



Application Management

The following sections provide information on ZENworks® for Desktops (ZfD) Application Management tasks you may need to perform and features you might want to use.

- ◆ [Chapter 4, “Understanding Application Launcher/Explorer,” on page 29](#)
- ◆ [Chapter 5, “Setting Up Application Launcher/Explorer,” on page 35](#)
- ◆ [Chapter 6, “Distributing Applications,” on page 53](#)
- ◆ [Chapter 7, “Uninstalling Applications,” on page 67](#)
- ◆ [Chapter 8, “Running in Disconnected Mode,” on page 73](#)
- ◆ [Chapter 10, “Organizing Applications into Folders,” on page 89](#)
- ◆ [Chapter 11, “Reporting on Application Management Events,” on page 95](#)
- ◆ [Chapter 12, “Metering Software Licenses,” on page 109](#)
- ◆ [Chapter 13, “SnAppShot,” on page 111](#)
- ◆ [Chapter 14, “Application Object Settings,” on page 115](#)
- ◆ [Chapter 15, “Macros,” on page 189](#)
- ◆ [Chapter 16, “Application Launcher Tools,” on page 199](#)
- ◆ [Chapter A, “Documentation Updates,” on page 203](#)

4

Understanding Application Launcher/Explorer

Novell® Application Launcher™ and Application Explorer (Application Launcher/Explorer) run on users' workstations to deliver the applications you've configured in NDS®.

Both Application Launcher and Application Explorer are 32-bit applications that support Windows* 95, Windows 98, Windows NT*, Windows 2000, and Windows XP. When a user starts Application Launcher/Explorer, Application Launcher/Explorer reads NDS to provide access to the applications the user has been given rights to.

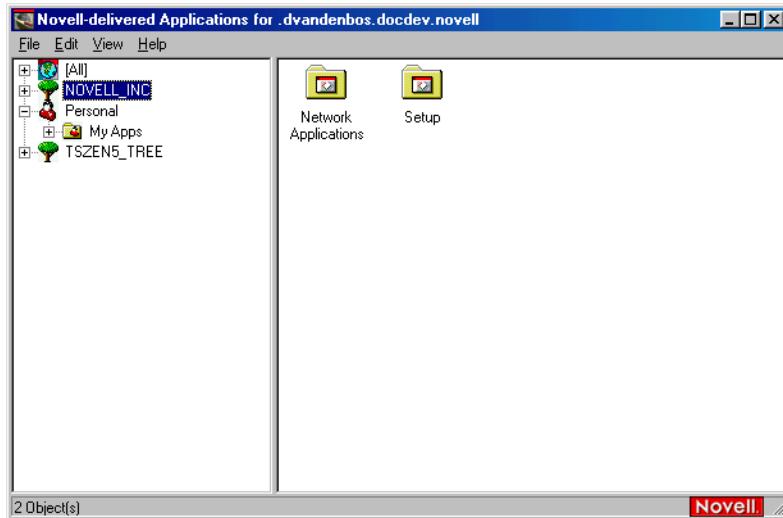
NOTE: If the user is not connected to NDS, Application Launcher/Explorer reads the user's local cache to see what applications the user has access to while disconnected from NDS.

Although Application Launcher and Application Explorer perform the same basic functions and both run on Windows 95/98 and Windows NT/2000/XP/XP, Application Launcher is designed to enable greater administrative control of the user's desktop and Application Explorer is designed to enhance and extend the desktop. As administrator, you determine whether users run Application Launcher or Application Explorer. The following sections provide information to help you understand Application Launcher/Explorer.

- ◆ “[Application Launcher](#)” on page 29
- ◆ “[Application Explorer](#)” on page 31
- ◆ “[Application Launcher/Explorer Service for Windows NT/2000/XP/XP](#)” on page 33
- ◆ “[Application Launcher/Explorer Workstation Helper](#)” on page 33

Application Launcher

Application Launcher, shown below, is a standalone window that can be accessed from the workstation's desktop.



NOTE: You can replace the Windows desktop with the Application Launcher window. For instructions, see "Using Application Launcher as the Windows Shell" on page 45.

Application Launcher Features

By default, the Application Launcher window is divided into two panes, much like Windows Explorer. In the left pane, Application Launcher 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.
- ◆ NDS 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.

When the 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.

Using the Application Launcher window, the user 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 NDS, before running the application Application Launcher may 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.
- ◆ Log in to NDS (or to another NDS tree). You can disable this functionality if desired.
- ◆ Disconnect Application Launcher from NDS so that the user can work offline. Disconnecting Application Launcher from NDS does not log the user out of NDS. You can disable this functionality if desired.

As administrator, you control the availability of certain features that enable users to do the following:

- ◆ Uninstall an application.
- ◆ Cache an application's installation files to the user's workstation so that the user can install or verify (fix) the application while disconnected from NDS.

For information about configuring Application Launcher, see "["Configuring Application Launcher/Explorer" on page 36](#)".

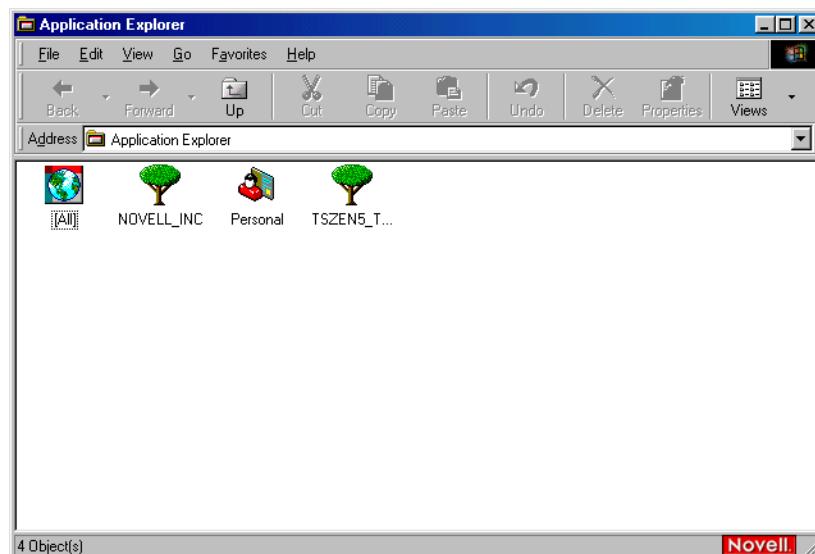
Application Launcher Files

Application Launcher consists of the following files:

- ◆ **NAL.EXE:** This is the program file that is used to execute Application Launcher. It is located on a server's SYS:PUBLIC directory. NAL.EXE serves as a "wrapper" file by performing some initialization functions, launching NALSTART.EXE, and then terminating. You can use various command line switches with NAL.EXE to control Application Launcher functionality. For information, see "["Application Launcher/Explorer Command Line Switches" on page 46](#)".
- ◆ **NALSTART.EXE:** This program updates the Application Launcher files on the user's workstation (if necessary), updates the Application Launcher/Explorer Windows Service (NALNTSRV.EXE) on a Windows NT/2000/XP/XP workstation (if necessary), registers and starts the Workstation Helper (ZENAPPWS.DLL) running under Workstation Manager, and launches NALWIN32.EXE.
- ◆ **NALWIN32.EXE:** This program reads NDS to build a list of the folders and applications that have been associated with the user and then displays the folders and applications in the Application Launcher window on the workstation.

Application Explorer

Application Explorer includes a standalone window, shown below, that can be accessed from the Windows desktop. The Application Explorer window is similar to the Application Launcher window.



The greatest benefit of Application Explorer, however, is its integration with the Windows desktop. In addition to being able to display Application objects in the Application Explorer window, you can configure Application objects to display on the Windows desktop, the Start menu, the system tray, and the Quick Launch tool bar.

Application Explorer Features

The Application Explorer window 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.
- ◆ NDS 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.

Using Application Explorer (either the window or one of the desktop integrations), the user can do the following:

- ◆ Run an application by double-clicking the application's icon. Depending on the application and how it is configured in NDS, before running the application Application Explorer may 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.
- ◆ Disconnect Application Explorer from NDS so that the user can work offline. Disconnecting Application Launcher from NDS does not log the user out of NDS. You can disable this functionality if desired.

As administrator, you control the availability of certain features that enable users to do the following:

- ◆ Uninstall an application.
- ◆ Cache an application's installation files to the user's workstation so that the user can install or verify (fix) the application while disconnected from NDS.

For information about configuring Application Explorer, see “[Configuring Application Launcher/Explorer](#)” on page 36.

Application Explorer Files

Application Explorer consists of the following files:

- ◆ **NALEXPLD.EXE:** This is the program file that is used to execute Application Launcher. It is located on a server's SYS:PUBLIC directory. NALEXPLD.EXE serves as a "wrapper" file by performing some initialization functions, launching NALSTART.EXE, and then terminating. You can use various command line switches with NALEXPLD.EXE to control

Application Explorer functionality. For information, see “[Application Launcher/Explorer Command Line Switches](#)” on page 46.

- **NALSTART.EXE:** This program updates the Application Explorer files on the user’s workstation (if necessary), updates the Application Launcher/Explorer Service (NALNTSRV.EXE) on a Windows NT/2000/XP/XP workstation (if necessary), registers and starts the Workstation Helper (ZENAPPWS.DLL) running under Workstation Manager, and launches NALDESK.EXE.
- **NALDESK.EXE:** This program reads NDS to build a list of the folders and applications that have been associated with the user and then displays the folders and applications in the designated locations (Application Explorer windows, desktop, system tray, and so forth) on the workstation.

Application Launcher/Explorer Service for Windows NT/2000/XP/XP

On Windows NT/2000/XP/XP, Application Launcher/Explorer includes a service (NALNTSRV.EXE) that does the following:

- **Distribution:** Ensures that applications can be distributed to the workstation regardless of the logged-in user’s Windows system credentials. For example, if a user logged in to a Windows 2000 workstation is a member of the Users group, he or she does not have the file system and registry rights required to install an application to the workstation. The Application Launcher/Explorer 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 running the application under 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 will be 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.

The Application Launcher/Explorer Service must be installed to the workstation by someone who has Administrator rights. By default, the service is installed with the Novell Client™, which also requires Administrator rights to install. During startup, Application Launcher/Explorer (through NALSTART.EXE) will also try to install and start the service if necessary; this will only be successful if the logged-in user has Administrator rights.

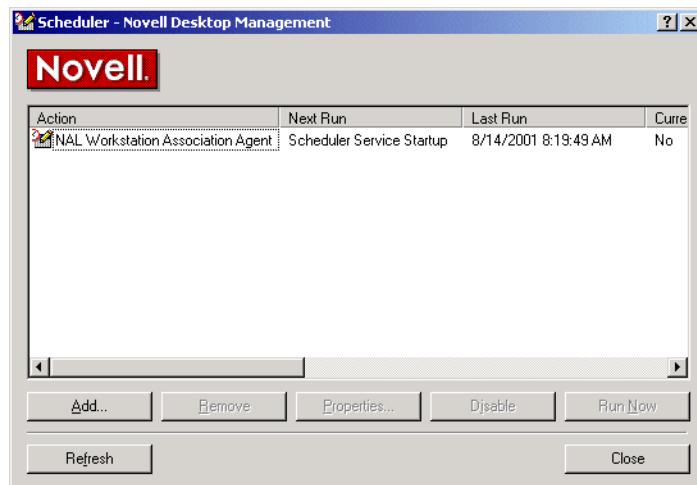
Application Launcher/Explorer Workstation Helper

In addition to associating Application objects with users, you can associate Application objects with workstations. This requires that the workstations be included in NDS as Workstation objects and that Workstation Manager be running on the workstations (see [Automatic Workstation Import and Removal](#) in *ZENworks for Desktops 3.2 Deployment Guide*).

When Application Launcher/Explorer starts, a Workstation Helper (ZENAPPWS.DLL) is loaded by Workstation Manager. The Workstation Helper accesses NDS through the Workstation object and passes Application Launcher/Explorer the list of applications associated with the Workstation

object. Application Launcher/Explorer then displays the workstation-associated applications just as it does the user-associated applications.

The Workstation Helper (ZENAPPWS.DLL) is installed on the workstation by Application Launcher/Explorer (through NALSTART.EXE) and is registered with Workstation Manager. As a result, Workstation Manager Scheduler, which is running on the workstation, adds an action called NAL Workstation Association Agent. You can view this scheduled action and others by double-clicking the Scheduler icon in the workstation's System tray.



In some cases, as in the above example, the Currently Running status may display as No. You should ignore this. As long as the NAL Workstation Association Agent is displayed in the list of actions, the Workstation Helper is running.

The Workstation Helper will refresh (reread NDS for changes to Application objects associated with the workstation) when 1) Application Launcher/Explorer is started, 2) Application Launcher/Explorer is manually refreshed, or 3) the Workstation Helper's scheduled refresh time occurs (see "[“Setting Configuration Options” on page 37](#)”).

5

Setting Up Application Launcher/Explorer

The following sections provide information to help you set up Novell® Application Launcher™/Explorer. For information about the differences between the two applications, see [Chapter 4, “Understanding Application Launcher/Explorer,” on page 29](#).

- ◆ “Starting Application Launcher/Explorer” on page 35
- ◆ “Configuring Application Launcher/Explorer” on page 36
- ◆ “Using Application Launcher as the Windows Shell” on page 45
- ◆ “Application Launcher/Explorer Command Line Switches” on page 46

Starting Application Launcher/Explorer

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

- ◆ “Rights for Windows NT/2000/XP/XP Users” on page 35
- ◆ “Manually Starting Application Launcher or Application Explorer” on page 35
- ◆ “Automating Application Launcher or Application Explorer Startup” on page 36

Rights for Windows NT/2000/XP/XP Users

Application Launcher/Explorer needs to be able to copy files to the workstation, write to the Windows registry, and so forth.

On Windows NT, users need to be members of the Users group to receive all the required rights.

On Windows 2000, users need to be members of the Power Users group.

Manually Starting Application Launcher or Application Explorer

You can run either Application Launcher or Application Explorer on a workstation. Do not run both on the same workstation.

To manually start Application Launcher or Application Explorer on a workstation:

- 1 Make sure that the Novell Client™ for Windows 95/98 (version 3.3 with SP4 or later) or the Novell Client for Windows NT/2000/XP/XP (version 4.8 with SP4 or later) is installed on the workstation.
- 2 To start Application Launcher, run NAL.EXE from the SYS:\PUBLIC directory on a server where you installed Application Management.

NAL.EXE copies Application Launcher files to the workstation and starts Application Launcher. For details about Application Launcher files, see [Chapter 5, “Setting Up Application Launcher/Explorer,” on page 35](#).

- 3 To start Application Explorer, Run NALEXPLD.EXE from the SYS:\PUBLIC directory on a server where you installed Application Management.

NALEXPLD.EXE copies Application Explorer files to the workstation and starts Application Explorer. For details about Application Explorer files, see [Chapter 5, “Setting Up Application Launcher/Explorer,” on page 35](#).

Automating Application Launcher or Application Explorer Startup

You can run either Application Launcher or Application Explorer on a workstation. Do not run both on the same workstation.

To automatically start Application Launcher each time a user logs in to NDS:

- 1 Make sure that the Novell Client for Windows 95/98 (version 3.3 with SP4 or later) or the Novell Client for Windows NT/2000/XP/XP (version 4.8 with SP4 or later) is installed on the workstation.

- 2 To use Application Launcher, enter the following line in the user’s login script:

```
@\\servername\sys\public\NAL.exe
```

where *servername* is the actual name of your network server.

You can enter this line in a container, profile, or user login script. If the login script also contains an entry for the Automatic Client Upgrade (ACU), make sure the NAL.EXE entry is listed before the ACU entry.

- 3 To use Application Explorer, enter the following line in the user’s login script:

```
@\\servername\sys\public\NALEXPLD.exe
```

where *servername* is the actual name of your network server.

You can enter this line in a container, profile, or user login script. If the login script also contains an entry for the Automatic Client Upgrade (ACU), make sure the NALEXPLD.EXE entry is listed before the ACU entry.

You can also add Application Launcher or Application Explorer to the Windows Startup folder. This enables Application Launcher/Explorer to start when the workstation is disconnected from NDS. You configure this option in ConsoleOne® and can have it apply to a single user, a group of users, or all users in a container. To do so, right-click a User, Group, or container object > click Properties > click the Application Launcher tab. On the Application Launcher page, click Edit > click the User tab > scroll to select the Auto-Start NAL When Disconnected option > set the option to Yes.

Configuring Application Launcher/Explorer

Application Launcher/Explorer includes configuration options you can use to determine how Application Launcher/Explorer works and the functionality that Application Launcher/Explorer exposes to users.

You use ConsoleOne to set configuration options at the container level, user level, or workstation level. When Application Launcher/Explorer searches the NDS tree for a user’s (or workstation’s) Application Launcher/Explorer settings, it starts with the User (or Workstation) object. If the

option has not been set for the User (or Workstation) object, Application Launcher/Explorer looks at the object's parent container to see if the option has been set at that level. If not, Application Launcher/Explorer continues up the NDS tree searching for the option's setting until it reaches a container object that has been designated as the top of the configuration tree. If it doesn't find any settings, then the setting is considered "unset" and the default setting is applied. Every configuration option has a default setting.

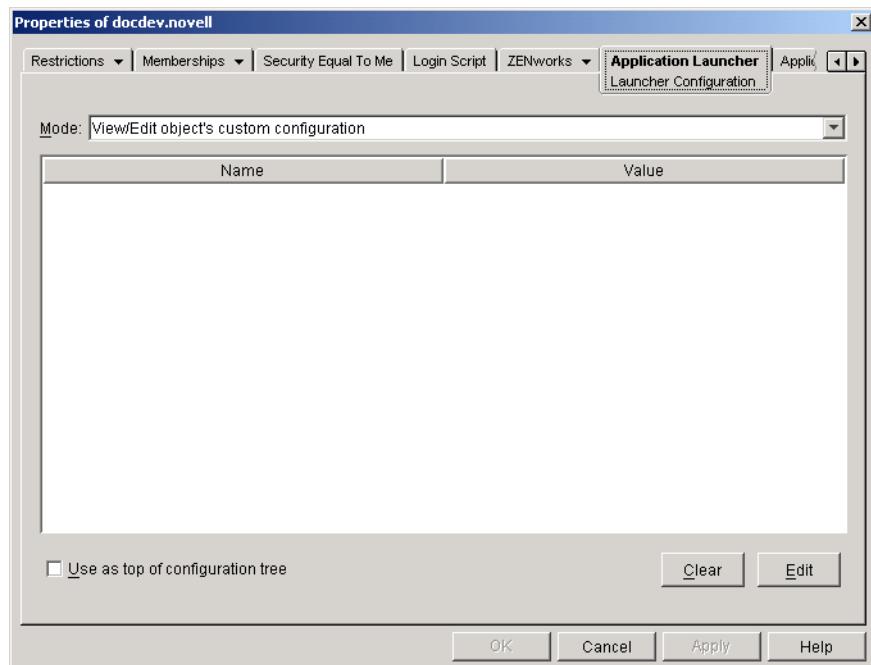
The following sections explain how to set configuration options and how to designate an object as the top of a configuration tree.

- ◆ “Setting Configuration Options” on page 37
- ◆ “Designating the Top of a Configuration Tree” on page 44

Setting Configuration Options

To set configuration options:

- 1 In ConsoleOne, right-click the User, Workstation, or container object to which you want to apply new settings > click Properties.
- 2 Click the Application Launcher tab 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.

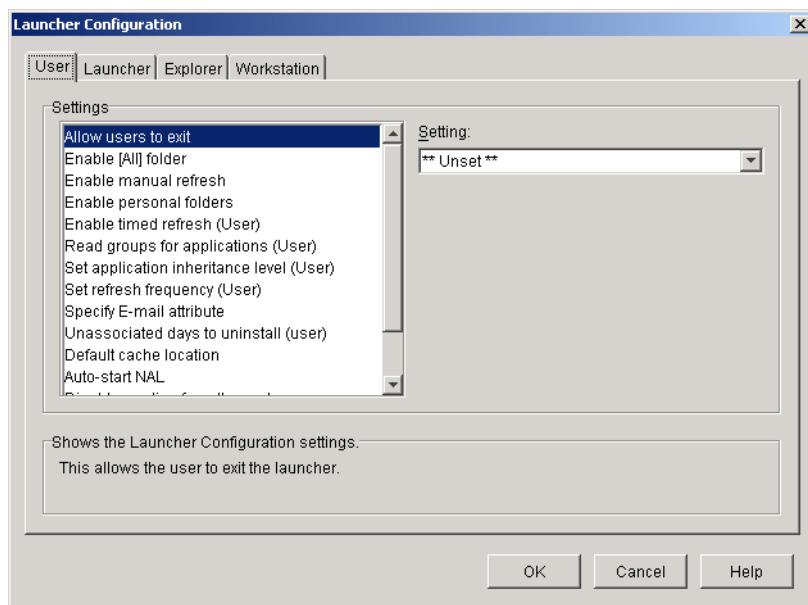
View/Edit Object's Custom Configuration (Default Mode): In this mode, the window lists all configuration options that have been set, or "customized," and lists the settings values. Options that are unset, meaning they receive their settings from their parent container (or higher), are not displayed.

This is the mode you must use to set a configuration option.

View Object's Effective Settings: In this mode, the window lists all configuration options with their effective settings. Effective settings may 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 NDS 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 what object a setting is being inherited from.

- 4 Click Edit to display the Launcher Configuration dialog box.



The Launcher Configuration dialog box has three or four tabs, depending on the type of object you are setting configuration options for: User, Launcher, Explorer, and Workstation.

User: Includes options that apply to both Application Launcher and Application Explorer.

Launcher: Includes options that apply to Application Launcher only.

Explorer: Includes options that apply to Application Explorer only.

