate the users with the Application object located at their site. Ensuring that users are accessing applications at their own site speeds user access to the applications and reduces cross-site network traffic.

If you have users who travel from site to site, you can set up site lists for any applications you want them to have access to at all sites. An application site list ensures that the user is accessing the application from the site where he or she is located, regardless of which Application object the user has been associated with. For more information about site lists, see [Chapter 48, “Reference: Application Object Settings,” on page 481](#).

Reference: Novell Application Launcher Authentication Hook

52

Novell® Application Launcher™ includes an authentication developer hook. You can write your own dynamic link library (DLL) that takes advantage of the Novell Application Launcher hook. In order to implement a Novell Application Launcher authentication hook, you should understand ZENworks® Desktop Management, Novell Application Launcher, and DLL development.

The DLL (typically created in C) must contain the function `NWAPPAAuthenticateExtern`. The DLL enables you to hook into the `zenlite.dll` library to authenticate application launches based on the criteria you choose. Novell Licensing Services, ZENworks OnDemand Services™, and other applications use this service to prompt user's input before launching an application.

As a practical example, once authenticated to a network, a user can retain a persistent connection, yet the DLL can be configured to prompt for any specified method (for example, a fingerprint scanner or other biometric device) to quickly reauthenticate if the user temporarily steps away from the workstation. Other examples of how the application hook can be used include the following:

- ♦ **Licensing:** You can regulate the number of users of an application. The authentication hook can track the application users in a database or text file and prevent the application from launching when the number of users reaches a predefined limit.
- ♦ **Specialized credentials:** You can prompt certain users who access an application for special credentials, such as biometric information.
- ♦ **Application grouping:** You can group applications, for example, that require purchase or billing information.

Refer to the following sections for development and implementation guidelines:

- ♦ [Section 52.1, “Creating an Authentication Hook,” on page 619](#)
- ♦ [Section 52.2, “Registering an Authentication Hook,” on page 620](#)
- ♦ [Section 52.3, “Authentication Function \(NWAPPAAuthenticateExtern\),” on page 620](#)

52.1 Creating an Authentication Hook

Write a standard DLL that exports a function called `NWAPPAAuthenticateExtern`.

The implementation is left up to you. If the function returns `FALSE`, the application does not launch (distribute or run), and the `NWAPPAAuthenticateExtern` function must display the reason to the user.

None of the DLL parameters passed in should be larger than 512 bytes (as designated by `MAX_DN_BYTES`).

For sample code and a compellable Microsoft Developer Studio Project file you can use to help create the DLL, download [nalauthhook.zip](http://www.novell.com/documentation/lg/zdpr/zdpradmndata/nalauthhook.zip) (<http://www.novell.com/documentation/lg/zdpr/zdpradmndata/nalauthhook.zip>).

52.2 Registering an Authentication Hook

- 1 Create a string value called Authentication DLL in the *HKCU\Software\NetWare\NAL\1.0* key of the Windows registry.
- 2 Enter the full path to the authentication hook DLL as the value of the Authentication DLL entry created in Step 1.

If the DLL library cannot be found, you receive the following error message:

```
Could not locate Third Party Authentication DLL\n%s
```

If the DLL is found inside the library but the *NWAPPAAuthenticateExtern* function cannot be found (such as when the function is not exported), you receive the message:

```
Could not locate Entry Point in Third Party Authentication DLL\n%s
```

NOTE: *%s* refers to a variable that is passed in by the user and prints out on the screen.

52.3 Authentication Function (NWAPPAAuthenticateExtern)

The *NWAPPAAuthenticateExtern* function provides an external authentication hook into the *zenlite.dll* library.

52.3.1 Syntax

```
#include <afxwin.h>

bool __stdcall NWAPPAAuthenticateExtern
{
    char          *szTreeName,
    char          *DSName,
    char          szWho
}
```

52.3.2 Parameters

szTreeName

(IN) Points to the name of the directory tree where the DLL is located on the client.

DSName

(IN) Points to the name of the application object provided to the user. For example, this application can request name and/or password or some other authentication method like a thumbprint for a biometric scanner.

szWho

(IN) Specifies the fully qualified user name.

52.3.3 Return Values

TRUE	Launch the application.
FALSE	Do not launch the application.

52.3.4 Remarks

The implementation of this function is left up to you. Values passed to `NWAPPAAuthenticateExtern` that exceed `MAX_DN_BYTES` bytes (`MAX_DN_BYTES = 512`) overflow the buffer and return an access violation.