Workstation: Includes options that apply to both Application Launcher and Application Explorer, but only when the user's workstation has been imported into NDS as a Workstation object and Workstation Manager is running on the workstation.

- 5 Click User > set the following options:

Allow Users to Exit: Specify whether or not to allow the user to exit Application Launcher/Explorer. If you allow the user to exit, you should let him or her know how to start Application Launcher/Explorer again.

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] applications folder in Application Launcher/Explorer. The [All] applications folder displays all applications the user has access to, regardless of the application's NDS 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: Specify whether or not the user can refresh Application Launcher/Explorer to distribute any changes since the last refresh. The Enable Manual Refresh and Enable Timed Refresh options 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 the user can create Application Launcher/Explorer folders and move the icons in them as he or she sees fit. Only Application object icons (or associated application icons or shortcuts) can be moved into personal folders. Use caution when offering the option if you want to exert more control and thus reduce support calls.

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: Specify whether or not Application Launcher/Explorer will automatically refresh the application so the user doesn't have to manually refresh icons.

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.

Read Group Objects for Applications: Specify whether or not you want Application Launcher/Explorer 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/Explorer 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/Explorer should search for applications associated with the user.

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

Set Refresh Frequency: Specify how often you want Application Launcher/Explorer to search NDS 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 (3600) is used if you select Unset and no parent container includes a customized setting.

Specify E-Mail Attribute: Specify the NDS 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 NDS attribute you specify here. Users must have the NDS rights required to read the selected attribute for all users defined as contacts.

The setting values include all the NDS attributes and Unset. The default value (Mailbox ID) 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/Explorer to uninstall an application when the user has been unassociated with the application.

The setting values are 0 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. The default value (0) 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 7, “Uninstalling Applications,” on page 67](#).

Default Cache Location: Select the drive where Application Launcher/Explorer should create the cache directory. The cache directory contains 1) the information required to uninstall the applications and 2) the files needed to install or repair the application when the workstation is disconnected from NDS (for example, a laptop).

The setting values are drives A through Z and the Windows Drive. The default is the Windows Drive.

Auto-Start NAL: Typically, Application Launcher/Explorer is started via a login script when a user logs in to NDS. Specify whether or not you want Application Launcher/Explorer to be included in the user's Startup folder so that it will be automatically started even when the user does not log in to NDS. This option will not cause Application Launcher/Explorer to run twice (from the login script and the Startup folder) when the user logs in to NDS.

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.

Disable Reading from the Cache: Specify whether or not you want to disable Application Launcher/Explorer's ability to read information from the cache directory (either local or on a removable media such as a CD, Jaz*, or Zip* drive). The main purpose of this option is to increase security by disabling a user's ability to launch or install applications from a removable media. With this option set to Yes, the Application Launcher/Explorer's Work Offline feature is also disabled, which means the user will not be able to work in disconnected mode.

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.

Always Evaluate Referrals: 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.

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

Enable Manage Applications Dialog: Specify whether or not you want to enable Application Launcher/Explorer's Manage Applications dialog box. This dialog box lets the user decide which disconnectable applications to include in the local cache and which cached applications to install or uninstall.

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 Automatic Icon Cleanup: Specify whether or not you want Application Launcher/Explorer, when exited, to remove application icons from the user's workstation. In general, you should set this option to Yes to have Application Launcher/Explorer clean up. However,

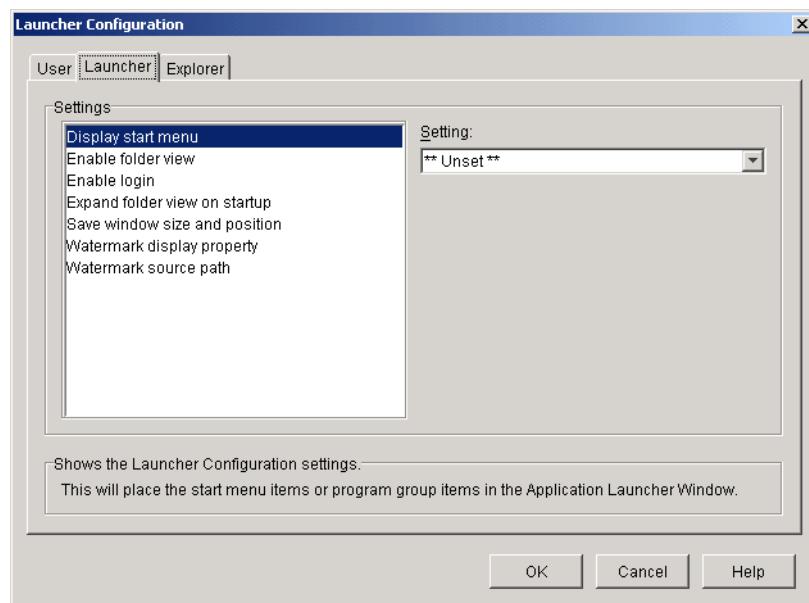
if the application is being run via a terminal server and multiple users log in with the same NDS username, you should disable this option. Otherwise, when one user exits Application Launcher/Explorer, the application icon's will disappear from all users' workstations.

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

OnDemand UsageServer: This setting applies only to ZENworks® OnDemand Services users. Specify the IP address or DNS hostname of the UsageServer to which the user's UsageClient will connect when Application Launcher/Explorer launches a usage-based application.

The setting values are Unset and Custom. To specify the UsageServer's address, select Custom > enter the UsageServer's IP address or DNS host name.

6 Click Launcher > set the following options:



Display Start Menu: Specify if and when you want Application Launcher to display the current Windows 95/98 or Windows NT/2000/XP/XP Start menu organization (including Programs and above) in the Application Launcher window.

The setting values are Show When Not Authenticated, Show When Authenticated, Show Always, Never Show, and Unset. The default value (Show When Not Authenticated) is used if you select Unset and no parent container includes a customized setting.

Enable Folder View: Specify whether or not Application Launcher displays folders in the Application Launcher 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 activate the Log In option found on the File menu in the Application Launcher window so the user can run the Novell Client to log in to NDS. Before you enable the Log In option, ensure that Application Launcher can find the login program (LOGINW32.EXE) on the user's workstation. Put the login program in the startup directory, working directory, or path.

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.

Expand Folder View on Startup: Specify whether or not you want Application Launcher to expand the entire folder tree when it starts.

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 you want Application Launcher to save its window size and position settings on a local drive.

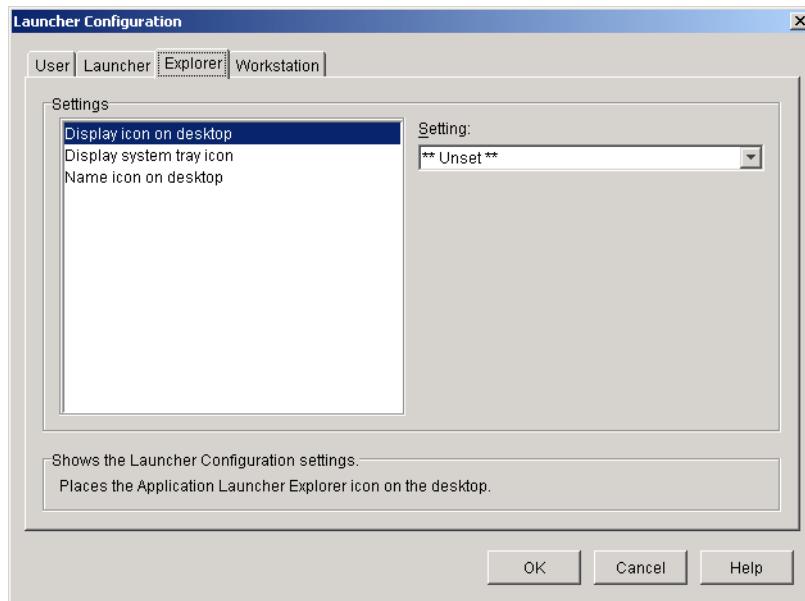
The setting values are Yes, No, and Unset. The default value (Yes) is used if you select Unset and no parent container includes a customized setting.

Watermark Display Property: This setting applies only if you've specified a watermark (Watermark Source Path setting). Select Default to have the watermark placed in the upper left corner in the Application Launcher 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. You can use any valid file path convention (for example, URL, mapped drive, or UNC path) and wallpaper graphic type (for example, .bmp, .gif, or .jpeg).

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.

- 7 Click Explorer > set the following options:



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 to access the applications that have been distributed to 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.

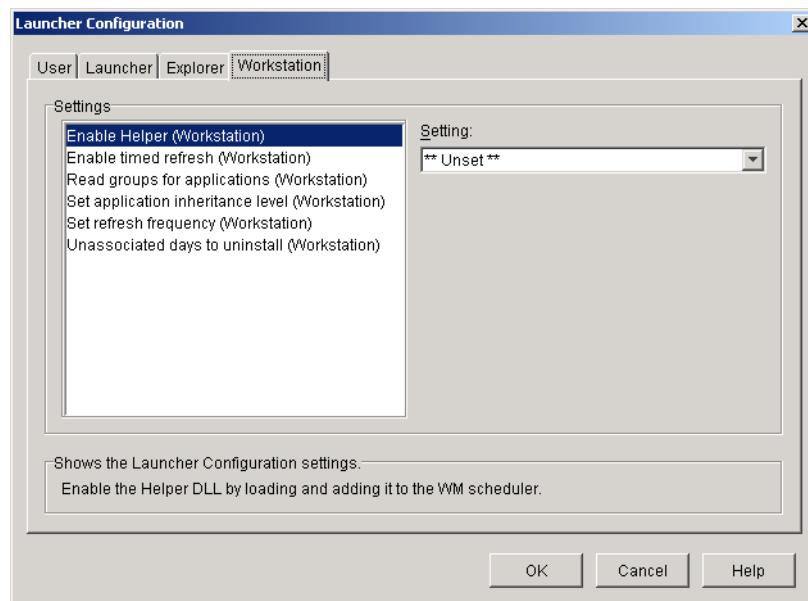
Display System Tray Icon: Specify whether or not you want the Application Explorer icon displayed in the Windows system tray. The Application Explorer icon lets users open the Application Explorer window to access the applications that have been distributed to them.

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.

Name Icon on Desktop: Use this setting to change the name of Application Explorer's desktop 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.

- 8 Click Workstation > set the following options:



Enable Helper: Specify whether or not you want to have the Workstation Helper installed and run by the Workstation Manager. The Workstation Helper must be running on a workstation in order to distribute workstation-associated applications to the workstation.

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 its information (from NDS) for any workstation-associated applications. This ensures that the most recent changes to the workstation-associated applications are reflected on the workstation. If timed refresh is not enabled, Workstation Helper will only read NDS when it starts or when Application Launcher/Explorer is manually refreshed.

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.

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: Applies only if the Enable Timed Refresh option is turned on. Specify how often you want Workstation Helper to read NDS for new or changed workstation-associated application information. 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 (3600) is used if you select Unset and no parent container includes a customized setting.

- 9 Click OK to return to the Launcher Configuration page > click OK to save the changes.

Designating the Top of a Configuration Tree

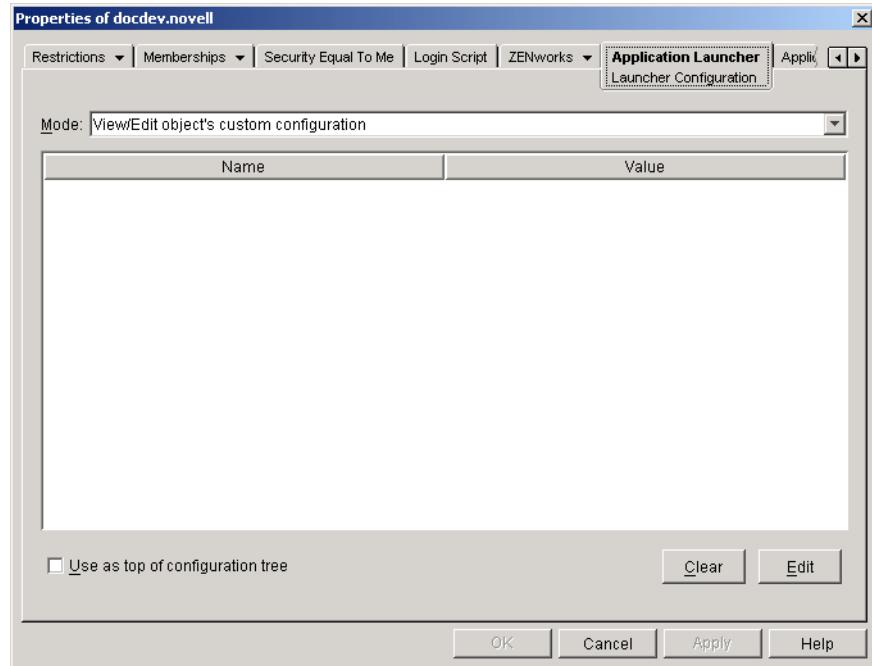
When Application Launcher/Explorer searches the NDS tree for a user's (or workstation's) Application Launcher/Explorer settings, it starts with the User (or Workstation) object. If the option has not been set for the User (or Workstation) object, Application Launcher/Explorer looks at the object's parent container to see if the option has been set at that level. If not, Application Launcher/Explorer continues up the NDS tree searching for the option's setting until it reaches an object that has been designated as the top of the tree.

You can designate User, Workstation, or container objects as the top of a configuration tree. For example, if you want Application Launcher/Explorer 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/Explorer 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 may want Application Launcher/Explorer 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 > click Properties.
- 2 Click the Application Launcher tab to display the Launcher Configuration page.



- 3** Check the Use as Top of Configuration Tree box to designate this object as the top of a configuration tree.
- 4** Click OK.

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 Windows 95/98 or Windows NT/2000/XP/XP workstations.

- ◆ “Setting Up Application Launcher as the Shell on Windows 95/98” on page 45
- ◆ “Setting Up Application Launcher as the Shell on Windows NT/2000/XP/XP” on page 46

Setting Up Application Launcher as the Shell on Windows 95/98

- 1** Copy the following files to the locations specified.

File	Copy From	Copy To
NALWIN32.EXE	Z:\PUBLIC	C:\NOVELL\CLIENT32
NALRES32.DLL	Z:\PUBLIC\NLS\ENGLISH	C:\NOVELL\CLIENT32\NLS\ENGLISH
NALBMP32.DLL	Z:\PUBLIC\NLS\ENGLISH	C:\NOVELL\CLIENT32\NLS\ENGLISH
NALEXP32.HLP	Z:\PUBLIC\NLS\ENGLISH	C:\NOVELL\CLIENT32\NLS\ENGLISH
NALEXP32.CNT	Z:\PUBLIC\NLS\ENGLISH	C:\NOVELL\CLIENT32\NLS\ENGLISH
NWAPP32.DLL	Z:\PUBLIC	C:\NOVELL\CLIENT32

- 2** Open the workstation's SYSTEM.INI file (typically, C:\WINDOWS\SYSTEM.INI) with a text editor.
- 3** Replace the SHELL=EXPLORER.EXE line with the following line:

```
SHELL=C:\NOVELL\CLIENT32\NALWIN32.EXE
```
- 4** Save and close the SYSTEM.INI file.
- 5** Restart Windows.

Setting Up Application Launcher as the Shell on Windows NT/2000/XP/XP

- 1** Copy the following files to the locations specified.

File	Copy From	Copy To
NALWIN32.EXE	Z:\PUBLIC	C:\WINNT\SYSTEM32
NALRES32.DLL	Z:\PUBLIC\NLS\ENGLISH	C:\WINNT\SYSTEM32\NLS\ENGLISH
NALBMP32.DLL	Z:\PUBLIC\NLS\ENGLISH	C:\WINNT\SYSTEM32\NLS\ENGLISH
NALEXP32.HLP	Z:\PUBLIC\NLS\ENGLISH	C:\WINNT\SYSTEM32\NLS\ENGLISH
NALEXP32.CNT	Z:\PUBLIC\NLS\ENGLISH	C:\WINNT\SYSTEM32\NLS\ENGLISH
NWAPP32.DLL	Z:\PUBLIC	C:\WINNT\SYSTEM32

- 2** Run REGEDIT.EXE and locate the following setting:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon

- 3** Modify the SHELL value as follows:

`SHELL=C:\WINNT\SYSTEM32\NALWIN32.EXE`

- 4** Close REGEDIT.EXE.

- 5** Restart Windows.

Application Launcher/Explorer Command Line Switches

The following command line switches can be used when running Application Launcher (NAL.EXE) or Application Explorer (NALEXPLD.EXE).

Switch	Description
SYNTAX:	This switch is valid for Application Launcher only. The /ns switch provides the same functionality for Application Explorer.
:	Skip the initial splash screen.
APPLICATION LAUNCHER EXAMPLE:	<code>nal :</code>

Switch	Description
SYNTAX: /a=".application_object"	Distributes and launches the specified application object.
or /a="tree:.application_object"	<i>Application_object</i> is the full object name of the application to run.
APPLICATION LAUNCHER EXAMPLES: nal /a=".snapshot.apps"	<i>Tree</i> is the name of the tree in which the object exists. The default tree is assumed if no tree name is specified.
APPLICATION EXPLORER EXAMPLES: nalexpld /a=".snapshot.apps"	NOTE: The quotation marks are required only if a space is used in the tree name or Application object name.
SYNTAX: /c="window_title"	Uses the specified text as the title for the Application Launcher/Explorer window.
APPLICATION LAUNCHER EXAMPLE: nal /c="Marketing Applications for %CN%"	The default window title is "Novell-delivered Applications for <i>User</i> ," where <i>User</i> is the distinguished name of the user.
APPLICATION EXPLORER EXAMPLE: nalexpld /c="Marketing Applications for %CN%"	The text specified can contain macros (variables) to display NDS information. The example, /c="Marketing Applications for %CN%" will display "Marketing Applications for JSmith" for a user logged in as JSmith.
SYNTAX: /dm	For information about macros, see Chapter 15, "Macros," on page 189 .
APPLICATION LAUNCHER EXAMPLE: nal /dm	Displays the installation progress of Microsoft* Windows Installer if the Windows Installer is not already installed on the workstation.
APPLICATION EXPLORER EXAMPLE: nalexpld /dm	
SYNTAX: /f	Instructs Application Launcher/Explorer not to read NDS to get the list of applications that have been associated with the logged-in user or the workstation. This is useful when using the /a, /v, or /l switches to simply distribute, verify, or uninstall a specific application.
APPLICATION LAUNCHER EXAMPLE: nal /f	This switch also requires you to use the /h switch.
APPLICATION EXPLORER EXAMPLE: nalexpld /f	

Switch	Description
SYNTAX: /h	Instructs Application Launcher/Explorer to hide its window from view when loaded.
APPLICATION LAUNCHER EXAMPLE: nal /h	This switch is required when using the /f switch and is useful with the /i, /l, and /v switches. APPLICATION EXPLORER EXAMPLE: nalexpld /h
SYNTAX: /i	Installs or updates the Application Launcher/Explorer files on the local hard drive, but does not start Application Launcher/Explorer.
APPLICATION LAUNCHER EXAMPLE: nal /i	This switch can be used on a secured NT workstation on which the SYSTEM32 directory is set to "read only" access for the user. APPLICATION EXPLORER EXAMPLE: nalexpld /i
SYNTAX: /l=".application_object"	Uninstalls the specified application object.
or	<i>Application_object</i> is the full object name of the application to run.
/l="tree:.application_object"	Tree is the name of the tree in which the object exists. The default tree is assumed if no tree name is specified.
APPLICATION LAUNCHER EXAMPLES: nal /l=".snapshot.apps"	NOTE: The quotation marks are required only if a space is used in the tree name or Application object name.
or	
nal /l="novell:.snapshot.apps"	
APPLICATION EXPLORER EXAMPLES: nalexpld /l=".snapshot.apps"	
or	
nalexpld /l="novell:.snapshot.apps"	
SYNTAX: /max	Displays the Application Launcher/Explorer window maximized when first loaded, overriding the window state from the previous running of Application Launcher.
APPLICATION LAUNCHER EXAMPLE: nal /max	
APPLICATION EXPLORER EXAMPLE: nalexpld /max	

Switch	Description
SYNTAX: /min	Displays the Application Launcher/Explorer window minimized when first loaded, overriding the window state from the previous running of Application Launcher/Explorer.
APPLICATION LAUNCHER EXAMPLE: nal /min	
APPLICATION EXPLORER EXAMPLE: nalexpld /min	
SYNTAX: /n	Displays the Application Launcher/Explorer window in its normal state (neither maximized nor minimized) when first loaded, overriding the window state from the previous running of Application Launcher/Explorer.
APPLICATION LAUNCHER EXAMPLE: nal /n	
APPLICATION EXPLORER EXAMPLE: nalexpld /n	
SYNTAX: /nbc	Stands for No Boot Choice. When updating Application Launcher/Explorer files, replaces the Yes/No buttons with an OK button when prompting the user to restart his or her workstation to update files in use.
APPLICATION LAUNCHER EXAMPLE: nal /nbc	
APPLICATION EXPLORER EXAMPLE: nalexpld /nbc	
SYNTAX: /nd	Suppresses the warning that a dial-up connection is present.
APPLICATION LAUNCHER EXAMPLE: nal /nd	
APPLICATION EXPLORER EXAMPLE: nalexpld /nd	
SYNTAX: /nodialup	Exits if a dial-up connection is present. Application Launcher/Explorer will not update any files nor run any other EXE if a dial-up connection is present.
APPLICATION LAUNCHER EXAMPLE: nal /nodialup	
APPLICATION EXPLORER EXAMPLE: nalexpld /nodialup	

Switch	Description
SYNTAX: /noupdate	Does not update any Application Launcher/Explorer files on the workstation. Application Launcher/Explorer will be run from the current directory, usually SYS:PUBLIC.
APPLICATION LAUNCHER EXAMPLE:	
nal /noupdate	
APPLICATION EXPLORER EXAMPLE:	
nalexpld /noupdate	
SYNTAX: /nows	Instructs Application Launcher/Explorer to not register the workstation helper (ZENAPPWS.DLL) and to remove the registration if it exists.
APPLICATION LAUNCHER EXAMPLE:	
nal /nows	This switch should only be used for temporary troubleshooting purposes. If you want to permanently disable the workstation helper, you should disable the Enable Helper configuration option (see “Setting Configuration Options” on page 37).
APPLICATION EXPLORER EXAMPLE:	
nalexpld /nows	
SYNTAX: /ns	This switch is valid for Application Explorer only. The: switch provides the same functionality for Application Launcher.
EXAMPLE:	Skips the initial splash screen.
nalexpld /ns	
SYNTAX: <i>/p="parameters"</i>	Instructs Application Launcher/Explorer to pass the specified parameters to the application being launched. It is only effective when used with /a or /v.
APPLICATION LAUNCHER EXAMPLE:	
nal /a=.snapshot.apps /p="/a /q"	
APPLICATION EXPLORER EXAMPLE:	
nalexpld /a=.snapshot.apps /p="/a /q"	
SYNTAX:	This switch is valid for Application Launcher only.
/s	Instructs Application Launcher to act like the Windows shell. In the Application Launcher window, the Exit the Application Launcher option on the File menu changes to Shutdown Windows and gives the user the standard Windows shutdown options.
EXAMPLE:	
nal /s	
NOTE: Application Launcher does not replace the Windows shell. If users minimize the Application Launcher window, they will have access to their normal desktop. If you want Application Launcher to replace the Windows shell, see “Using Application Launcher as the Windows Shell” on page 45 .	

Switch	Description
SYNTAX: /time=seconds	Instructs Application Launcher/Explorer to pause the specified number of seconds before continuing to load.
APPLICATION LAUNCHER EXAMPLE: nal /time=10	The default wait time is 3 seconds. Use this switch only if you want Application Launcher/Explorer to wait extra time before continuing to load. 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 being displayed at the same time and overlapping each other.
APPLICATION EXPLORER EXAMPLE: nalexpld /time=10	
SYNTAX: /u	Unloads Application Launcher/Explorer from memory after all applications have terminated.
APPLICATION LAUNCHER EXAMPLE: nal /u	
APPLICATION EXPLORER EXAMPLE: nalexpld /u	
SYNTAX: /v=" <i>application_object</i> " or /v=" <i>tree:.application_object</i> "	Verifies and launches the specified application. <i>Application_object</i> is the full object name of the application to run. <i>Tree</i> is the name of the tree in which the object exists. The default tree is assumed if no tree name is specified.
APPLICATION LAUNCHER EXAMPLE: nal /v=".snapshot.apps"	NOTE: The quotation marks are required only if a space is used in the tree name or Application object name.
or APPLICATION EXPLORER EXAMPLE: nalexpld /v=".snapshot.apps"	
or nalexpld /v="novell:.snapshot.apps"	

6

Distributing Applications

ZENworks® for Desktops (ZfD) Application Management lets you distribute applications to Windows* 95/98 and Windows NT*/2000/XP workstations. Because of the flexibility provided, you can also distribute individual files and software patches, or simply make changes to a workstation's Windows registry or configuration files.

The following sections provide information to help you understand the Application Management software components involved in distributing (and launching) applications, the NDS® and file system rights required by these software components, and the basic process of distributing an application. The final section references you to other places in the documentation that provide additional information about configuration settings you can use to control the distribution and launching of applications.

- ◆ “[Understanding the Components, NDS Rights, and File System Rights Used When Distributing and Launching Applications](#)” on page 53
- ◆ “[Distributing an Application](#)” on page 56
- ◆ “[Other Things You Can Do To Control the Distribution and Launching of Applications](#)” on page 66

Understanding the Components, NDS Rights, and File System Rights Used When Distributing and Launching Applications

Distribution of an application to a workstation, and the subsequent launching of the application on the workstation, can be handled by three different components: Application Launcher/Explorer, Application Launcher/Explorer Service for Windows NT/2000/XP/XP (NAL Service), or Application Launcher/Explorer Workstation Helper. For a description of each of these components, see [Chapter 4, “Understanding Application Launcher/Explorer,” on page 29](#).

Which components handle the distribution and launching depends on the following:

- ◆ The workstation's Windows version (95/98 or NT/2000)
- ◆ Whether the application is associated with the workstation or with a user logged into the workstation
- ◆ Whether the application is configured to run as the logged-in user or as the System user (Windows NT/2000/XP/XP only)

The tables in the following two sections describe which components are used to distribute and launch user-associated applications and workstation-associated applications.

- ◆ “[Distribution and Launching of User-Associated Applications](#)” on page 54
- ◆ “[Distribution and Launching of Workstation-Associated Applications](#)” on page 55

Because these components require NDS rights and file system rights to distribute and launch applications, and these rights can come from various sources, the table also lists which sources provide these rights. Knowing whose NDS and file system rights are providing access for the components can help you avoid distribution and launching problems caused by insufficient or incorrect rights.

Distribution and Launching of User-Associated Applications

Event	Windows 95/98	Windows NT/2000/XP/XP
Distribution	<ul style="list-style-type: none"> ◆ Distributed by Application Launcher/Explorer ◆ Access to NDS provided by the logged-in NDS user ◆ Access to the workstation's file system provided by default 	<ul style="list-style-type: none"> ◆ Distributed by NAL Service (Application Launcher/Explorer Service for Windows NT/2000/XP/XP) ◆ Access to NDS provided by the logged-in NDS user ◆ Access to the workstation's file system provided by the Windows System user
Launching (normal)	<ul style="list-style-type: none"> ◆ Launched by Application Launcher/Explorer ◆ Access to NDS provided by the logged-in NDS user ◆ Access to the workstation's file system provided by default 	<p>Run as logged-in user:</p> <ul style="list-style-type: none"> ◆ Launched by Application Launcher/Explorer ◆ Access to NDS provided by the logged-in NDS user ◆ Access to the workstation's file system provided by the logged-in Windows user <p>Run as secure/unsecure System user**:</p> <ul style="list-style-type: none"> ◆ Launched by NAL Service ◆ Access to NDS provided by the logged-in NDS workstation (through Workstation object) ◆ Access to the workstation's file system provided by the Windows System user
Launching (Force Run*)	Same as normal launch	Same as normal launch

* The Force Run setting causes the application to automatically run after being distributed. For information about configuring an application as Force Run, see “[Associating the Application Object with Users](#)” on page 62.

** The Secure System User and Unsecure System User settings apply to applications running on Windows NT/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 locally-installed applications even if they have limited file system rights on the workstation. If an application (running as a secure or unsecure System user) will be executed from a network location or will require access to a network location, the

workstation must also be logged in to NDS, which requires that the workstation have a Workstation object in NDS. For information about adding Workstation objects to NDS, see [Automatic Workstation Import and Removal](#) in the *ZENworks for Desktops 3.2 Deployment Guide* guide. For more about running an application as a secure or unsecure System user, see “[Environment Page](#)” on page 143.

Distribution and Launching of Workstation-Associated Applications

Event	Windows 95/98	Windows NT/2000/XP
Distribution	<ul style="list-style-type: none"> ◆ Distributed by Workstation Helper ◆ Access to NDS provided by the logged-in NDS workstation (through the Workstation object) ◆ Access to the workstation's file system provided by default 	<ul style="list-style-type: none"> ◆ Distributed by NAL Service via the Workstation Helper ◆ Access to NDS provided by the logged-in NDS workstation (through Workstation object) ◆ Access to the workstation's file system provided by the Windows System user
Launching (normal)	<ul style="list-style-type: none"> ◆ Launched by Application Launcher/Explorer ◆ Access to NDS provided by the logged-in NDS workstation (through the Workstation object) ◆ Access to the workstation's file system by default 	<p>Run as logged-in user:</p> <ul style="list-style-type: none"> ◆ Launched by Application Launcher/Explorer ◆ Access to NDS provided by the logged-in NDS user ◆ Access to the workstation's file system provided by the logged-in Windows user <p>Run as secure/unsecure System user**:</p> <ul style="list-style-type: none"> ◆ Launched by NAL Service ◆ Access to NDS provided by the logged-in NDS workstation ◆ Access to the workstation's file system provided by the Windows System user

Event	Windows 95/98	Windows NT/2000/XP
Launching (Force Run*)	<p>Launched by Workstation Helper</p> <p>Access to NDS provided by the logged-in NDS workstation (through the Workstation object)</p> <p>Access to the workstation's file system provided by default</p>	<p>Run as logged-in user:</p> <ul style="list-style-type: none"> ◆ Launched by NAL Service ◆ Access to NDS provided by the logged-in NDS workstation (through the Workstation object) ◆ Access to the workstation's file system provided by the Windows System user <p>Run as secure/unsecure System user:</p> <ul style="list-style-type: none"> ◆ Launched by NAL Service ◆ Access to NDS provided by the logged-in NDS workstation ◆ Access to the workstation's file system provided by the Windows System user

* The Force Run setting causes the application to automatically run after being distributed. For information about configuring an application as Force Run, see “[Application Launcher Features](#)” on page 30.

** The Secure System User and Unsecure System User settings apply to applications running on Windows NT/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 file system rights on the workstation. For more information, see “[Environment Page](#)” on page 143.

Distributing an Application

The process of distributing an application consists of four main tasks:

1. Create the application's installation package and copy it to a network location.
2. Create an Application object in NDS using the installation package.
3. Define the system requirements that a workstation must meet before Novell® Application Launcher™/Explorer will distribute the application to it.
4. Associate the Application object with the users or workstations you want to distribute the application to.

This process assumes that you have already set up Application Launcher/Explorer on users' workstations. Application Launcher/Explorer displays, installs, and configures the application on the workstation after you complete the tasks listed above. If you have not installed Application Launcher/Explorer, see [Chapter 5, “Setting Up Application Launcher/Explorer,”](#) on page 35.

The following sections provide instructions for completing the tasks listed above:

- ◆ “[Creating an Application's Installation Package](#)” on page 57
- ◆ “[Creating an Application Object](#)” on page 58
- ◆ “[Defining the Application's System Requirements](#)” on page 61

- ◆ “[Associating the Application Object with Users or Workstations](#)” on page 62

HINT: After you create an Application object, there are many different Application object property settings you can configure to better manage the application. The above tasks provide information about the basic settings you must configure in order to distribute the application. For information about additional settings you can configure, see [Chapter 14, “Application Object Settings,” on page 115](#).

Creating an Application’s Installation Package

An installation package consists of the files and information needed to install the application on a workstation. The following sections explain the options available to you to help you create an application’s installation package:

- ◆ “[Manually Creating an Installation Package](#)” on page 57
- ◆ “[Creating a snAppShot Installation Package](#)” on page 57
- ◆ “[Creating a Microsoft Windows Installer Package](#)” on page 58

Manually Creating an Installation Package

Some applications require few files to be copied to the workstation and few (or no) changes to the workstations configuration (registry, INI files, environment variables, and so forth). The installation package for these applications may very well consist of only one or two files. In this case, you can copy the files to a network directory. After creating the NDS object for the application, you can configure the Application object to instruct Application Launcher/Explorer to simply run the application from the network directory or to first copy the files to the workstation and then run the application locally. You can also configure the Application object to modify any workstation settings required by the application.

To manually create an installation package:

- 1 Create a network directory for the application’s files.
- 2 Copy the files to the network directory.

Creating a snAppShot Installation Package

Some applications require many files to be installed and many changes to be made to the workstation. This can require a complex installation package.

ZENworks® for Desktops (ZfD) Application Management includes the snAppShot™ utility to automate the process of creating complex installation packages. You run snAppShot on a workstation that has never had the application installed to it. SnAppShot records the workstation’s state (files and configuration) before you install the application and after you install the application. From this information, snAppShot creates an installation package consisting of an Application object template file (.AOT or .AXT file) and the application’s files (.FIL files). This installation package is saved to a network directory you specify. When you create the NDS object for the application, the .AOT file is used. When Application Launcher/Explorer distributes the application to a workstation, it uses the .FIL files as well as the configuration information (registry changes, INI file changes, and so forth) stored in the Application object.

To use snAppShot to create an installation package:

- 1 Set up a "clean" workstation. This needs to be a workstation where the application has never been installed.
- 2 Run snAppShot (SNAPSHOT.EXE) from the SYS:\PUBLIC\SNAPSHOT directory.

- 3** Follow the on-screen prompts to create the installation package. For more detailed information about snAppShot, see [Chapter 13, “SnAppShot,” on page 111](#).

Creating a Microsoft Windows Installer Package

Microsoft* Windows* 2000 (and patched versions of Windows 95/98 and Windows NT* 4.0) include an installation and configuration service referred to as Microsoft Windows Installer. Microsoft Windows Installer, similar to Application Launcher/Explorer, installs applications to a workstation from an installation package consisting of a .MSI file and various support files.

ZfD Application Management enables you to distribute applications that are installed through Windows Installer. Rather than using a snAppShot .AOT/.AXT file when creating the Application object, you simply use the application’s .MSI file. Distributing .MSI applications through Application Launcher/Explorer enables you to control who has access to the application while still gaining the benefits associated with the Windows Installer (such as on-demand installation, transforms, and source resiliency).

To use a Windows Installer package:

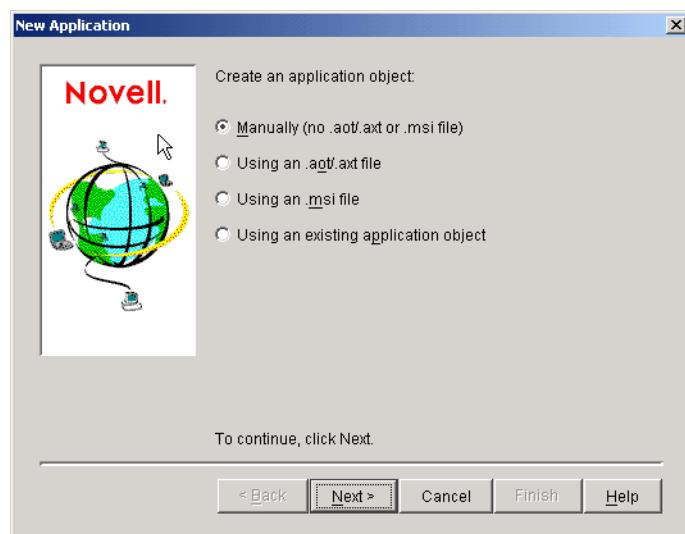
- 1** Create the installation package in a network location by running the application’s setup program using the administrative option. For more information, see your application’s installation documentation.

or

Copying the application’s preconfigured .MSI files and supporting files to a network location (if this method is supported by the application).

Creating an Application Object

- 1** In ConsoleOne®, right-click the container where you want to create the Application object > click New > click Object to display the New Object dialog box.
- 2** Click App:Application > OK to display the New Application dialog box.



- 3** Select the option you want to use to create the Application object:

Manually (No .AOT/.AXT File or .MSI File): Lets you enter a path to the application's executable file. After the Application object is created, you can modify additional properties if necessary. Skip to “[Manually Creating an Application Object](#)” on page 59.

Using an .AOT/.AXT File: Lets you specify an .AOT or .AXT file you've created with snAppShot. The .AOT or .AXT file is used to populate the Application object's property fields. Skip to “[Using an .AOT/AXT File to Create an Application Object](#)” on page 60.

Using an .MSI File: Lets you specify an .MSI file. The .MSI file is used to populate the Application object's property fields. Skip to “[Using an .MSI File to Create an Application Object](#)” on page 60.

Using an Existing Application Object: Lets you make a copy of an existing Application object. Skip to “[Using an Existing Application Object to Create a New Application Object](#)” on page 60.

Manually Creating an Application Object

1 In the New Application dialog box, select Manually (No .AOT/.AXT File or .MSI File) > click Next.

2 In the Object Name box, type a name for the Application object.

3 In the Path to Executable box, enter the path to the location from which the application's executable file will be run. Include the executable file in the path.

IMPORTANT: If you plan for users to run the application from the network or the file is already located on the users' workstations, the path should be to the actual executable file on the network or the workstation. If you plan to have Application Launcher/Explorer copy the file to a target directory on the workstation, the path must be to the executable file in the target directory.

4 Click Finish to create the Application object.

5 If Application Launcher/Explorer will need to copy files to the workstation, right-click the Application object > click Properties > complete the remaining steps.

or

If the application will be run from the network or is already located on the workstation, skip the remain steps and continue with “[Defining the Application's System Requirements](#)” on page 61.

6 Click the Distribution Options tab > Application Files.

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

7 Click Add > File > fill in the following fields.

Source File: Select the file you want copied to the workstation (for example, J:\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).

8 Click OK to add the file to the list.

9 Repeat Step 7 and Step 8 for each file to be copied.

10 When finished adding files, click OK to save the Application object information.

11 Continue with “[Defining the Application's System Requirements](#)” on page 61 to define the application's system requirements.

Using an .AOT/AXT File to Create an Application Object

1 In the New Application dialog box, select Using an .AOT/AXT File > click Next.

2 Enter the path to the .AOT or .AXT file > click Next.

HINT: The file should be in the network location you saved it to when creating it with snAppShot.

3 If necessary, 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 to create the installation package. You can change the name if you want. If another Application object in the same container has the same name, you need to select a different name. No two objects in the same container can have the same name.

SOURCE_PATH: This field defaults to the location where the application's installation files (.FIL) files were stored when running snAppShot to create the installation package. You should verify that the path is correct.

TARGET_PATH: This field defaults to the location where the application was installed when running snAppShot to create the installation package. You should verify that this is the workstation directory where you want the application installed.

4 Click Next.

5 Review the Application object settings > click Finish to create the Application object.

6 Continue with “[Defining the Application’s System Requirements](#)” on page 61 to define the application’s system requirements.

Using an .MSI File to Create an Application Object

1 In the New Application dialog box, select Using an .MSI File > click Next.

2 Enter the path to the .MSI file > click Next.

HINT: The file should be in the network location where you installed or copied the application.

3 If necessary, 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 to create the installation package. You can change the name if you want. If another Application object in the same container has the same name, you need to select a different name. No two objects in the same container can have the same name.

SOURCE_PATH: This field defaults to the administrative installation location defined in the .MSI file. You should verify that the path is correct.

4 Click Next.

5 Review the Application object settings > click Finish to create the Application object.

6 Continue with “[Defining the Application’s System Requirements](#)” on page 61 to define the application’s system requirements.

Using an Existing Application Object to Create a New Application Object

1 In the New Application dialog box, select Using an Existing Application Object > click Next.

2 Browse for and select the Application object you are duplicating > click Next.

3 Fill in the following fields to customize the new Application object.

Object Name: Specify a name for the new Application object.

SOURCE_PATH: Specify the network location where the application's installation files are located.

TARGET_PATH: Specify the workstation location where you want to install the application.

4 Click Next.

5 Review the Application object settings > click Finish to create the Application object.

6 Continue with “[Defining the Application’s System Requirements](#)” on page 61 to define the application’s system requirements.

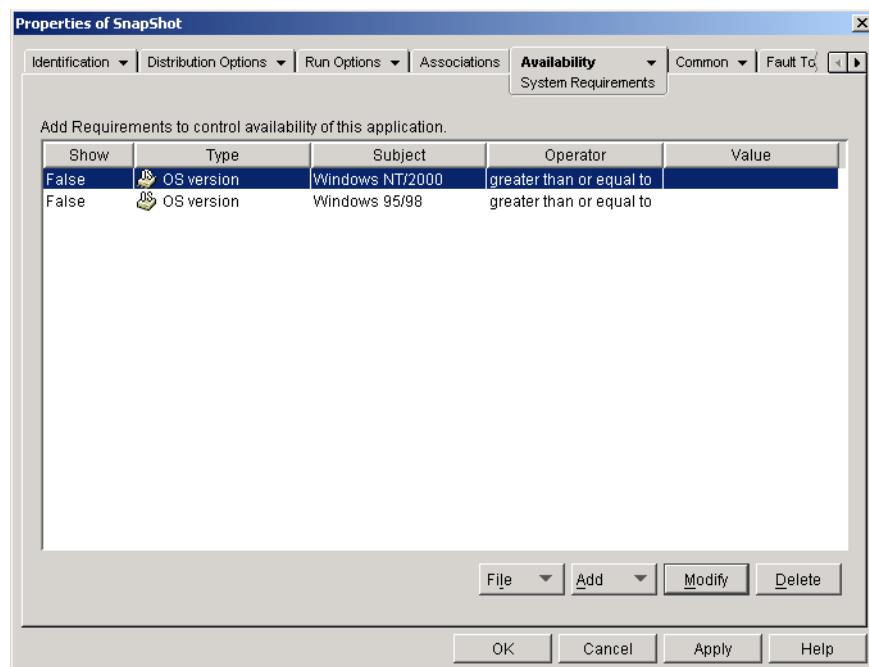
Defining the Application’s System Requirements

To ensure that Application Launcher/Explorer does not distribute the application on a workstation that cannot support the application, you need to define the application’s system requirements. An application will not even be displayed on users’ workstations until you do so.

For example, the application may run on Windows 95/98 but not on Windows NT/2000/XP. In this case, you would want to set one of the system requirements to be Windows 95/98. Or, the application may require 100 MB of free disk space, in which case you could establish that as a system requirement.

1 In ConsoleOne, right-click the Application object > click Properties.

2 Click the Availability tab > System Requirements to display the System Requirements page.



3 Verify that the list contains an OS Version requirement (Windows NT/2000/XP or Windows 95/98) for the workstation you want to distribute the application to. If the correct OS version is not listed, click Add > Operating System > enter the requirements for the operating system > click OK to add it to the list.

IMPORTANT: An OS Version requirement must be defined before an application will be distributed.

4 Define any additional requirements you want.

For information about each requirement type, see “[System Requirements Page](#)” on page 151.

- 5 Click OK to save the changes.