IMPORTANT: Make certain you export the function.

Documentation Updates

J

This section contains information on documentation content changes that have been made in the *Administration* guide since the initial release of Novell® ZENworks® 7 Desktop Management (August 26, 2005). The information will help you to keep current on updates to the documentation.

All changes that are noted in this section were also made in the documentation. The documentation is provided on the Web in two formats: HTML and PDF. The HTML and PDF documentation are both kept up-to-date with the documentation changes listed in this section.

The documentation update information is grouped according to the date the changes were published. Within a dated section, the changes are alphabetically listed by the names of the main table of contents sections for ZENworks 7 Desktop Management.

If you need to know whether a copy of the PDF documentation you are using is the most recent, the PDF document contains the date it was published on the front title page.

The documentation was updated on the following dates:

- ◆ Section J.1, “July 27, 2007,” on page 623
- ◆ Section J.2, “June 15, 2007,” on page 624
- ◆ Section J.3, “May 4, 2007,” on page 624
- ◆ Section J.4, “February 9, 2007,” on page 625
- ◆ Section J.5, “January 2, 2007,” on page 626
- ◆ Section J.6, “November 17, 2006,” on page 626
- ◆ Section J.7, “November 6, 2006,” on page 626
- ◆ Section J.8, “July 14, 2006 (Support Pack 1),” on page 627
- ◆ Section J.9, “February 28, 2006,” on page 630
- ◆ Section J.10, “January 31, 2006,” on page 631
- ◆ Section J.11, “December 9, 2005,” on page 633
- ◆ Section J.12, “October 24, 2005,” on page 633
- ◆ Section J.13, “September 30, 2005,” on page 633

J.1 July 27, 2007

Updates were made to the following sections. The changes are explained below.

- ◆ Section J.1.1, “Reference: Application Object Settings,” on page 623
- ◆ Section J.1.2, “Reference: Novell Application Launcher Tools,” on page 624

J.1.1 Reference: Application Object Settings

The following updates were made in this section:

Location	Change
"Run After Distribution" on page 507 in the Distribution Scripts Page section	Added a note to explain the behavior of a <code>Run After Distribution</code> script when the application is associated to a workstation and a new user logs in for the first time.

J.1.2 Reference: Novell Application Launcher Tools

The following updates were made in this section:

Location	Change
Step 5 on page 614 in the Create Virtual CD section	Modified the content in the paragraphs describing Create CD autorun files : to clarify the behavior of ZENworks 7 Desktop Management with SP1.

J.2 June 15, 2007

Updates were made to the following sections. The changes are explained below.

- ◆ [Section J.2.1, "Controlling Rogue Processes," on page 624](#)

J.2.1 Controlling Rogue Processes

The following updates were made in this section:

Location	Change
Section 41.3.1, "Manually Modifying the Registry," on page 426	Revised Step 4 on page 426 and Step 8 on page 428 to clarify the process of creating an exception list and to explain the behavior that occurs when the <code>Default Action</code> setting is changed.
Section 41.3.2, "Creating an Application Object to Distribute the Registry Modifications," on page 428	Revised Step 5 on page 429 and Step 9 on page 430 to clarify the process of creating an exception list and to explain the behavior that occurs when the <code>Default Action</code> setting is changed.

J.3 May 4, 2007

Updates were made to the following sections. The changes are explained below.

- ◆ [Section J.3.1, "Reference: Application Object Settings," on page 624](#)
- ◆ [Section J.3.2, "Reporting Application Events," on page 625](#)

J.3.1 Reference: Application Object Settings

The following updates were made in this section:

Location	Change
Section 48.2.6, “Distribution Scripts Page,” on page 506	Added a note in “Script Execution Order” on page 508 to warn about script execution order when the <i>Distribute Always</i> option is selected.

J.3.2 Reporting Application Events

The following updates were made in this section:

Location	Change
Section 44.4.4, “Configuring the Reporting Servlet to be Used with an Apache Web Server,” on page 454	Added quotes to the line to be included at the bottom of the <code>httpd.conf</code> file (see Step 2).

J.4 February 9, 2007

Updates were made to the following sections. The changes are explained below.