Associating the Application Object with Users or Workstations

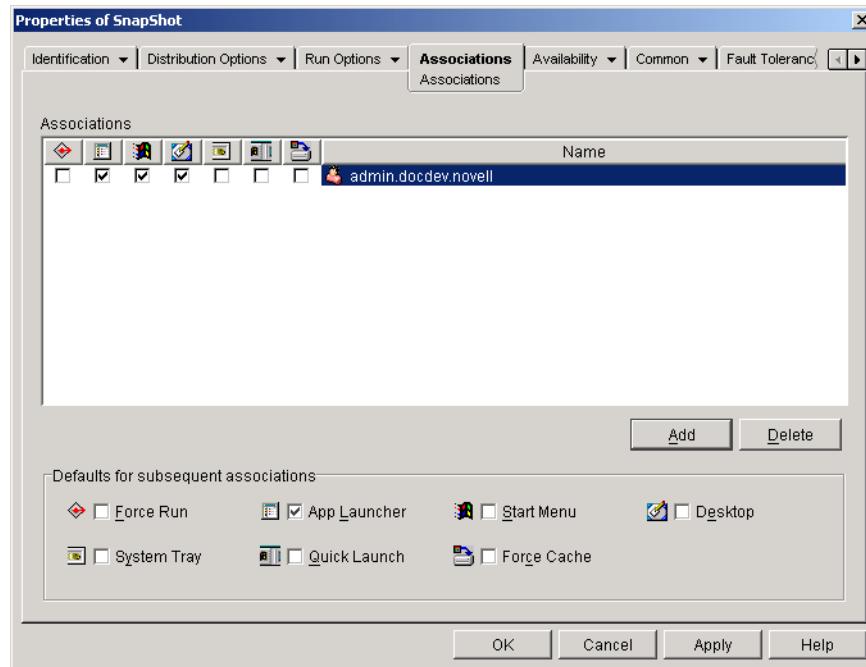
For Application Launcher/Explorer to display the application on a workstation, you must associate the Application object with the user logged in at the workstation or with the workstation itself.

- ◆ “[Associating the Application Object with Users](#)” on page 62
- ◆ “[Associating the Application Object with Workstations](#)” on page 63

Associating the Application Object with Users

You associate the application with a user by adding the User object, a Group object to which the user belongs, or a container object (Organizational Unit, Organization, or Country) in which the user resides to the Application object’s association list.

- 1 In ConsoleOne, right-click the Application object > click Properties.
- 2 Click the Associations tab to display the Associations page.



- 3 Click Add > browse and select a User object, Group object, or container object > click OK to add the user, group, or container to the Associations list.
- 4 Select the options you want applied to the Application object. By default, the App Launcher option is already checked, which causes the Application object to be displayed in the Application Launcher/Explorer window.

Force Run: Automatically runs the application. With a user-associated application, the application runs immediately after Application Launcher/Explorer starts.

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 will be 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 associating an Application object with a workstation, Force Run will not work if the Application object uses a Microsoft Windows Installer (.MSI) package. This is true only when the Application object is associated with a workstation. Force Run works with .MSI Application objects associated with users.

App Launcher: Adds the Application object's icon to the Application Launcher/Explorer window.

Start Menu: If the workstation is running Application Explorer, this option adds the Application object to the Windows Start menu. The Application object will be 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 toolbar.

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 NDS. The files are compressed to save space on the workstation's local drive. For more information about caching applications, see [Chapter 8, “Running in Disconnected Mode,” on page 73](#).

IMPORTANT: This option is required only if you want to ensure that the user can install or repair the application while disconnected from NDS. Without this option selected, the user will still be able to launch the application in disconnected mode, provided the application has already been distributed (installed) to the workstation.

- 5 Click OK to save the Application object information.

Associating the Application Object with Workstations

To associate an Application object with a workstation and have it successfully distributed and run on the workstation, you need to complete the following tasks:

- ◆ [“Importing Workstations into NDS” on page 63](#)
- ◆ [“Associating the Application Object with Workstations” on page 64](#)
- ◆ [“Assigning Users as Trustees of the Application Object” on page 65](#)
- ◆ [“Giving Workstations File System Rights to the Application” on page 65](#)

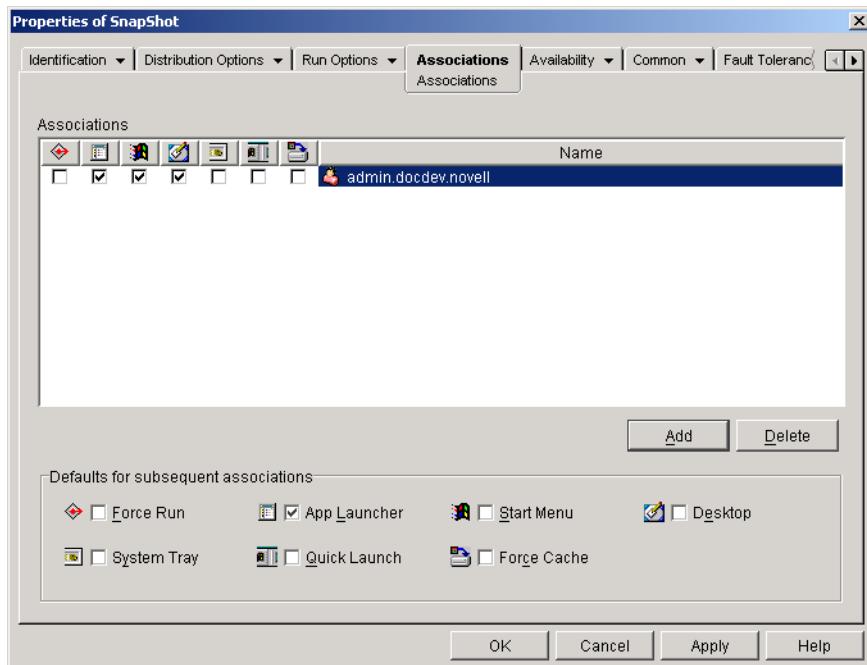
Importing Workstations into NDS

Each workstation that will be associated with applications must first be imported into NDS as a Workstation object. For details about importing workstations, see [Automatic Workstation Import and Removal in ZENworks for Desktops 3.2 Deployment Guide](#).

Associating the Application Object with Workstations

You associate the application with a workstation by adding the Workstation object, a Workstation Group object to which the workstation belongs, or a container object (Organizational Unit, Organization, or Country) in which the workstation resides to the Application object's associations list.

- 1 In ConsoleOne, right-click the Application object > click Properties.
- 2 Click the Associations tab to display the Associations page.



- 3 Click Add > browse and select a Workstation object, Workstation Group object, or container object > click OK to add the workstation, group, or container to the Associations list.
- 4 Select the options you want applied to the Application object. By default, the App Launcher option is already checked, which causes the Application object to be displayed in the Application Launcher/Explorer window.

Force Run: Automatically runs the application. With a workstation-associated application, the application runs whenever 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 will be 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 associating an Application object with a workstation, Force Run will not work if the Application object uses a Microsoft Windows Installer (.MSI) package. This is true only when the

Application object is associated with a workstation. Force Run works with .MSI Application objects associated with users.

App Launcher: Adds the Application object's icon to the Application Launcher/Explorer window.

Start Menu: If the workstation is running Application Explorer, this option adds the Application object to the Windows Start menu. The Application object will be 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 toolbar.

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 NDS. The files are compressed to save space on the workstation's local drive. For more information about caching applications, see [Chapter 8, “Running in Disconnected Mode,” on page 73](#).

IMPORTANT: This option is required only if you want to ensure that the user can install or repair the application while disconnected from NDS. Without this option selected, the user will still be able to launch the application in disconnected mode, provided the application has already been distributed (installed) to the workstation.

- 5 Click Apply to save the changes > continue with the next section, [“Assigning Users as Trustees of the Application Object” on page 65](#)

Assigning Users as Trustees of the Application Object

Associating the Application object with the workstation enables Application Launcher/Explorer to display the object on the workstation. In addition, any users who will run the application on the workstation must be trustees of the Application object. Trustee rights are not granted automatically; you must manually give default trustee rights to each user.

- 1 On the Application objects properties page > click the NDS Rights tab > Trustees of This Object.
- 2 Click Add Trustee > select the user you want to add as a trustee > click OK to add the user to the trustee list. Repeat this step for all users you need to add.
- 3 Click Apply to save the changes > continue with the next section, [“Giving Workstations File System Rights to the Application” on page 65](#)

Giving Workstations File System Rights to the Application

With workstation-associated applications that are run on Windows NT/2000/XP, the Application Launcher/Explorer Service for Windows NT/2000/XP launches and runs the applications. You need to ensure that the Application Launcher/Explorer Service has the appropriate file system rights to run the application.

If the application's executable resides on the workstation's local drive, the Application Launcher/Explorer Service automatically has the required file system rights. You don't need to do anything additional.

If the application's executable resides in a network location, you need to assign the appropriate file system rights for that location to the Workstation object. The Application Launcher/Explorer Service receives its network file system rights through the Workstation object.

Other Things You Can Do To Control the Distribution and Launching of Applications

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

- ◆ Automatically cache the application's installation files to the local workstation. This enables users to install or fix problems with the application when they are disconnected from NDS and your network. See [Chapter 8, “Running in Disconnected Mode,” on page 73](#).
- ◆ Add the Application object to a folder in the Application Launcher/Explorer window or on the Windows Start menu. See [Chapter 10, “Organizing Applications into Folders,” on page 89](#).
- ◆ Report on the success or failure of events such as distributing, caching, and uninstalling the application on workstations. See [Chapter 11, “Reporting on Application Management Events,” on page 95](#).
- ◆ Track the number of licenses being used for the application. See [Chapter 12, “Metering Software Licenses,” on page 109](#).
- ◆ Determine the schedule of when the application will be available to users. See [“Schedule Page” on page 162](#).
- ◆ Set up backup copies of the application source files to enable fault tolerance. See [“Fault Tolerance Page” on page 183](#).
- ◆ Define multiple Application objects for the same application and enable load balancing. See [“Load Balancing Page” on page 184](#).
- ◆ Set up site lists to enable users who travel from site to site to continue to have access to their applications. See [“Site List Page” on page 186](#).
- ◆ Define many additional Application object properties to control the behavior of the application. For information about all Application object properties, see [Chapter 14, “Application Object Settings,” on page 115](#).

7

Uninstalling Applications

Any application (including MSI-based applications) distributed through ZENworks® for Desktops (ZfD) Application 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 cache 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 8, "Running in Disconnected Mode," on page 73](#).

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 with users or workstations, you can configure an application to be uninstalled after a certain period of unuse, or your users can use Novell® Application Launcher™/Explorer to uninstall an application (provided you've enabled this functionality). See the following sections for information:

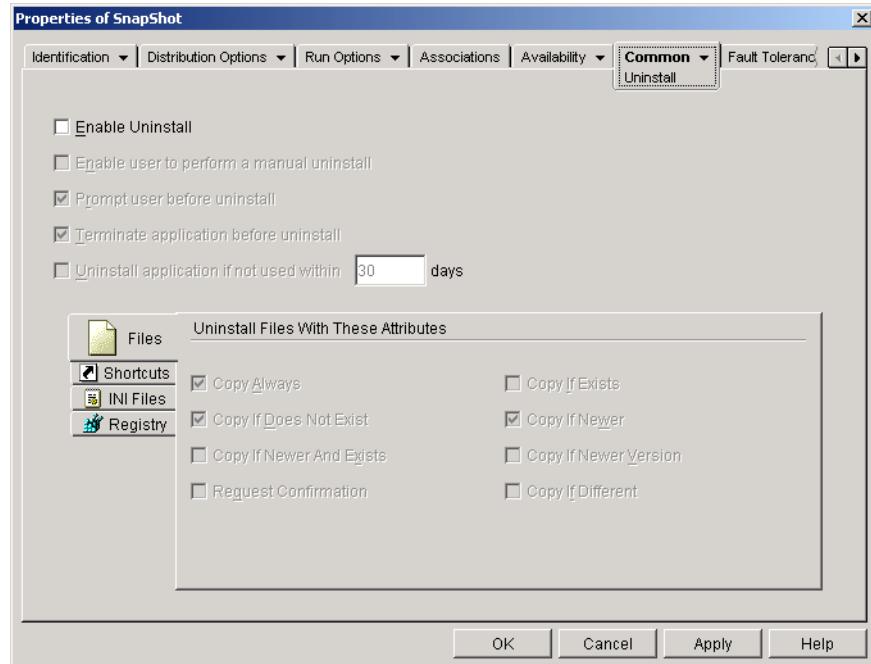
- ◆ “[Enabling an Application to be Uninstalled” on page 67](#)
- ◆ “[Uninstalling Applications by Disassociating Users or Workstations” on page 69](#)
- ◆ “[Uninstalling Unused Applications” on page 70](#)
- ◆ “[Using Application Launcher/Explorer to Uninstall an Application” on page 71](#)

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 > 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 the Enable User to Perform a Manual Uninstall to turn on the option.

For information about how users can uninstall the application, see “[Using Application Launcher/Explorer to Uninstall an Application](#)” on page 71.

- 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 will not be removed.

Terminate Application Before Uninstall: Select this option to ensure that Application Launcher/Explorer 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 “[Uninstalling Unused Applications](#)” on page 70.

- 6 Select the attributes you want used when determining whether or not to uninstall files, shortcuts, INI settings, and registry settings.

Files: Click Files > check 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). By default, the options are configured (Copy Always, Copy if Does Not Exist, Copy if Newer) to 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.

Shortcuts: Click Shortcuts > check 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 > check 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). By default, the options are configured (Create Always, Create if Exists, Create or Add to Existing Section) to 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 > check 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). By default, the options are configured (Create Always, Create if Exists) to 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.

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/Explorer during distribution of the application to the workstation. The default settings cause Application Launcher/Explorer 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 may 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.

NOTE: 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.

Uninstalling Applications by Disassociating Users or Workstations

You can force an application to be uninstalled by disassociating it from the user or workstation, provided you have enabled the application to be uninstalled (see “[Enabling an Application to be Uninstalled](#)” on page 67). If you disassociate an application that has not been configured as un/installable, the application’s files, shortcuts, INI settings, and registry settings will not be removed from the workstation.

- 1 In ConsoleOne, right-click the Application object > click Properties.
- 2 Click the Associations tab.
- 3 Select the user or workstation you want to disassociate from the application > click Delete.
- 4 Click OK.

The next time Application Launcher/Explorer refreshes, it will uninstall the application from the user’s workstation. In addition, the Application object will no longer be displayed on the user’s workstation.

HINT: You can use the Unassociated Days to Uninstall option to specify a longer grace period for the user. To do so, right-click the user's object in ConsoleOne > click Properties > click the Application Launcher tab > click Edit > modify the Unassociated Days to Uninstall option. For additional information, see “[Configuring Application Launcher/Explorer](#)” on page 36.

Uninstalling Unused Applications

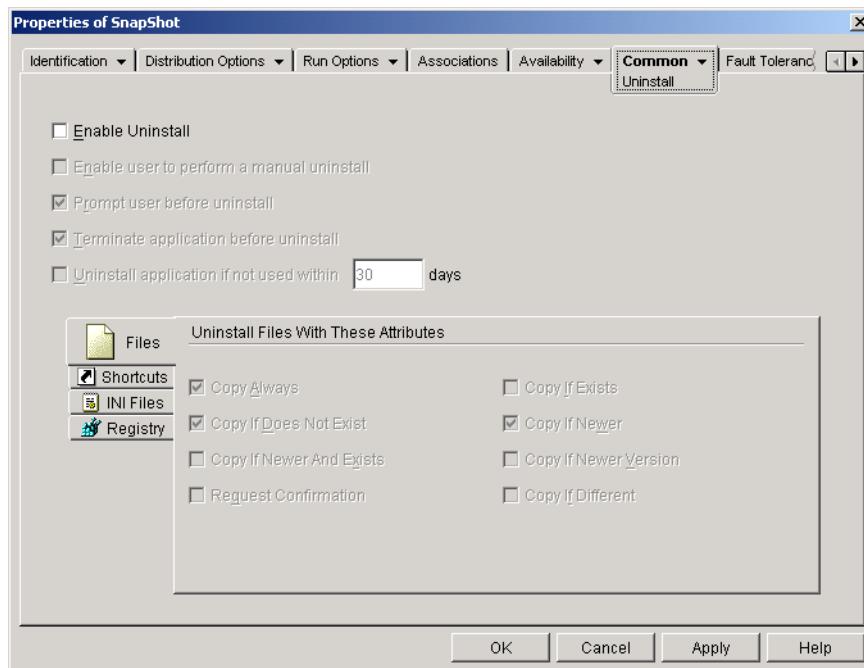
Users may install some applications, run them a few times, and then not use them again for a long period of time. These applications can unnecessarily take up valuable workstation disk space.

To free up disk space, you can have Application Launcher/Explorer uninstall an application that has not been run for a period of time. You can determine the time period; the default is 30 days. For example, you may want one application removed after 15 days of unuse, but you may not want to remove another application until it has not been run for 90 days.

Unless the user or workstation is disassociated from the Application object, Application Launcher/Explorer will continue to display the Application object icon on the workstation after the application is uninstalled. If the user double-clicks the icon, the application will be installed again.

To configure an unused application to be removed after a period of time:

- 1 In ConsoleOne, right-click the Application object > click Properties.
- 2 Click the Common tab > Uninstall to display the Uninstall page.



- 3 If the Enable Uninstall option is not turned on, click the box to turn it on.
- 4 Click the Uninstall Application If Not Used Within *XX* Days box to turn on the option > 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 “[Enabling an Application to be Uninstalled](#)” on page 67.
- 6 Click OK to save the changes.

Using Application Launcher/Explorer to Uninstall an Application

By default, users are not allowed to uninstall applications that you've distributed to them. However, if you've enabled users to uninstall applications (see "[Enabling an Application to be Uninstalled](#)" on page 67), they can do so through Application Launcher/Explorer.

- 1 On the user's workstation, right-click the Application object > click Uninstall.

When Application Launcher/Explorer 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 will remain on the workstation. This enables the user to reinstall the application at a later time.

8

Running in Disconnected Mode

Novell® Application Launcher™/Explorer enables users to install, run, verify (fix), and uninstall applications while disconnected from NDS® and the network. This enables users to run the same applications when connected to or disconnected from NDS.

The following sections provide information and instructions to help you understand the Application Launcher/Explorer disconnected mode and to support disconnected users:

- ◆ “Disconnectable Applications” on page 73
- ◆ “Workstation Cache” on page 74
- ◆ “Configuring Application Launcher/Explorer to Start Automatically” on page 76
- ◆ “Configuring Applications as Disconnectable” on page 77
- ◆ “Forcing an Application to be Cached” on page 78
- ◆ “Changing the Default Cache Location” on page 78
- ◆ “Distributing Applications to Disconnected Workstations” on page 80
- ◆ “Enabling Users to Manage Applications” on page 81

Disconnectable Applications

When you create an Application object, the application is automatically configured as being disconnectable. After a disconnectable application is distributed (installed) to a workstation, Application Launcher/Explorer will continue to display the Application object icon even after the workstation has disconnected from NDS and will attempt to launch the application when the user double-clicks the Application object icon.

Only applications that can run locally on a disconnected workstation should be configured as disconnectable. Applications that rely on network resources should not be configured as disconnectable. This includes:

- ◆ Applications that access a network database
- ◆ Client/server applications
- ◆ Applications that depend on network drive mappings or print captures
- ◆ Applications that use NDS macros (other than %CN%) in any of the Application object property fields
- ◆ Applications that require a persistent connection to the network

Workstation Cache

To support disconnected mode, Application Launcher/Explorer creates a cache directory on the workstation's Windows* drive (the drive containing the SYSTEM32 directory). This directory, called NALCACHE, is created as a hidden directory at the root of the drive. If necessary, the location can be changed (see “[Configuring Application Launcher/Explorer](#)” on page 36).

The NALCACHE directory contains subdirectories for each application associated with the user. Each application's folder contains the following two types of cache:

- ◆ “[Launch Cache](#)” on page 74
- ◆ “[Install Cache](#)” on page 75

Launch Cache

Application Launcher/Explorer automatically creates an application's launch cache when the application is distributed to a user or workstation. The launch cache contains all the NDS information Application Launcher/Explorer needs to launch and run an application while disconnected. This cache also contains the information needed to uninstall the application.

The launch cache files are stored in the NALCACHE\NDS_TREE\APP_OBJECT directory, where *APP_OBJECT* is the NDS context of the Application object and *NDS_TREE* is the tree where the Application object resides. By default, all files except those containing application files, INI settings, shortcut icons, and registry settings are refreshed each time Application Launcher/Explorer performs a timed or manual refresh. These files remain the same until the Version Number (Application object > Distribution Options tab > Options page) is changed so that the application can be successfully uninstalled.

The following table explains the files that may reside in the launch cache. An asterisk next to a file name indicates that Application Launcher/Explorer does not update that file when it refreshes.

File	Description
DSATTR.BIN	Contains all of the Application object's NDS attributes.
FOLDERS.BIN	Contains the list of folders where the Application object will be displayed in the Application Launcher/Explorer window and the Start menu.
COMPLETE.BIN	Contains the Application object's complete NDS name. If this file does not exist, Application Launcher/Explorer assumes that the cache is bad and recreates the cache.
STRM1.BIN	Contains the Application object's icon.
STRM2.BIN	Contains all text file modifications.
STRM3.BIN*	Contains all files to be copied to the workstation.
STRM4.BIN*	Contains registry modifications that will be made during initial distribution.
STRM5.BIN*	Contains INI modifications that will be made during initial distribution.
STRM6.BIN*	Contains shortcut modifications that will be made during initial distribution.

File	Description
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*	Contains registry modifications to be made each time the application is launched.
STRM14.BIN*	Contains INI modifications to be made each time the application is launched.
STRM15.BIN*	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*	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.
STRM21.BIN	Contains the pre-install schedule information.

Install Cache

Application Launcher/Explorer creates an application's install cache only if you select the Force Cache option when associating the Application object with users or workstations, or if individual users use Application Launcher/Explorer's Application Management feature to cache the application. The install cache contains the application source files needed to install or verify an application while disconnected. Although the files are compressed, they can still require a large amount of disk space on the workstation local drive, so an application should only be cached if users will need to install the application or verify (fix) the already installed application while disconnected.

IMPORTANT: An application can still be 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 while in disconnected mode, or to be verified (fixed) if a problem occurs.

Configuring Application Launcher/Explorer to Start Automatically

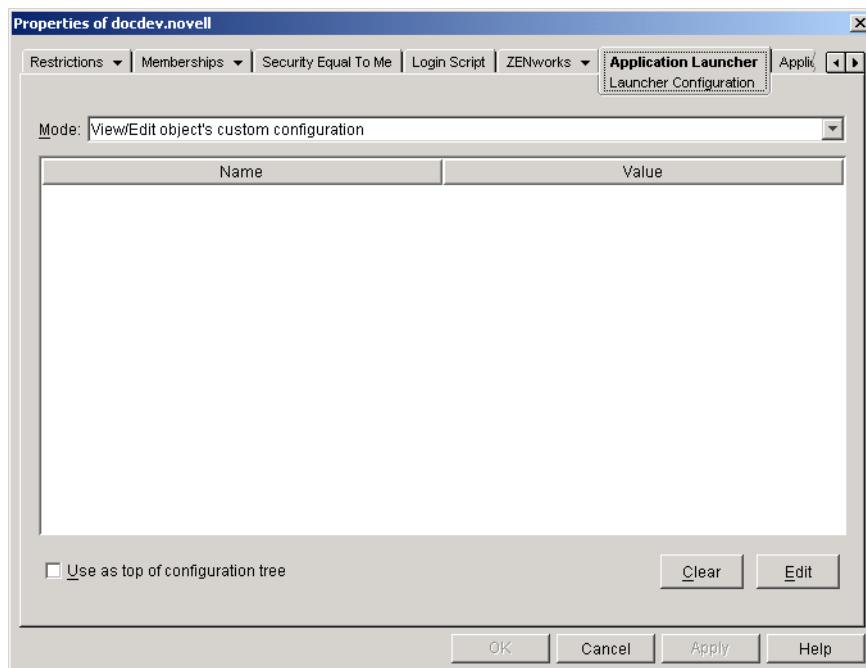
Typically, when connected to the network, Application Launcher/Explorer is started through a command in the user's login script. To automatically start Application Launcher/Explorer in disconnected mode, you can add Application Launcher or Application Explorer to the Windows Startup folder by using the Launcher Configuration settings in NDS.

Depending on which program is running at the time, either an Application Launcher (NALWIN32.EXE) shortcut or an Application Explorer (NALDESK.EXE) shortcut is added to the Startup folder.

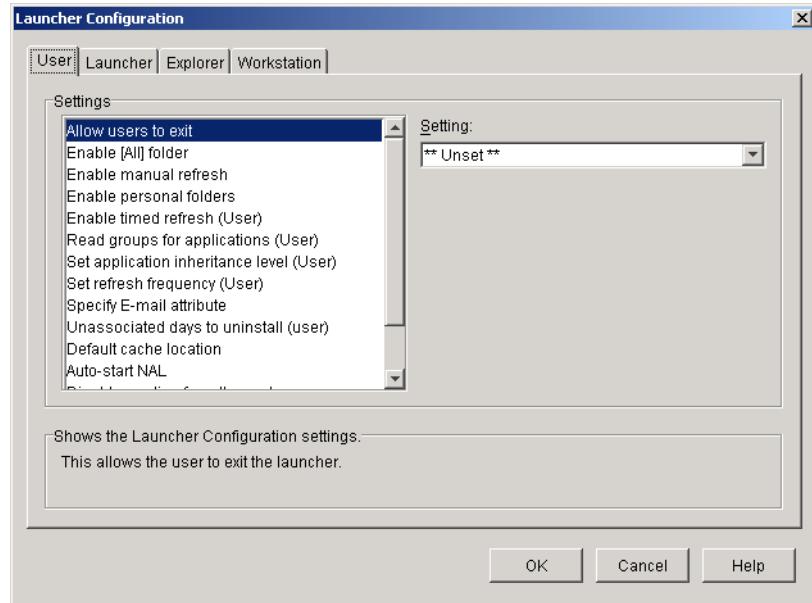
If Application Launcher or Application Explorer attempts to start twice, once through the user's login script and once through the Startup folder, only one instance of the program will be launched.

To add Application Launcher or Application Explorer to the Startup folder:

- 1 In ConsoleOne®, right-click the User, Group, or container object for the users whose Application Launcher/Explorer you want to configure > click Properties.
- 2 Click the Application Launcher tab to display the Launcher Configuration page.



- 3 Click Edit to display the Launcher Configuration dialog box.

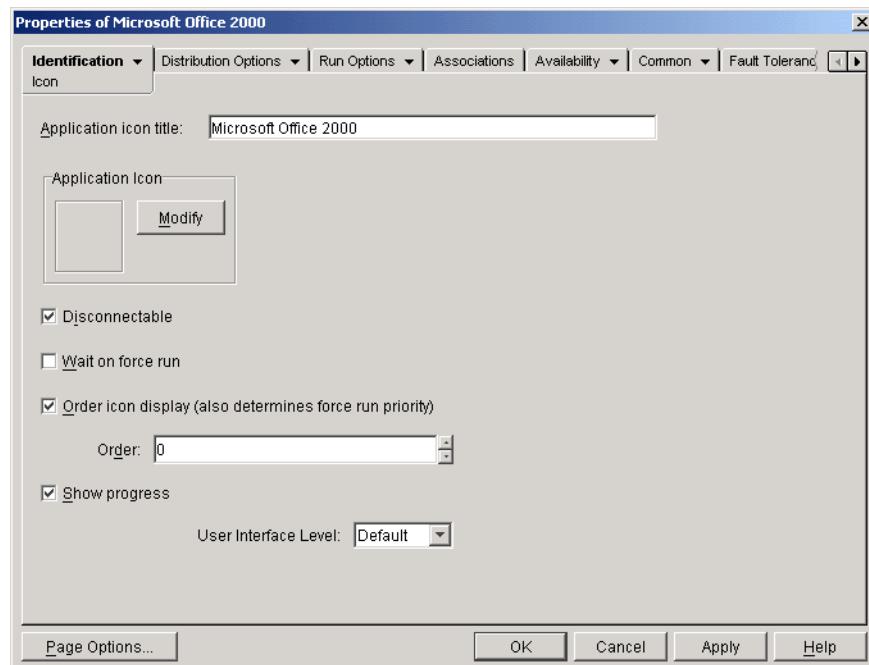


- 4** In the Settings list (User tab), select Auto-Start NAL > set the option to Yes.
- 5** Click OK > OK to save the changes.

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 > click Properties to display the Icon page.



- 2** Select the Disconnectable option to make the application disconnectable.

or

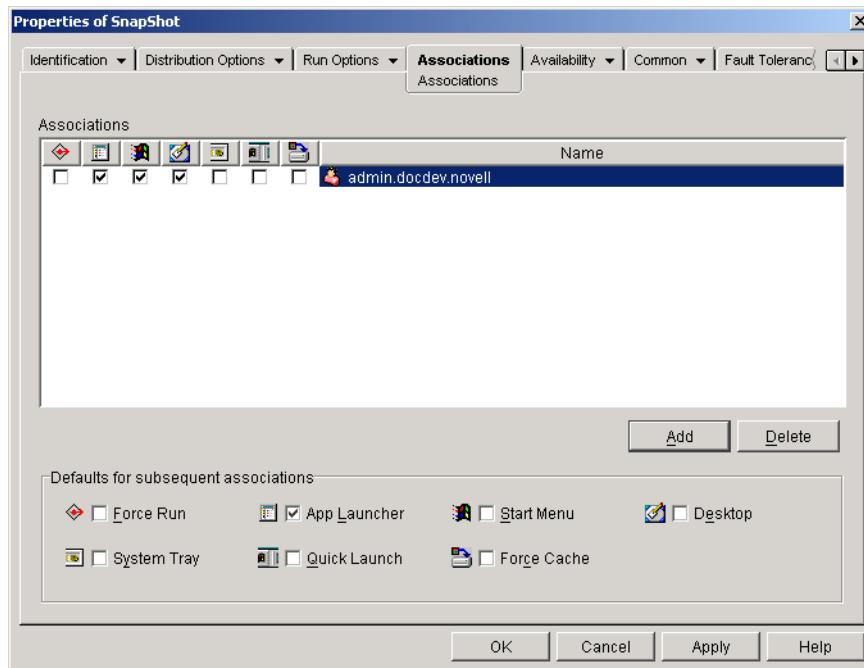
Deselect the option to make the application not disconnectable.

- 3 Click OK to save the changes.

Forcing an Application to be Cached

If you want to ensure that a user has an application cached to his or her local drive so that he or she can install or verify the application while disconnected, you can force cache the application. For information about enabling users to cache applications themselves, see “[Enabling Users to Manage Applications](#)” on page 81.

- 1 In ConsoleOne, right-click the Application object > 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.
- 4 Click OK to save the changes.

HINT: The same task can be accomplished through the Applications tab on a User, Workstation, Group, Workstation Group, or container object.

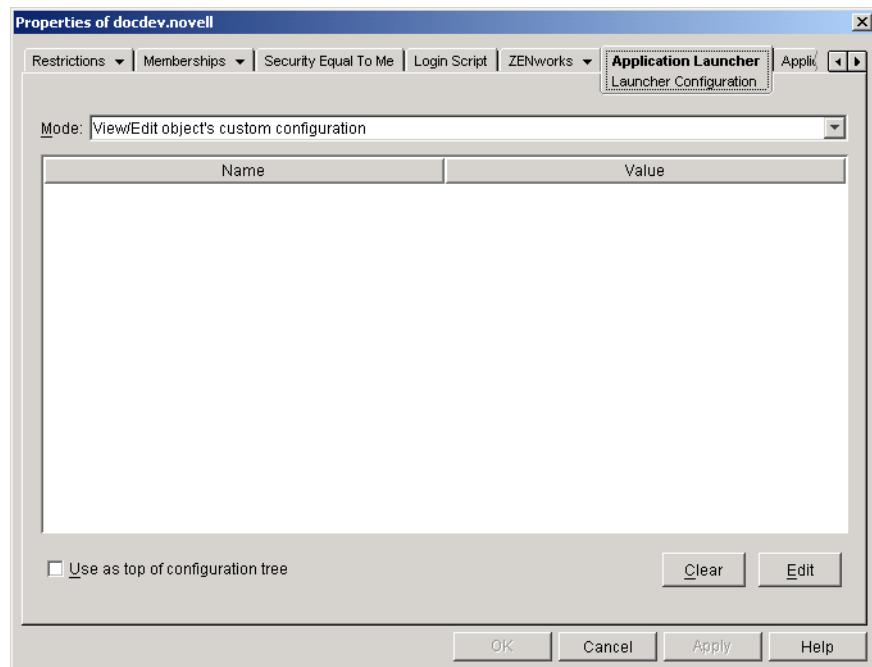
Changing the Default Cache Location

As mentioned in “[Workstation Cache](#)” on page 74, the default cache (NALCACHE) is created as a hidden directory on the root of the workstation’s Windows drive.

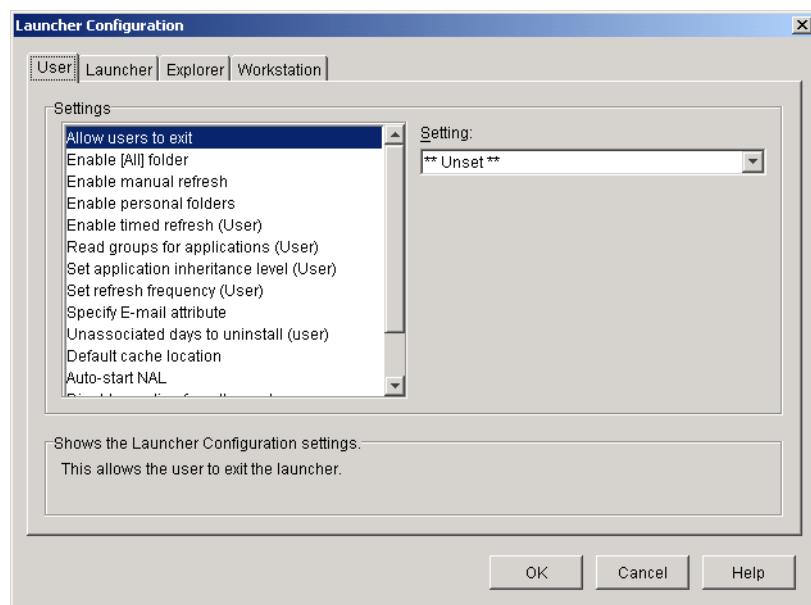
You can change the location by specifying a different drive letter. The next time Application Launcher/Explorer refreshes (either a timed refresh or a manual refresh), the NALCACHE directory will be recreated at the root of the location that the workstation’s drive letter is mapped to. The old NALCACHE directory is not removed automatically.

To change the location of the cache directory:

- 1 In ConsoleOne, right-click the User, Group, or container object for the users whose cache location you want to change > click Properties.
- 2 Click the Application Launcher tab to display the Launcher Configuration page.



- 3 Click Edit to display the Launcher Configuration dialog box.



- 4 In the Settings list (User tab), select Default Cache Location > select the drive letter you want to use.
- 5 Click OK > OK to save the changes.

Distributing Applications to Disconnected Workstations

As long as a disconnected workstation has Application Launcher/Explorer 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 then send it to the user. The user inserts the CD into his or her workstation's drive, Application Launcher/Explorer reads the CD and displays the Application object's icon in the places you've configured (Application Launcher/Explorer 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.

NOTE: The removable media includes an AUTORUN.INF file that causes Application Launcher/Explorer to refresh and display the Application object. If the user's workstation is not configured to autorun CDs, or you use another media type, the user may have to manually refresh Application Launcher/Explorer. This is done by right-clicking the Application Launcher/Explorer icon and clicking Refresh.

To create a cache on a removable media:

- 1 In ConsoleOne, click the Tools menu > click Application Launcher Tools > click Create Virtual CD to start the Create CD Wizard.
- 2 Click Add > browse for and select the Application object for the application you want to distribute. Repeat this step to add additional applications.
- 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. For information about each of these options, see "[Associations Page](#)" on [page 148](#).
- 4 After you've finished adding applications, click Next.

- 5 In the Select Directory for Virtual CD, select the location where you want to create the cache for the applications.

IMPORTANT: Make sure the location you select has sufficient disk space for the application cache. If enough disk space does not exist, you'll receive a message stating that an error has occurred.

- 6 If you want to restrict the amount of time the user can continue to use the application (or applications) after the media has been removed from the workstation, select the When CD Is Removed Disassociation Will Occur After option > select the number of days after which the disassociation will occur.

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.

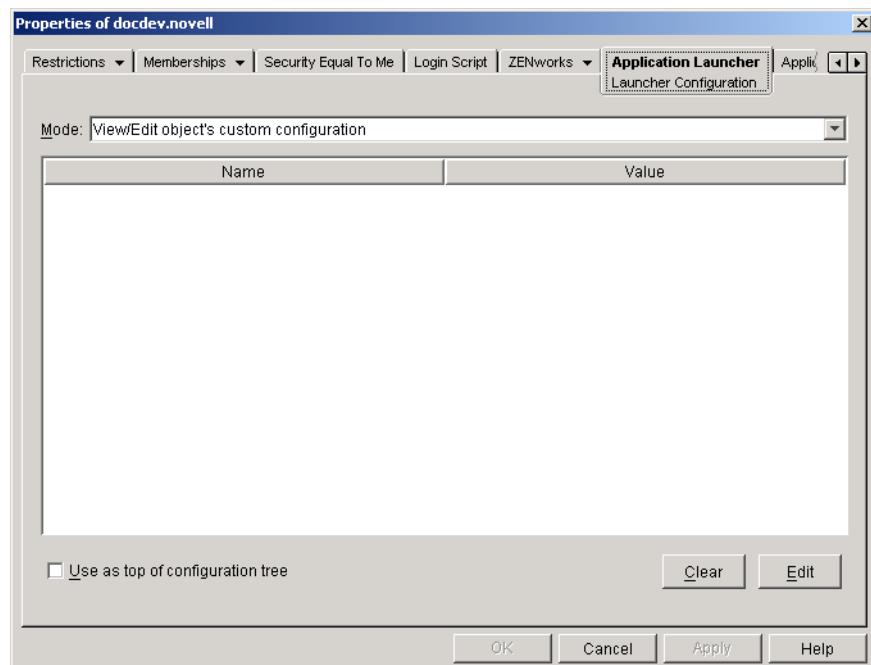
- 7 Click Next > review the settings > click Finish to create the cache.
- 8 If necessary, copy the cache to the removable media (for example, burn the CD).

Enabling Users to Manage Applications

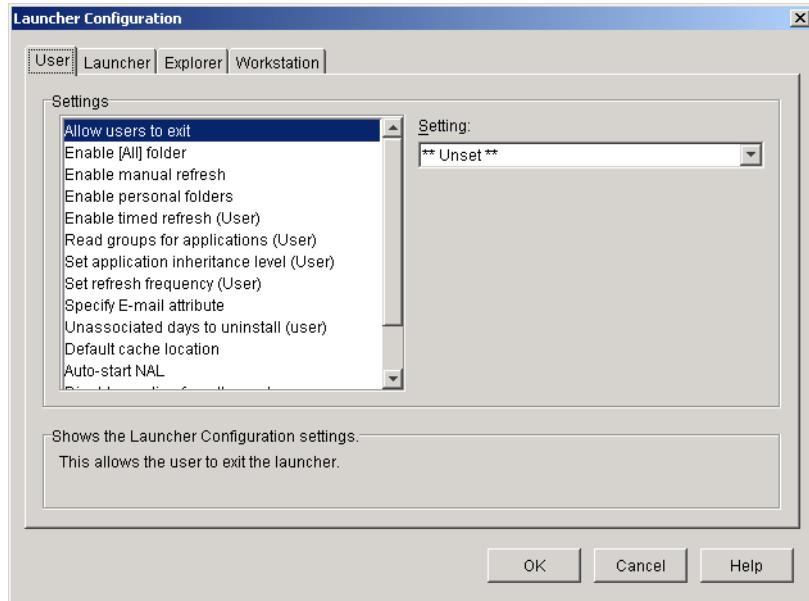
Application Launcher/Explorer includes a Manage Applications feature that enables users to cache applications, remove applications from the cache, install applications from the cache, and uninstall applications. By default, this feature is disabled to maintain administrative control over these processes. However, you may find it necessary or desirable to enable this feature for some users:

To enable users to manage their applications:

- 1** In ConsoleOne, right-click the User, Group, or container object for the users whom you want to enable the Manage Applications feature > click Properties.
- 2** Click the Application Launcher tab to display the Launcher Configuration page.



- 3** Click Edit to display the Launcher Configuration dialog box.



- 4 In the Settings list (User tab), select Enable Manage Applications Dialog > set the option to Yes.
- 5 Click OK > OK to save the changes.
- 6 To use the feature, at a user's workstation, right-click the Application Launcher/Explorer icon > click Manage Applications.

For information about using the Manage Applications features, click Help in the Application Launcher/Explorer window.

9

Configuring Application Fault Tolerance and Load Balancing

Novell® ZENworks® for Desktops (ZfD) Application Management provides a variety of methods to help you make sure that users always have access to their critical applications. These methods are discussed in the following sections:

- ◆ “Setting Up Fault Tolerance” on page 83
- ◆ “Setting Up Load Balancing” on page 85
- ◆ “Setting Up Site Lists” on page 87

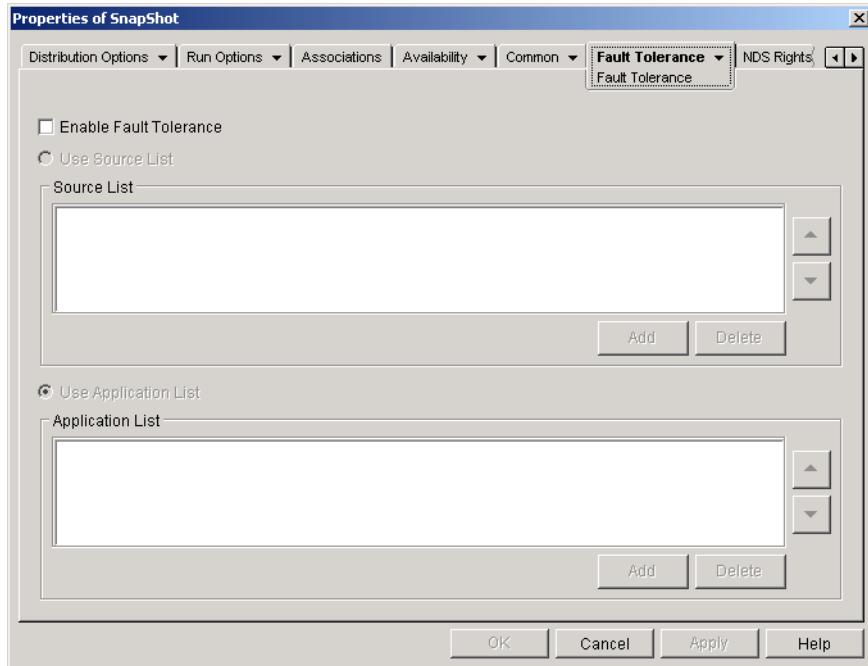
Setting Up Fault Tolerance

Network problems can make Application objects or their file packages unavailable. You can establish fault tolerance for an application by specifying backup Application objects and file packages. When an Application object or its file package is unavailable, Novell Application Launcher™/Explorer will use the backup Application objects or file packages.