- ♦ Section J.4.1, “Novell Application Launcher: Managing Authentication and File System Access,” on page 625
- ♦ Section J.4.2, “Reference: Application Object Settings,” on page 625

J.4.1 Novell Application Launcher: Managing Authentication and File System Access

The following updates were made in this section:

Location	Change
Section 23.3.2, “Windows 2000/XP (User-Associated Applications),” on page 297	Added a row to Table 23-2 to explain file system access when applications are launch from the server by a secure/unsecure user.

J.4.2 Reference: Application Object Settings

The following updates were made in this section:

Location	Change
Section 48.2.3, “Application Files Page,” on page 495	Added a note to clarify the designed default behavior: a parent folder cannot be deleted by an Application object unless all files and subfolders under it are deleted first.

J.5 January 2, 2007

Updates were made to the following sections. The changes are explained below.

- ♦ [Section J.5.1, “Authentication and File System Access for User-Associated Applications,” on page 626](#)

J.5.1 Authentication and File System Access for User-Associated Applications

The following updates were made in this section:

Location	Change
Table 23-2, “Windows 2000/XP (User-Associated Applications),” on page 297	Changed the details of the network server file system access for a normal launch of a user-associated application. Content originally listed was incorrect.

J.6 November 17, 2006

Updates were made to the following sections. The changes are explained below.

- ♦ [Section J.6.1, “Configuring User Settings,” on page 626](#)
- ♦ [Section J.6.2, “Configuring Workstation Settings,” on page 626](#)

J.6.1 Configuring User Settings

The following updates were made in this section:

Location	Change
Section 21.3, “Configuring User Settings,” on page 268	Added information that was previously missing from the documentation about a Launcher Configuration setting : “Attempt to go online during refresh.”

J.6.2 Configuring Workstation Settings

The following updates were made in this section:

Location	Change
Section 21.7, “Configuring Workstation Settings,” on page 277	Added information about two Launcher Configuration settings that was formerly missing: “Attempt to go online during refresh” and “Enable reading from removable cache.”

J.7 November 6, 2006

Updates were made to the following sections. The changes are explained below.

- ♦ [Section J.7.1, “Distribution: Simple Applications,” on page 627](#)

J.7.1 Distribution: Simple Applications

The following updates were made in this section:

Location	Change
Section 28.2, “Configuring the Application in eDirectory,” on page 326	Deleted the paragraph in this section that indicated ZENworks Application Objects support naming with special characters. ZENworks no longer supports naming Application Objects with special characters.

J.8 July 14, 2006 (Support Pack 1)

Updates were made to the following sections. The changes are explained below.

- ◆ [Section J.8.1, “Advanced Distribution: Transferring Applications Using BITS,” on page 627](#)
- ◆ [Section J.8.2, “Metering Software Licenses,” on page 627](#)
- ◆ [Section J.8.3, “Novell Application Launcher: Configuring Settings,” on page 628](#)
- ◆ [Section J.8.4, “Novell Application Launcher: Customizing Views,” on page 628](#)
- ◆ [Section J.8.5, “Novell Application Launcher: Installing and Starting,” on page 628](#)
- ◆ [Section J.8.6, “Novell Application Launcher: Managing the Cache,” on page 628](#)
- ◆ [Section J.8.7, “Reference: AdminStudio ZENworks Edition,” on page 629](#)
- ◆ [Section J.8.8, “Reference: Application Object Settings,” on page 629](#)
- ◆ [Section J.8.9, “Reference: ZENworks sNappShot,” on page 629](#)
- ◆ [Section J.8.10, “Reporting Application Events,” on page 629](#)

J.8.1 Advanced Distribution: Transferring Applications Using BITS

The following section was added to cover new functionality:

Location	Change
Chapter 34, “Advanced Distribution: Transferring Applications Using BITS,” on page 365	New section added.

J.8.2 Metering Software Licenses

The following updates were made in this section:

Location	Change
Section 45.1, “Installing Novell Licensing Services,” on page 469	Updated the links to information about installing Novell Licensing Services on NetWare 6.x servers and OES NetWare servers.

Location	Change
Section 45.2, "Creating Metered Certificates," on page 469	Added information about how to create the metered certificates required for software license metering.

J.8.3 Novell Application Launcher: Configuring Settings

The following updates were made in this section:

Location	Change
Section 21.3, "Configuring User Settings," on page 268	Provided best practice information for using the Unassociated Days to Uninstall Application Launcher configuration setting.
Section 21.7, "Configuring Workstation Settings," on page 277	Provided best practice information for using the Unassociated Days to Uninstall Application Launcher configuration setting.