We recommend you use fault tolerance, rather than load balancing (see “[Setting Up Load Balancing](#)” on page 85) when servers are located at various sites throughout a large wide area network. However, the servers must reside in the same Novell eDirectory™ tree. You can implement both methods if necessary. Application Launcher/Explorer will try load balancing first, then fault tolerance.

To set up fault tolerance for an application:

- 1 In ConsoleOne®, right-click the Application object for which you want to establish fault tolerance > click Properties to display the Application object’s property pages.
- 2 Click the Fault Tolerance tab > Fault Tolerance to display the Fault Tolerance page.



The Fault Tolerance page specifies the installation package sources or Application objects that Application Launcher/Explorer will use as backups if the Application object becomes unavailable. Application Launcher/Explorer tries the package sources or Application objects in the order shown in the Source List or Application List, from top to bottom.

IMPORTANT: The Fault Tolerance page does not apply to Application objects that use Microsoft Windows Installer (.MSI) packages. For a .MSI Application object, you can provide source resiliency (fault tolerance) through the use of additional source locations for the installation package (Common > Sources page).

3 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/Explorer use a list of Package sources as the backup. You must have already created at least one installation package source (Common tab > Sources page).

This option is not available on Application objects for terminal server applications or Web applications.

To add a package source, click Add > browse to and select the package source > click OK to add it to the Source List.

To delete a package source, select the package source from the Source List > click Delete to remove it.

Application Launcher/Explorer tries the sources in the order they are listed, from top to bottom. To change the order the package sources, select a package source from the Source List > 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/Explorer 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 too busy or is not available, then Application Launcher/Explorer tries each Application object in the specified order.

Only one-level-deep fault tolerance is supported. If Application Launcher/Explorer fails over to one of the backup Application objects in the list and the backup Application object fails, the distribution will fail even if fault tolerance has been enabled for the backup Application object.

NOTE: If Application Launcher/Explorer has detected a remote connection and is running in remote mode, it will not use the Application List for fault tolerance. Application Launcher/Explorer will only use the Source List.

To add an Application object to the list, click Add > browse to and select the Application object > click OK to add it.

To remove an Application object from the list, select the Application object > click Delete to remove it.

Application Launcher/Explorer 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 > 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.

- 4 Click OK to save your changes.

Setting Up Load Balancing

Many users accessing the same Application object and its file package can cause delays with the application's distribution. You can use load balancing to spread access to an application among multiple Application objects and file packages.

With load balancing, you create multiple copies of the Application object and file package for the same application, storing the file packages on separate network servers. When a user launches the application, Application Launcher/Explorer can use the Application object and file package associated with that application, or it can be redirected to another copy of the Application object or file package.

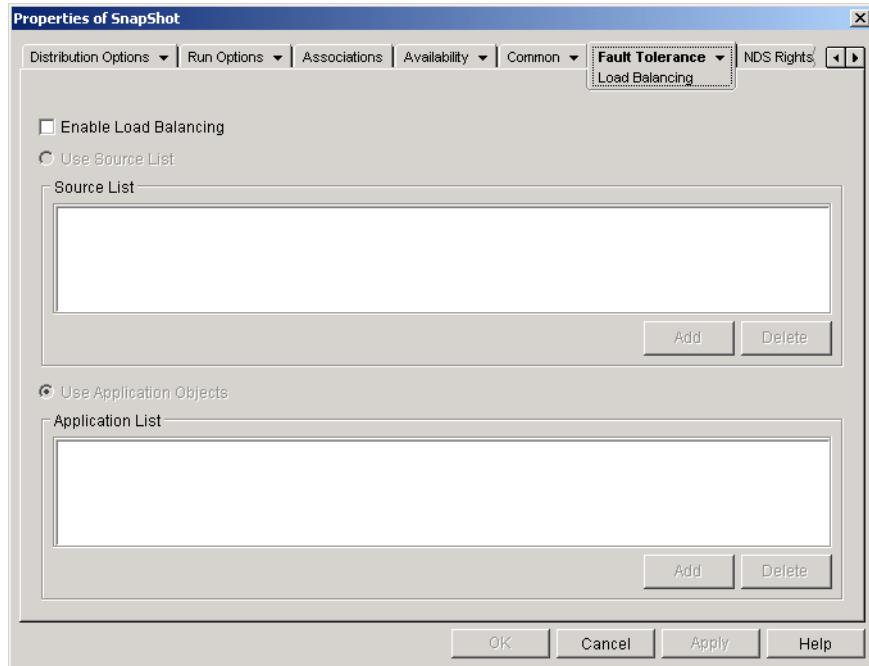
We recommend you use load balancing when all the servers for the Application objects are located at a single site. If the servers are located at various sites throughout a large wide area network, we recommend you use fault tolerance (see “[Setting Up Fault Tolerance](#)” on page 83). You can implement both methods if necessary. Application Launcher/Explorer will try load balancing first, then fault tolerance.

To set up load balancing for an application:

- 1 Make sure that you've created multiple installation package sources or Application objects for the application, with the application source files located on different servers.

When a user launches the application, Application Launcher/Explorer selects one of the package sources or Application objects to use. If the package source or Application object is unavailable, Application Launcher/Explorer selects another one.

- 2 In ConsoleOne, right-click the Application object for which you want to establish load balancing > click Properties to display the Application object's property pages.
- 3 Click Fault Tolerance > Load Balancing to display the Load Balancing page.



4 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.

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 a .MSI Application object, check 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/Explorer will randomly select source locations from the list to balance the workload.

Use Source List: Select this option to have Application Launcher/Explorer use a list of package sources for load balancing. You must have already created at least one package source (Common tab > Sources page).

This option is not available on Application objects for terminal server applications or Web applications.

To add a package source to the list, click Add > browse to and select the package source > click OK.

To remove a package source from the list, elect the package source > click Delete.

Use Application List: Select this option to have Application Launcher/Explorer 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.

To add an Application object to the list, click Add > browse to and select the Application object > click OK.

To remove an Application object from the list, select the Application object > click Delete.

Setting Up Site Lists

Site lists let you link an application at one site (Site A) to an application at another site (Site B). When a user at Site A launches the application, the Site A application is used. However, if the user travels to Site B and launches the application, the application from Site B is used. This ensures that users who travel from site to site continue to have fast access to their applications while reducing WAN traffic and associated costs.

For example, users at two different sites run a spreadsheet application from servers that are local to their sites. Different Application objects are used to distribute and launch the spreadsheet application from the two servers. If you link the two Application objects, any Site1 users who travel to Site2 will be able to run the application from the Site2 server the same way they would run it from their own Site1 server. The same would be true for Site2 users who travel to Site1.

You can link to one other Application object only. 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:

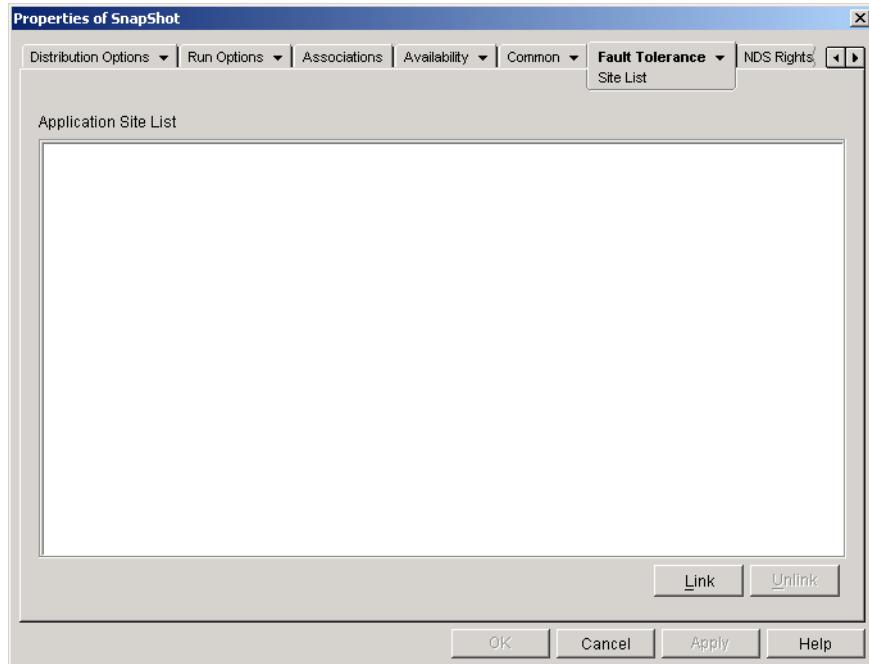
```
App1 linked to: App2  
App2 linked to: App1  
App3 linked to: (nothing)
```

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 linked to: App2, App3  
App2 linked to: App1, App3  
App3 linked to: App1, App2
```

To establish a site list for an Application object:

- 1** In ConsoleOne, right-click the Application object for which you want to establish a site list > click Properties to display the Application object's property pages.
- 2** Click Fault Tolerance > Site List to display the Site List page.



- 3** Click Link > browse for and select the Application object you want to link to > 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 will also be added to the list

- 4** Repeat the previous step to link to additional Application objects.

- 5** Click OK when you are finished.

10

Organizing Applications into Folders

ZENworks® for Desktops Application Management enables you to organize Application objects into folders. Folders show up in the Novell® Application Launcher™/Explorer window and on the Windows* Start menu. Four types of folders are supported:

- **Application Folder (Administrator Created):** An application folder is created as an NDS® 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.
- **Custom Folder (Administrator Created):** A custom folder is created specifically for a single Application object in NDS. 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\NOTE PAD\NOTE PAD.EXE).
- **Personal Folder (User Created):** A personal folder is created in Application Launcher/Explorer 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 “[Configuring Application Launcher/Explorer](#)” on page 36.
- **System Folder (System Created):** A system folder is automatically created by Application Launcher/Explorer 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 NDS 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.

The following sections provide instructions about creating an Application Folder object and adding Application objects to application folders or custom folders:

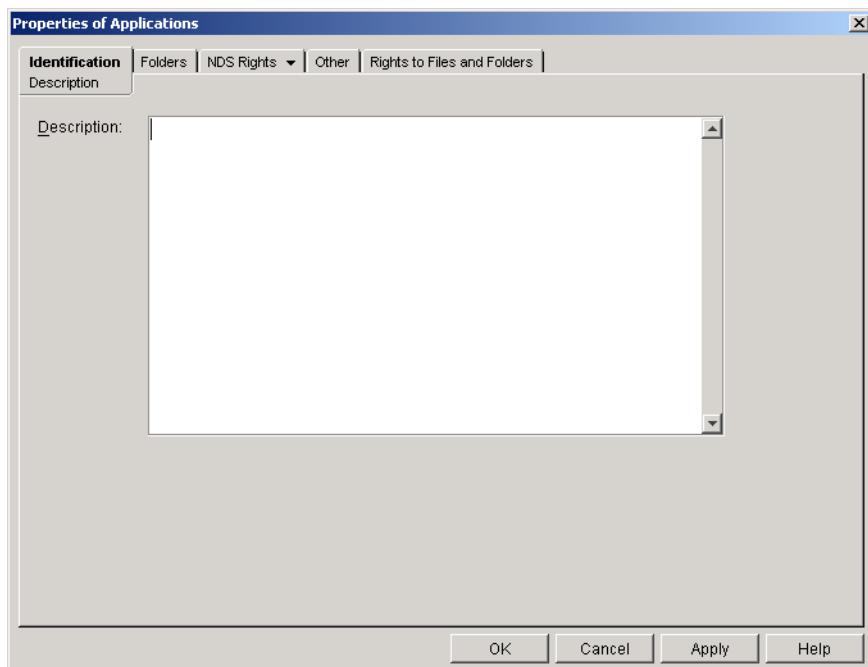
- “[Creating Application Folder Objects](#)” on page 90
- “[Adding Applications to Application Folders](#)” on page 91
- “[Adding Applications to Custom Folders](#)” on page 93

Creating Application Folder Objects

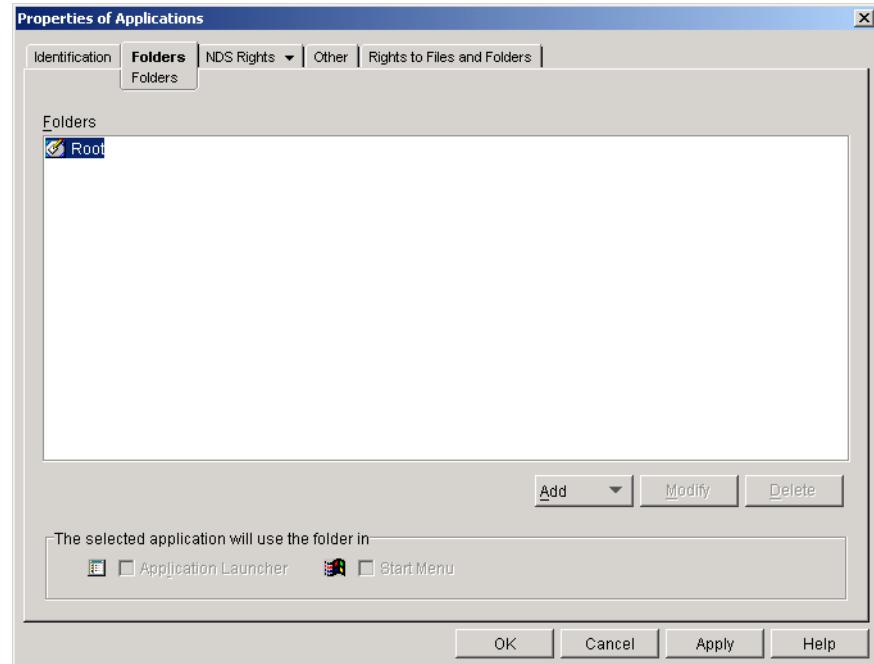
- 1 In ConsoleOne™, right-click the container where you want to create the Application Folder object > click New > click Object to display the New Object dialog box.
- 2 Click App:Folders > OK to display the New App:Folders dialog box.



- 3 In the Name field, enter a name for the object. The name is displayed only in NDS and does not need to be the same name as the one you want to appear in the Application Launcher/Explorer window and on the Start menu on the users' workstations.
- 4 Check the Define Additional Properties box > 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/Explorer 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 > press Enter.

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

Delete: To remove a folder, select the folder in the Folders tree > click Delete.

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

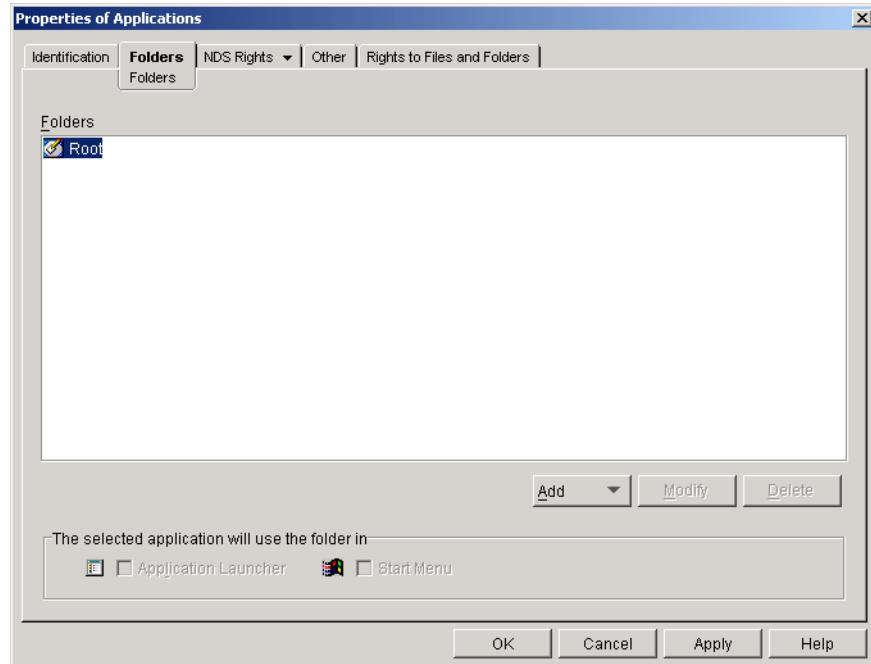
Adding Applications to Application Folders

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.

- ◆ “[Using the Application Folder Object to Add Applications](#)” on page 91
- ◆ “[Using the Application Object to Add the Application to a Folder](#)” on page 92

Using the Application Folder Object to Add Applications

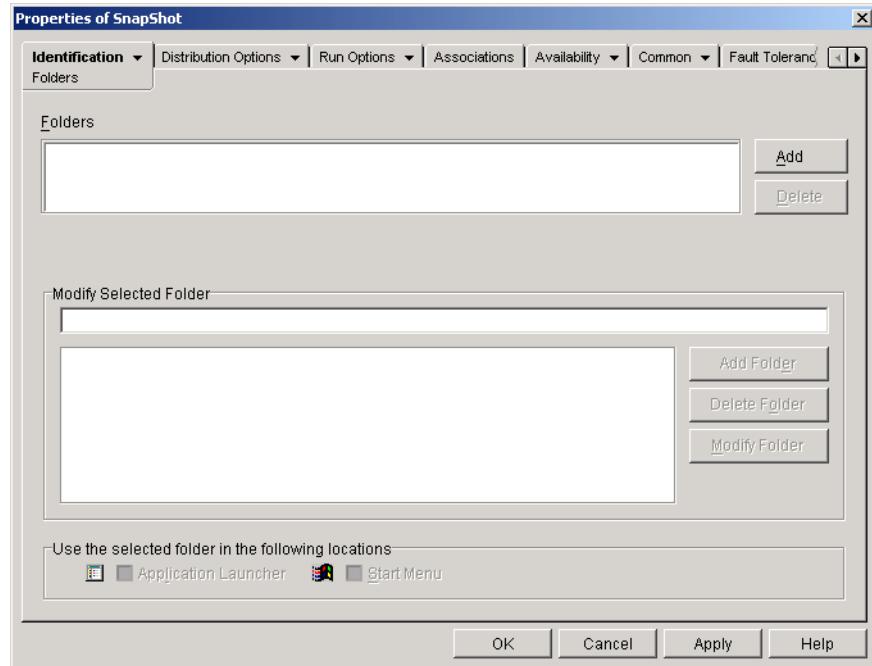
- 1** In ConsoleOne, right-click the Application Folder object you want to add the Application object to > click Properties.
- 2** Click the Folders tab to display the Folders page.



- 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 > click OK to add it to the folder in the folders list.
- 5** When finished adding Application objects > click OK to save your changes.

Using the Application Object to Add the Application to a Folder

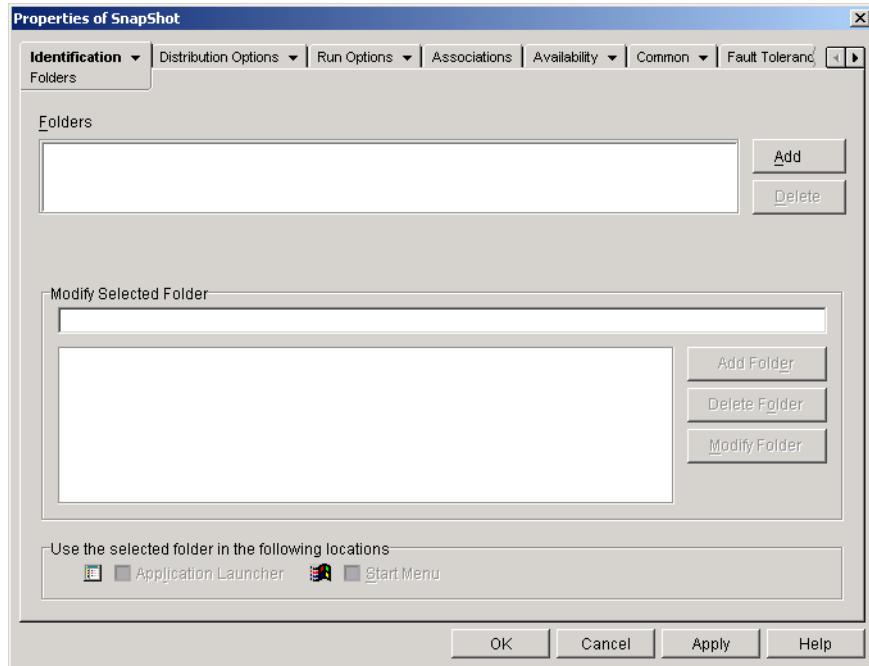
- 1** In ConsoleOne, right-click the Application object you want to add to a folder > click Properties.
- 2** Click the Identification tab > Folders to display the Folders page.



- 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 > click OK to display the Folder Object Structure dialog box.
- 5** Select the folder you want to add the Application object to > click OK to add it to the Folders list.
HINT: If you select the Root folder, the Application object will be displayed directly on the Start menu and at the root of the Application Launcher/Explorer window.
By default, the Application object will be displayed in the selected folder on the Start menu and the Application Launcher/Explorer 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/Explorer window and deselect Start Menu if you don't want the Application object included in that folder on the Windows Start menu.
NOTE: A folder only displays in the Application Launcher/Explorer window and on the Start menu 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.

Adding Applications to Custom Folders

- 1** In ConsoleOne, right-click the Application object you want to add to a folder > 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 > press Enter.
- 5 If you want to add subfolders, select the folder > click Add Folder > type a name for the subfolder > press Enter.
IMPORTANT: If you add subfolders, the Application object is displayed in the lowest level folder.
- 6 Click OK to save the changes.

11

Reporting on Application Management Events

ZENworks® for Desktops (ZfD) Application Management supports reporting on the success or failure of the following events:

- ◆ Launching of an application
- ◆ Distribution of an application
- ◆ Filtering of an application when the workstation does not meet the application's system requirements
- ◆ Removal of an application (uninstall)
- ◆ Caching of an application's installation source files

Events can be sent to a database, as SNMP traps to a management console, or recorded in a text log file. Because event reporting is established on a per application basis, you can use a database for reports from one application, SNMP traps for reports from a second application, and a text log file for reports from a third application. Or, you can use all three methods for different events for the same application (for example, launch events sent to a database and caching events sent to a log file).

The following sections provide information about setting up each type of event reporting and configuring Application objects to use one or more reporting methods:

- ◆ “Setting Up Database Reporting” on page 95
- ◆ “Setting Up SNMP Trap Reporting” on page 102
- ◆ “Setting Up Log File Reporting” on page 104
- ◆ “Configuring Applications to Use Reporting” on page 104
- ◆ “Generating Database Reports” on page 105
- ◆ “Understanding Log File Reports” on page 106

If you are distributing .MSI applications (applications installed by Microsoft® Windows® Installer), you can turn on verbose logging so that the Windows Installer will output information to a log file on the user's workstation. The following section provides information:

- ◆ “Enabling Windows Installer Verbose Logging” on page 107

Setting Up Database Reporting

Novell® Application Launcher™/Explorer can record events to most ODBC-compatible databases, provided:

- ◆ Each user's workstation has the correct database driver installed.
- ◆ The connection is configured correctly in NDS®.

ZfD includes a Sybase* database that you can install. 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.

NOTE: Although the same database installation can be used for both Application Management and Workstation Inventory, each component still uses its own database file. Application Management creates a NAL.DB database file and Workstation Inventory creates a MGMTDB.DB database file.

To set up database reporting, you need to complete the following tasks:

- ◆ “[Installing the Sybase Database](#)” on page 96
- ◆ “[Installing ODBC Drivers to Workstations](#)” on page 96
- ◆ “[Creating a ZENworks Database Object](#)” on page 96
- ◆ “[Configuring the Database’s JDBC and ODBC Information](#)” on page 97
- ◆ “[Granting ODBC Property Rights to Users](#)” on page 99
- ◆ “[Enabling the ZENworks Database Policy](#)” on page 100
- ◆ “[Configuring Application Objects to Use Database Reporting](#)” on page 101

Installing the Sybase Database

If you have not installed a database to use, you can install the Sybase database included on the ZENworks for Desktops *Program* CD. For instructions, see [Installing Application Management](#) in [Application Management](#) in *ZENworks for Desktops 3.2 Deployment Guide*.

The Sybase database installation requires approximately 16 MB on a network server. However, as with all reporting databases, the database can expand rapidly to consume large amounts of disk space.

Installing ODBC Drivers to Workstations

Application Launcher/Explorer uses an ODBC driver to write application event information to the database. You need to ensure that the correct ODBC driver (for the database you are using) is installed on each workstation for which you want to record event information.

The correct ODBC driver (DBODBC6.DLL) for the Sybase database is installed by default. However, on Windows 95/98, you may need to install two additional DLLs (ODBCP32.DLL and ODBCINT.DLL) that are not installed during a Windows minimal installation. To install these files, copy them from the NAL REPORTING directory on the *Companion* CD 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 NT/2000/XP. The files are already present.

Creating a ZENworks Database Object

If you are using the Sybase database that comes with ZfD and have already installed it, the installation program creates the database object in NDS. You can skip to the next section, “[Configuring the Database’s JDBC and ODBC Information](#)” on page 97.

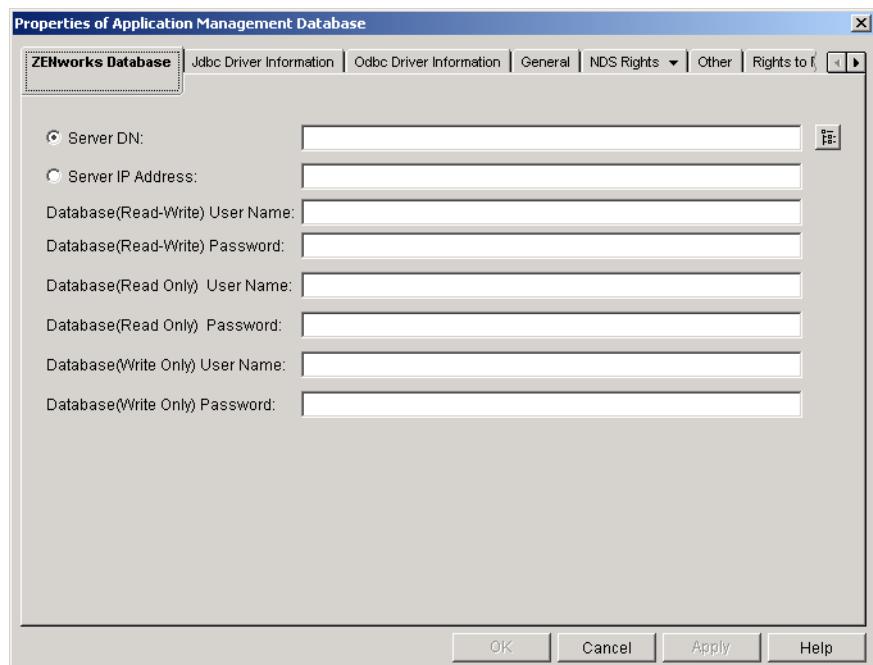
If you are using another database, such as the Oracle* database the comes with NetWare® 5.1, you will need to create a ZENworks Database object in NDS to represent the database.

To create a ZENworks Database object:

- 1 Right-click the container in which you want to create the object > click New > click Object to display the New Object dialog box.
- 2 Select ZENworks Database > click OK to display the New ZENworks Database dialog box.



- 3 In the Name box, type a name for the database, such as ZfDAppManReports.
- 4 Select the Define Additional Properties box > click OK.



- 5 In the Server DN field, browse for and select the Server object for the server where the database is physically installed and running.
- IMPORTANT:** If the ZENworks database is located on a NetWare 4.x server, you need to enter the server's IP address in the Server IP Address field rather than adding the server's object to the Server DN field.
- 6 If you want to set up Read-Write, Read-Only, or Write-Only users, enter information in the appropriate user name and password fields.
- 7 Click OK to save the information.

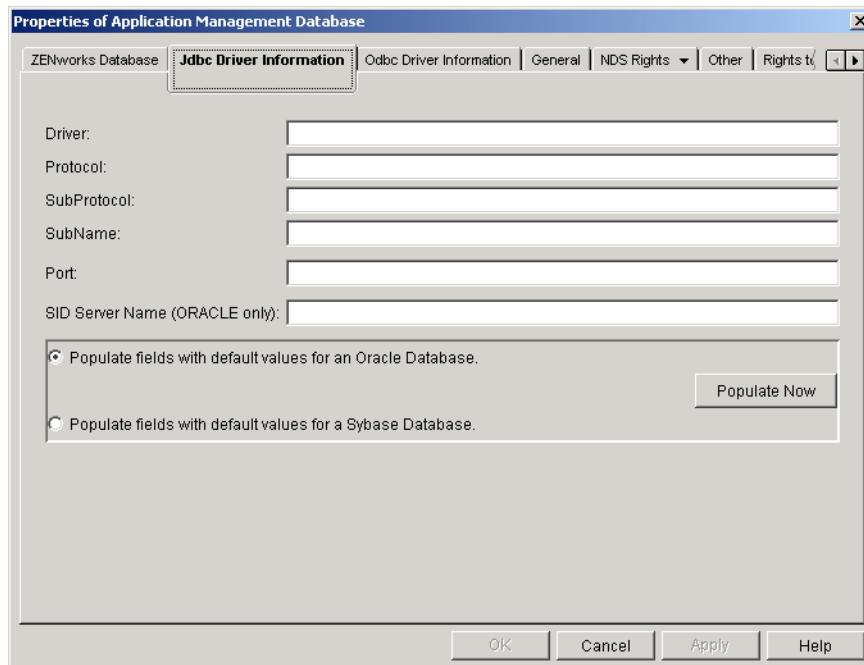
Configuring the Database's JDBC and ODBC Information

ConsoleOne® uses a JDBC driver to pull information from the database for a set of predefined Application Management reports. If you want to use these predefined reports, you need to configure the database object with the correct JDBC driver information.

Application Launcher/Explorer 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/Explorer will need.

To provide the JDBC and ODBC information:

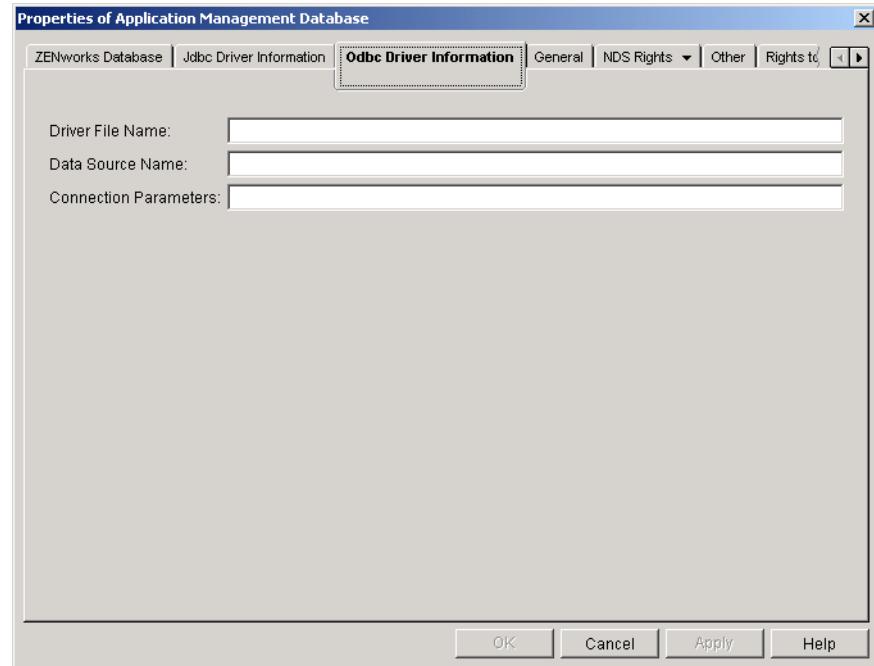
- 1** Right-click the database object > click Properties.
- 2** Click the JDBC Driver Information tab.



- 3** If you are using a Sybase or Oracle database, click the appropriate Populate option > click Populate Now to automatically populate the fields with the default information for a Sybase or Oracle database.

or

If you are using a different type of database, fill in the fields with the appropriate information for your database. Click the Help button for a description of each field.
- 4** Click the ODBC Driver Information tab.



5 Fill in the following fields:

Driver File Name: Enter the name of the ODBC driver file that resides on the workstation to enable Application Launcher/Explorer to access the database. For Sybase, the driver file name is dbodbc6.dll.

Data Source Name: Enter **NAL Reporting**.

Connection Parameters: Enter the appropriate commands to connect to the database. For Sybase, these commands are:

```
CommLinks=TCPIP{Host=IPAddress:2638};AutoStop=Yes;Integrated=No;DBN=NAL;  
ENG=IPAddress
```

where *IPAddress* is the actual IP address of the server where the database resides.

6 Click OK to save the JDBC and ODBC driver information.

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/Explorer to retrieve the ODBC information it needs to access the database.

To grant rights:

- 1** Right-click the ZENworks Database object > click Trustees of This Object.
- 2** Click Add Trustee.
- 3** Select [PUBLIC] > 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] > click Delete Property to remove it from the list.
- 5** Click Add Property to display the Add Property dialog box > select zendbODBCConnectionParameters > 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 two properties: zendbODBCDataSourceName and zendbODBCDriverFileName. Keep the default rights (Read and Compare).
- 7** Click OK > OK to save the changes.

Enabling the ZENworks Database Policy

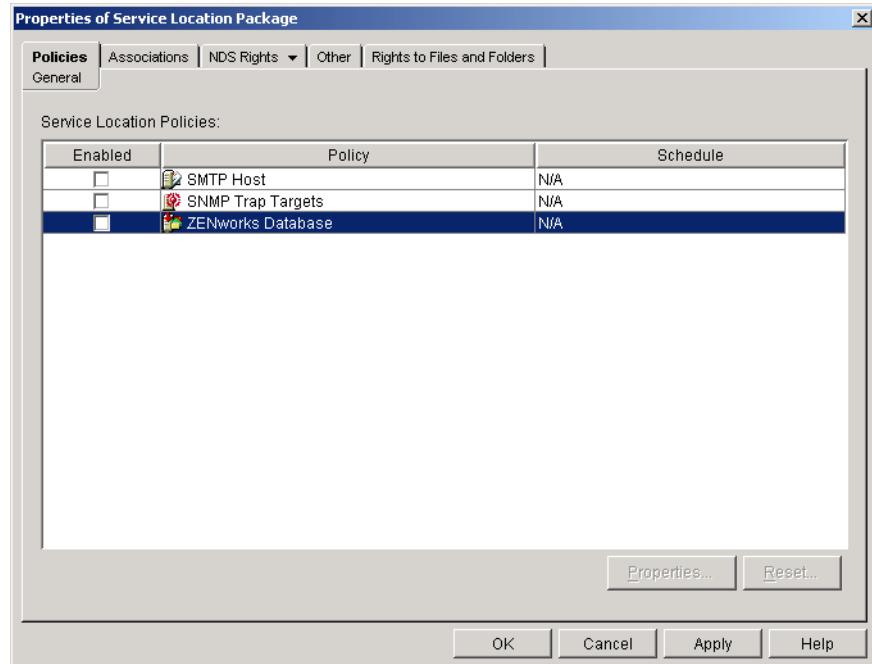
Before Application Management can use the database, you need to:

- ♦ Activate a Database policy in an NDS Service Location Package. The Database policy simply points to the location of the ZENworks database you are using. A Service Location Package can have one Database policy only, and each Database policy can point to one database only. If you haven't created a Service Location Package, or if your current packages' Database policies are being used for other databases, you will 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/Explorer 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 > click Properties.
or

If you do not have a Service Location Package, right-click the container where you want to create one > click New > click Policy Package. Follow the instructions provided in the Policy Package Wizard to create a Service Location Package.



- 2 On the General page, check the 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 Database Location page.
- 4 In the ZENworks Database DN field, browse for and select the database object you want to use for Application Management reporting. This should be the same object you configured in “Configuring the Database’s JDBC and ODBC Information” on page 97.
- 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 > 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.

Configuring Application Objects to Use Database Reporting

After you’ve set up the database you will use for reporting, you can start configuring Application objects to use database 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 database reporting, SNMP trap reporting, or log file reporting, see “Configuring Applications to Use Reporting” on page 104.

Setting Up SNMP Trap Reporting

If you have a management console to collect SNMP traps, you can have Application Launcher/Explorer send SNMP traps to the management console.

To set up SNMP trap reporting, you need to complete the following tasks:

- ◆ “[Enabling the SNMP Trap Targets Policy](#)” on page 102
- ◆ “[Configuring Application Objects to Use SNMP Trap Reporting](#)” on page 103

Enabling the SNMP Trap Targets Policy

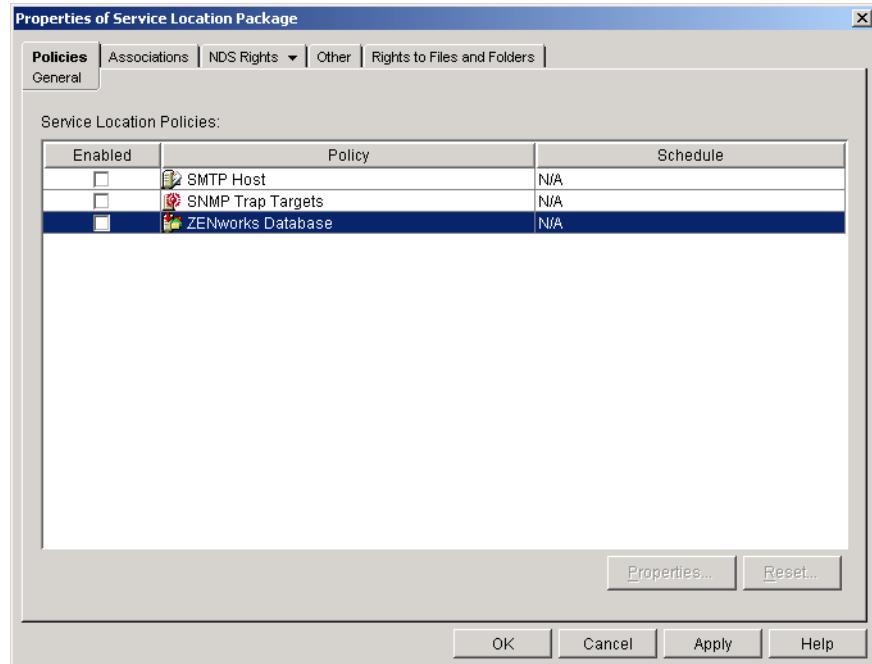
Before Application Management can use SNMP traps for event reporting, you need to:

- ◆ Activate an SNMP Trap Targets policy in an NDS Service Location Package. The SNMP Trap Targets policy simply points to the IP address (or addresses) of the management console that will display the traps. A Service Location Package can have one SNMP Trap Targets policy only. 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 will 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 who you want the SNMP Trap Targets policy applied to. This association is how Application Launcher/Explorer 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 > click Properties.
or

If you do not have a Service Location Package, right-click the container where you want to create one > click New > click Policy Package. Follow the instructions provided in the Policy Package Wizard to create a Service Location Package.



- 2 On the General page, check the box in the Enabled column to enable the SNMP Trap Targets Policy.
- 3 Select the policy in the list > click Properties to display the SNMP Trap Targets list.
- 4 Click Add to display the SNMP Target dialog box > enter the IP address of the workstation or server where the management console is running > 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 who 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 > 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.

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 “[Configuring Applications to Use Reporting](#)” on page 104.

Setting Up Log File Reporting

You can have Application Launcher/Explorer 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/Explorer will automatically authenticate them to the log file location.

To set up log file reporting, you need to complete the following tasks:

- ◆ “[Setting Up a Common Log File Location](#)” on page 104
- ◆ “[Configuring Application Objects to Use Log File Reporting](#)” on page 104

Setting Up a Common Log File Location

With log file reporting, you have two options. You can have Application Launcher/Explorer log events for each individual user to a file on the user's local drive, or you can have Application Launcher/Explorer log events for all users to a file in a common network location.

If you want Application Launcher/Application Explorer 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 Explorer/Launcher save the files to the users' local drives.

Configuring Application Objects to Use Log File Reporting

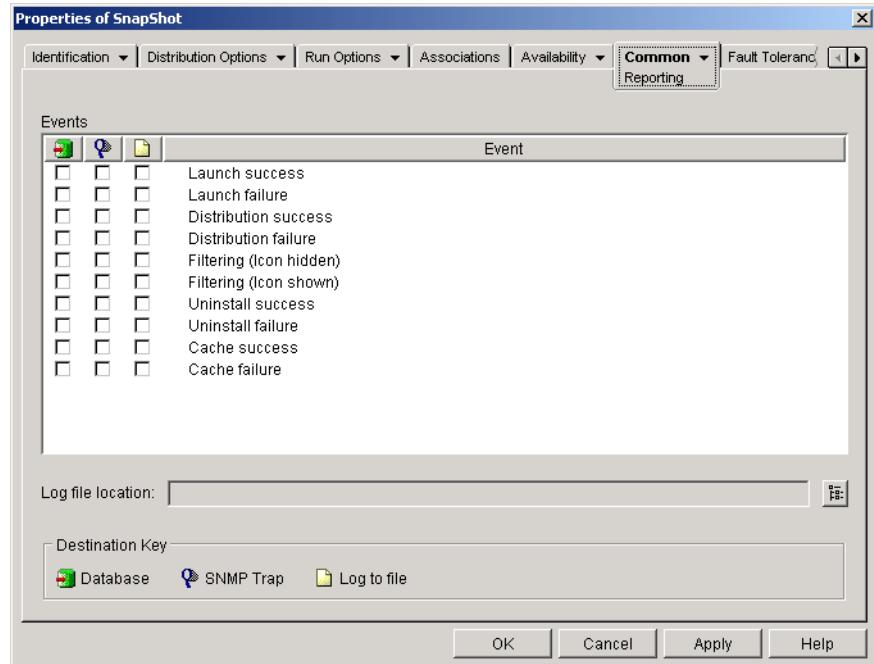
After you've set up a common log file location (if you plan to use a common log file), you can start configuring Application objects to use log file reporting. Because event reporting is configured on a per-application basis, you can choose which applications you want to collect event reports for and which ones you don't.

For detailed instructions about configuring an application to use log file reporting, database reporting, or SNMP trap reporting, see “[Configuring Applications to Use Reporting](#)” on page 104.

Configuring Applications to Use Reporting

Application Launcher/Explorer 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 > click Properties.
- 2 Click the Common tab > Reporting.



3 Fill in the following fields:

Events: Select the events (launch, distribution, filtering, uninstall, and cache) that you want reported and the destination where you want the events sent. You can send events to a database, to a management console via an SNMP trap, or to a text log file. If necessary, you can send events to multiple destinations (for example, both a database and a text log file).

Log File Location: If you selected a log file as the destination for any of the event reports, click the Browse button to display the Open dialog box > browse to and open the folder in which you want the log file located > type a name for the log file in the File Name box > click OK.

If you want to save the events for all users to the same log file, specify a log file in a common network location (one that users have Read and Write rights to). If you want users to have individual log files, specify a location on their local drive (for example, c:\temp).

4 Click OK to save the changes.

Generating Database Reports

When an event is recorded to a database, the following information is included: 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.

You can use a database query to search for specific information. See your database documentation for information about performing queries.

In addition, Application Management includes a set of predefined reports for you to use. There are six basic reports that let you report on the success or failure of an event either by application, by user, or by workstation. For example, you could use the Failure by App report to view a list of all application distributions, sorted by user, that failed. Or you could use the Failure by User to view a list of all application launches, sorted by user, that failed.

To access the predefined reports:

- 1 Right-click the ZENworks Database object > click Reporting.
- 2 In the Available Reports list, expand the ZENworks NAL Reports category > select the report you want.
- 3 In the Event Type list, select the event you want to include in the report.
- 4 Click Run Selected Report.

Understanding Log File Reports

The following is a log file entry for a single event. Each field in the entry is described below.