J.8.4 Novell Application Launcher: Customizing Views

The following updates were made in this section:

Location	Change
Section 22.2, "Customizing the Application Browser View," on page 282	Added information about a new configuration file, axnalview.js, that you must modify in addition to the myapps.html file in order to customize the Application Browser view. Information was added specifically to Section 22.2.1, "File Location," on page 283 and Section 22.2.2, "Modifying the Myapps.html and Axnalview.js Files," on page 284 .

J.8.5 Novell Application Launcher: Installing and Starting

The following updates were made in this section:

Location	Change
Section 20.5, "Application Window Command Line Switches," on page 254	Added the /restrictonline=3 option and updated the descriptions to reflect the modified functionality.
Section 20.6, "Application Explorer Command Line Switches," on page 261	Added the /restrictonline=3 option and updated the descriptions to reflect the modified functionality.

J.8.6 Novell Application Launcher: Managing the Cache

The following updates were made in this section:

Location	Change
Section 24.1.2, “Install Cache,” on page 306	Added information about how Application Launcher, when running in remote mode, creates an application’s install cache on the local workstation prior to launching the application.

J.8.7 Reference: AdminStudio ZENworks Edition

The following updates were made in this section:

Location	Change
Chapter 46, “Reference: AdminStudio ZENworks Edition,” on page 473	Updated version information to AdminStudio 7.5 ZENworks Edition.
Section 46.2, “Using AdminStudio ZENworks Edition,” on page 474	Updated the AdminStudio screenshot to reflect the new version. Also revised information about where to get AdminStudio help.

J.8.8 Reference: Application Object Settings

The following updates were made in this section:

Location	Change
Section 48.2.10, “BITS Settings Page,” on page 518	New section added.
Section 48.3.1, “Application Page,” on page 520	Clarified the information under “Force Run As User If Application Is Workstation Associated” on page 522 to better explain that distribution of the application is done in the workstation’s system space while running of the application is done in the logged-in user’s space.

J.8.9 Reference: ZENworks snAppShot

The following updates were made in this section:

Location	Change
Chapter 47, “Reference: ZENworks SnAppShot,” on page 475	Added recommendation to use Windows Installer packages rather than snAppShot packages whenever possible. Included a link to an article explaining the reason for this recommendation.
Section 47.3, “Creating an Installation Package,” on page 477	Under Step 1 on page 477 , added information about the location of <code>snapshot.exe</code> on Windows and Linux servers. Also updated the snAppShot screenshot.

J.8.10 Reporting Application Events

The following section was added to cover new functionality:

Location	Change
Section 44.2, "Setting Up SNMP Trap Reporting," on page 448	Added Section 44.2.1, "Enabling Firewall Access," on page 448 to provide information about how to ensure communication between user workstations and the management console when using SNMP traps as the reporting method.
Section 44.4.6, "Enabling the XML Targets Policy," on page 455	Under Step 4 on page 457, corrected the Reporting Servlet's port number in the URL for a Linux server (OES Linux and SLES). The correct port number (8180) is now listed.

J.9 February 28, 2006

Updates were made to the following section. The changes are explained below.

- ◆ Section J.9.1, "Advanced Distribution: Pre-Installing Applications," on page 630
- ◆ Section J.9.2, "Reference: Application Object Settings," on page 630
- ◆ Section J.9.3, "Reporting Application Events," on page 631
- ◆ Section J.9.4, "Setting Up Database Reporting," on page 631

J.9.1 Advanced Distribution: Pre-Installing Applications

The following updates were made in this section:

Location	Change
Section 35.3, "Scheduling an Application Pre-Install," on page 380	Modified Step 4b on page 382 by adding additional, clarifying information about the <i>Time for selected dates</i> and the <i>Spread from start time</i> fields. Modified Step 5b on page 384 by adding additional, clarifying information about the <i>Spread from start time</i> field.

J.9.2 Reference: Application Object Settings

The following updates were made in this section:

Location	Change
Section 48.5, "Availability Tab," on page 537	Modified the <i>Time for selected dates</i> and the <i>Spread from start time</i> fields in "Specified Days" on page 553 to further clarify their function. Modified the <i>Spread from start time</i> field in "Range of Days" on page 554 to further clarify its function.
Section 48.2, "Distribution Options Tab," on page 489	Modified the <i>Time for selected dates</i> and the <i>Spread from start time</i> fields in "Specified Days" on page 511 to further clarify their function. Modified the <i>Spread from start time</i> field in "Range of Days" on page 512 to further clarify its function.

J.9.3 Reporting Application Events

The following updates were made in this section:

Location	Change
Chapter 44, "Reporting Application Events," on page 441	Revised the first sentence in the second paragraph to read: "Application Launcher can write events to the Sybase ODBC-compatible database...." Testing has shown that only Sybase is fully supported for ZENworks Application Management ODBC reporting.

J.9.4 Setting Up Database Reporting

The following updates were made in this section:

Location	Change
Section 44.1.1, "Installing the Sybase Database," on page 442	Deleted a section entitled "Using an ODBC-Compliant Database" because it implied that ODBC databases other than Sybase are supported in ZENworks Application Management. Testing has shown that only Sybase is fully supported for ZENworks Application Management ODBC reporting.
Section 44.1.1, "Installing the Sybase Database," on page 442	Modified the first sentence in the section to read: "The Sybase database included on the Novell ZENworks 7 Desktop Management CD is the only supported database for direct ODBC reporting in ZENworks Application Management." Testing has shown that only Sybase is fully supported for ZENworks Application Management ODBC reporting.

J.10 January 31, 2006

Updates were made to the following section. The changes are explained below.

- ◆ [Section J.10.1, "Configuring Applications to Use Reporting," on page 631](#)
- ◆ [Section J.10.2, "Establishing File System Access," on page 632](#)
- ◆ [Section J.10.3, "INI Settings," on page 632](#)
- ◆ [Section J.10.4, "Installing Novell Licensing Services," on page 632](#)
- ◆ [Section J.10.5, "Reference: Macros," on page 632](#)
- ◆ [Section J.10.6, "Registry Settings," on page 632](#)

J.10.1 Configuring Applications to Use Reporting

The following updates were made in this section:

Change	Location
Section 44.5, "Configuring Applications to Use Reporting," on page 458.	Added reference to Linux servers in the note at the end of Step 3 .

J.10.2 Establishing File System Access

The following updates were made in this section:

Change	Location
Section 28.3, "Establishing File System Access," on page 329.	Added reference to Linux servers, where file system rights also need to be assigned.

J.10.3 INI Settings

The following updates were made in this section:

Change	Location
Section 48.2.4, "INI Settings Page," on page 499.	Added information in the note to clarify that INI values will not roll back in the case of a failure during installation.

J.10.4 Installing Novell Licensing Services

The following updates were made in this section:

Change	Location
Section 45.1, "Installing Novell Licensing Services," on page 469.	Clarified text to indicate that license metering cannot be used either in an all-Windows or in an all-Linux environment.

J.10.5 Reference: Macros

The following updates were made in this section:

Change	Location
Chapter 49, "Reference: Macros," on page 593.	Added a note in the chapter introduction to indicate that using macros causes increased network traffic because the ZENworks macro lookup checks with eDirectory on the network before it checks the local environment.

J.10.6 Registry Settings

The following updates were made in this section:

Change	Location
Section 48.2.2, "Registry Page," on page 492.	Added information in the note to clarify that registry values will not roll back in the case of a failure during installation.

J.11 December 9, 2005

Page design was reformatted to comply with revised Novell documentation standards.

J.12 October 24, 2005

Updates were made to the following section. The changes are explained below.

- ♦ [Section J.12.1, “Reporting Application Events,” on page 633](#)

J.12.1 Reporting Application Events

The following updates were made in this section:

Location	Change
“Enabling the XML Targets Policy” on page 455.	Revised Step 4 to show the correct Reporting servlet URLs for Windows/NetWare and Linux.

J.13 September 30, 2005

Updates were made to the following section. The changes are explained below.

- ♦ [Section J.13.1, “Novell Application Launcher: Installing and Starting,” on page 633](#)

J.13.1 Novell Application Launcher: Installing and Starting

The following updates were made in this section:

Change	Location
Section 20.5, “Application Window Command Line Switches,” on page 254.	Deleted the <code>/restrictonline=3</code> switch. This has been disabled in the code due to modified product functionality.
Section 20.6, “Application Explorer Command Line Switches,” on page 261.	Deleted the <code>/restrictonline=3</code> switch. This has been disabled in the code due to modified product functionality.