```
"LaunchFail", "11", "5/10/2000:8:28PM",
"MAIN_TREE::ssmith.xyzcorp",
"MAIN_TREE::ssmith_winnt.xyzcorp", "01010480:00C04F6AABC6",
"MAIN_TREE::testapp.org.xyzcorp",
"119C2580-949-11D3-BD13-00C04F6AABC6",
"5", "-53429", "", "", "", "", "", "", "524288"
```

Field	Example	Description
Event Type	LaunchFail	The event that occurred and whether it was successful or failed. Possible event types are: LaunchSuccess LaunchFail DistributionSuccess DistributionFail FilterShow FilterHide UninstallSuccess UninstallFail CacheSuccess CacheFail
Event Type Code	11	The code associated with the event. Possible codes are: 10 - LaunchSuccess 11 - LaunchFail 20 - DistributionSuccess 21 - DistributionFail 30 - FilterHide 40 - FilterShow 50 - UninstallSuccess 51 - UninstallFail 60 - CacheSuccess 61 - CacheFail
Date and Time	5/10/2000:8:28PM	The date (5/10/2000) and time (8:28PM) the event occurred.
User's Distinguished Name and Tree	MAIN_TREE::ssmith.xyzcorp	The distinguished name and tree of the user for which the event occurred.

Field	Example	Description
Workstation Distinguished Name and Tree	MAIN_TREE::ssmith_winnt.xy zcorp	The distinguished name and tree of the workstation on which the event occurred. This field contains information only if the workstation has been imported into NDS as a Workstation object.
Workstation Address	01010480:00C04F6AABC6	The IPX™ or IP address of the workstation on which the event occurred.
Application Distinguished Name and Tree	MAIN_TREE::testapp.org.xyz corp	The distinguished name and tree of the Application object for which the event occurred.
Application GUID	119C2580-949-11D3-BD13-00C04F6AABC6	The global unique identifier assigned to the Application object. The GUID is located on the Application object's Options page (Distribution Options tab).
Application Version Number	5	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).
Event Code Major	-53429	The error code generated by Application Launcher/Explorer.
Event Code Minor	no example	Additional error code.
Event String 1	no example	Error string.
Event String 2	no example	Error string.
Event String 3	no example	Error string.
Event String 4	no example	Error string.
Event String 5	no example	Error string.
Application Flag	no example	Application object bitmask

Enabling Windows Installer Verbose Logging

When you distribute an application based on a .MSI package rather than a .AOT/.AXT package, Application Launcher/Explorer 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 a MSIxxxx.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 will create a log file, ZAPPMSI.LOG, in the user's temporary directory on the workstation.

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 MSIxxxx.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/hh/msi/erro_89f7.asp) (http://msdn.microsoft.com/library/default.asp?url=library/en-us/msi/hh/msi/erro_89f7.asp) at the Microsoft Developer Network (MSDN) site.

For information about Windows Installer error codes returned through Application Launcher/Explorer, see the [Error Codes document](http://msdn.microsoft.com/library/default.asp?url=library/en-us/msi/hh/msi/code_13ub.asp) (http://msdn.microsoft.com/library/default.asp?url=library/en-us/msi/hh/msi/code_13ub.asp) at the Microsoft Developer Network site.

12 Metering Software Licenses

ZENworks® for Desktops (ZfD) Application Management integrates with Novell® Licensing Services (NLS) to enable you to track an application's usage and comply with the application's license agreement. When a user launches an application that has been configured as part of NLS, Novell Application Launcher™/Explorer checks to make sure that a license is available before running the application.

To set up software metering, complete the tasks in the following sections:

- Create a separate License container and one or more Metered Certificates for each application you want to track. For instructions, see the NLS documentation at the [Novell Documentation Web site](http://www.novell.com/documentation) (<http://www.novell.com/documentation>).
- Configure the Application object in NDS® to use NLS and software metering. You won't be able to complete this step until after you've created an Application object for the application. For details about creating an Application object, see [Chapter 6, "Distributing Applications," on page 53](#). For details about configuring the Application object to use software metering, see [Chapter 12, "Metering Software Licenses," on page 109](#).

Installing Novell Licensing Services

Novell Licensing Services (NLS) must be installed before you can use Application Launcher/Explorer to meter software licenses. NLS is included with NetWare 4.x, 5.x, and 6, and with Novell Cluster Services™. For information about installing NLS, see the documentation for the above listed products at the [Novell Documentation Web site](http://www.novell.com/documentation) (<http://www.novell.com/documentation>).

Because NLS administration is performed through NetWare® Administrator, software metering is not available in a pure Windows 2000 environment.

Creating License Containers and Metered Certificates

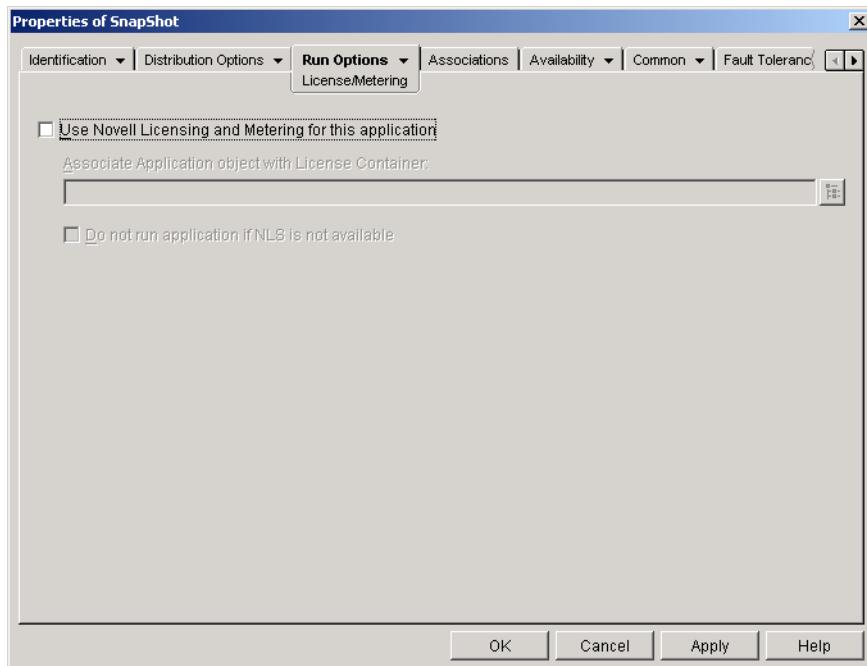
You must create a License container for each application you want to meter. Within the License container, you then need to create one or more Metered Certificates that account for the number of licenses you have available for the application. For example, if you initially have 200 licenses for the application, 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. For instructions about creating License containers and Metered Certificates, see the NLS documentation at the [Novell Documentation Web site](http://www.novell.com/documentation) (<http://www.novell.com/documentation>).

Configuring Applications to Use License Metering

Once you have installed NLS and have a License container and Metered Certificate created for an application, you need to configure the Application object to use NLS and license metering. This enables Application Launcher/Explorer to enforce the licensing you've established for the application.

To enable Application Launcher/Explorer to enforce an application's licensing:

- 1** In ConsoleOne, right-click the Application object > click Properties.
- 2** Click the Run Options tab > License/Metering.



- 3** Check the Use Novell Licensing and Metering for this Application 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/Explorer to run the application when NLS is not available, check the Do Not Run if NLS Is Not Available box. Otherwise, Application Launcher/Explorer will run the application.
- 6** Click OK.

13 SnAppShot

To help you create installation packages for applications installed to Windows® 95/98 or Windows NT®/2000/XP workstations, ZENworks® for Desktops (ZfD) Application Management provides the snAppShot™ utility. The following sections provide information about snAppShot and instructions for using the utility.

- ◆ “[Understanding SnAppShot](#)” on page 111
- ◆ “[Preparing a SnAppShot Workstation](#)” on page 112
- ◆ “[Creating an Installation Package](#)” on page 112
- ◆ “[Command Line Switches](#)” on page 113

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.

Application Object Templates

You use the Application object template file to create the Application object in NDS®. 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 NDS 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.

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™/Explorer uses these source files when installing the application to the workstation.

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
```

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 the Novell Client™ installed.
 - 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 GX100 running Windows 2000.
- Depending on the application and what occurs during an install, it may be necessary to create different Application objects to be used for different types of workstations.

Creating an Installation Package

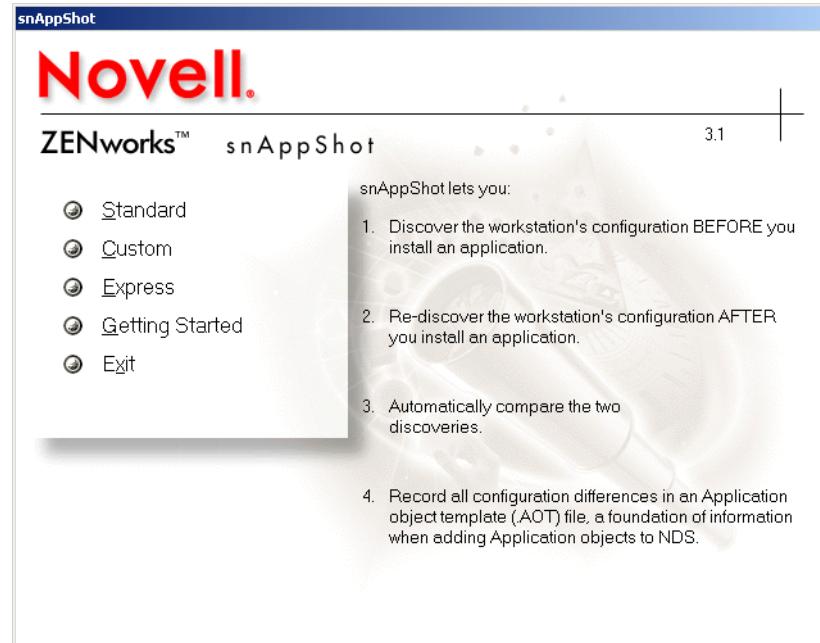
To create an installation package, snAppShot uses the following process:

- 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.
- Lets 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 “[Preparing a SnAppShot Workstation](#)” on [page 112](#)), start snAppShot (SNAPSHOT.EXE) from the SYS:\PUBLIC\SNAPSHOT directory.

For information about switches you can use when starting snAppShot, see “[Command Line Switches](#)” on [page 113](#).



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.

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 will retrieve 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 will run slower.

14 Application Object Settings

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.

- ◆ “Identification Tab” on page 115
- ◆ “Distribution Options Tab” on page 122
- ◆ “Run Options Tab” on page 140
- ◆ “Associations Tab” on page 148
- ◆ “Availability Tab” on page 150
- ◆ “Common Tab” on page 166
- ◆ “Fault Tolerance Tab” on page 183

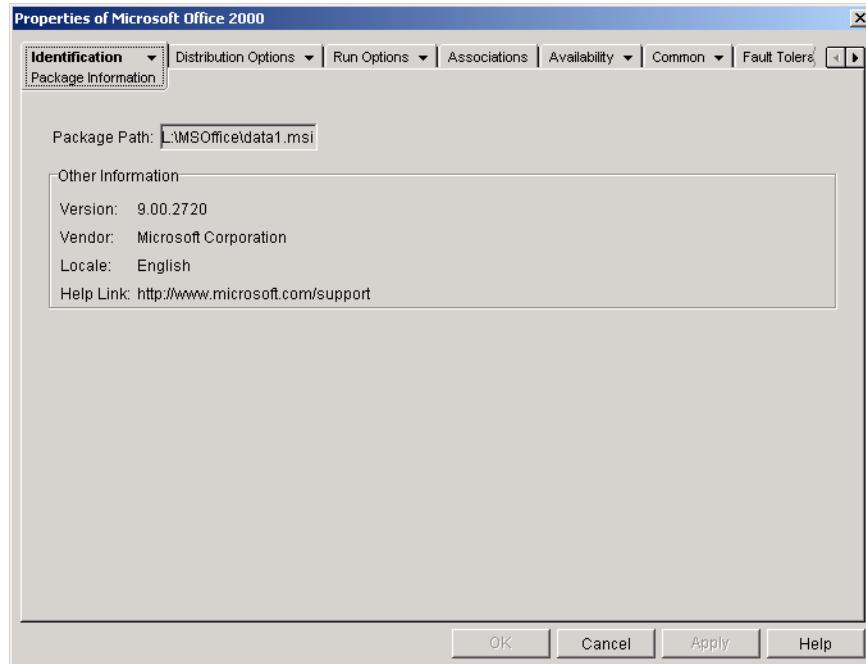
Identification Tab

The Identification tab includes the following pages to help you configure how the Application object is displayed to users:

- ◆ “Package Path” on page 116
- ◆ “Icon Page” on page 116
- ◆ “Description Page” on page 118
- ◆ “Folders Page” on page 119
- ◆ “Contacts Page” on page 121
- ◆ “Administrator Notes Page” on page 122

Package Information Page (.MSI Application Object Only)

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.



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

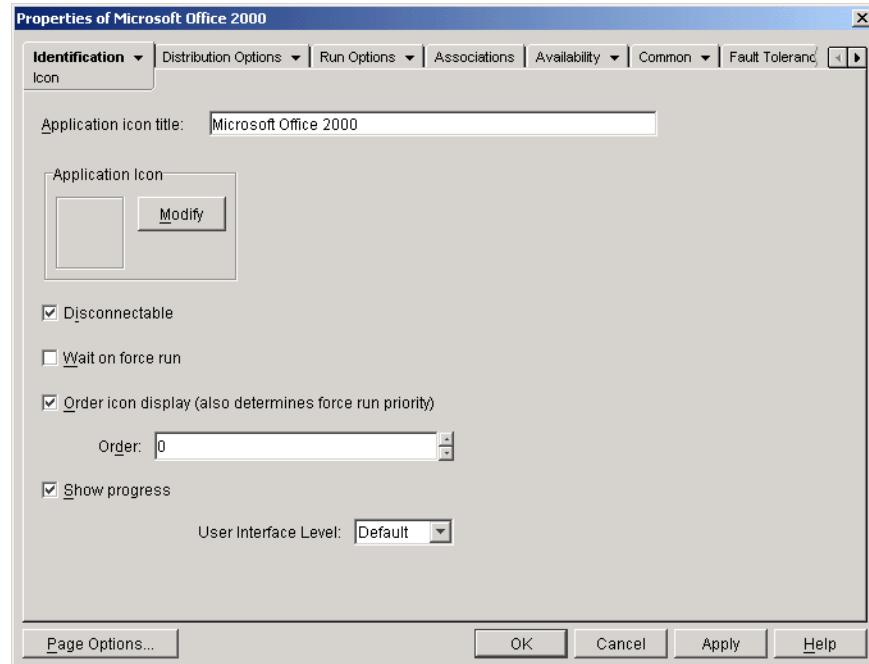
Displays the locale defined in the .MSI file.

Help Link

Displays the Web location to visit for information and help about the application.

Icon Page

The Icon property page, shown below, determines the Application object's icon that Novell® Application Launcher™ /Explorer 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/Explorer should continue to display the Application object on the workstation when the user has disconnected from NDS®.



Application Icon Title

Enter the text you want to appear as the title for the Application object's icon wherever the icon is displayed. If necessary, you can use the Description page (Identification tab) for longer descriptions of the application.

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

Check this option to mark the application as being able to be run on a workstation that is disconnected from NDS.

IMPORTANT: The application must be distributed to the workstation before the user can run it in disconnected mode, or the application must be cached on the workstation. You can force the application to be distributed 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).

Check this option to force the application to wait until the application before it terminates. The application order is defined in the Order Icon Display (Also Determines Force Run Priority) field. Reboots will be queued until the final application has terminated.

Order Icon Display (Also Determines Force Run Priority)

This option dictates the order in which applications set as Force Run will be started. Select this option to enable it, and then use the Order list to select the application's position in the Force Run order.

IMPORTANT: Despite the field title (Order Icon Display), this option does not determine the order in which Application Launcher/Explorer displays icons. Icon order is determined by the user through the View > Arrange Icon setting in Application Launcher/Explorer.

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 999. 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/Explorer will run the application without waiting for the previously started application to terminate, unless the Wait on Force Run option is enabled.

Show Progress

This option displays a progress bar to users anytime 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 will take.

User Interface Level

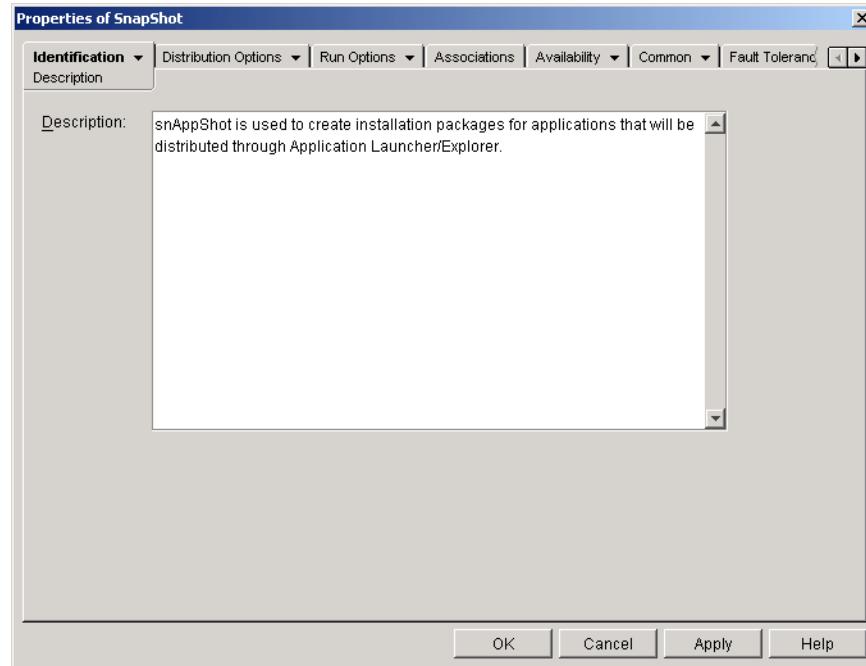
This option appears only if the Application object uses a Microsoft Windows Installer (.MSI) package. During distribution of a .MSI Application object, Application Launcher/Explorer launches Windows Installer to install the application. As a result, rather than showing the standard Application Launcher/Explorer 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 will display 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 full user interface with Wizard dialog boxes suppressed.
- ◆ Full: Displays full user interface (Wizard dialog boxes, progress information, error messages and prompts, and so forth).

Application Launcher/Explorer passes the selected setting to Windows Installer as a startup parameter. For more information about these settings, see the Microsoft Windows Installer documentation.

Description Page

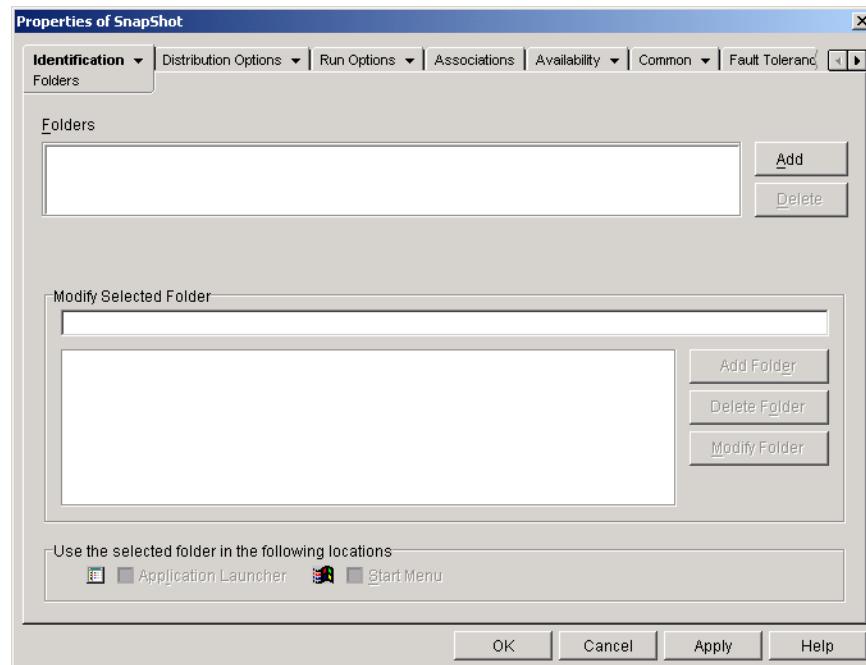
The Description property page, shown below, provides users more complete information about the Application object than the application icon title allows.



If you have enabled the Prompt User Before Distribution option (Distribution Options > Options page), users will see this description when Application Launcher/Explorer 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) > click Properties.

Folders Page

The Folders property page, shown below, lets you specify the folders where you want Application Launcher/Explorer to place the application when distributing it to a workstation.



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 otherwords, WINAPPS\CALCULATOR\CALC.EXE and WINAPPS\NOTE PAD\NOTE PAD.EXE).

A linked folder is simply an association to an existing Application Folder object. The Application Folder object must already exist in NDS. If the Application Folder object includes multiple folders (a folder structure) you can add the application to any folder in the structure.

HINT: 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 will cause Application Launcher/Explorer 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/Explorer will create (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 Folders below).

Delete

Select a folder in the folder list > click Delete to remove the folder from the list.

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 > click Add to add a subfolder to the folder. After you add a subfolder and Application Launcher/Explorer refreshes, users will see the application in the subfolder rather than in the folder.

Delete Folder

Select the folder in the folder tree > click Delete to delete the folder.

Modify Folder

Select the folder in the folder tree > click Modify 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*.

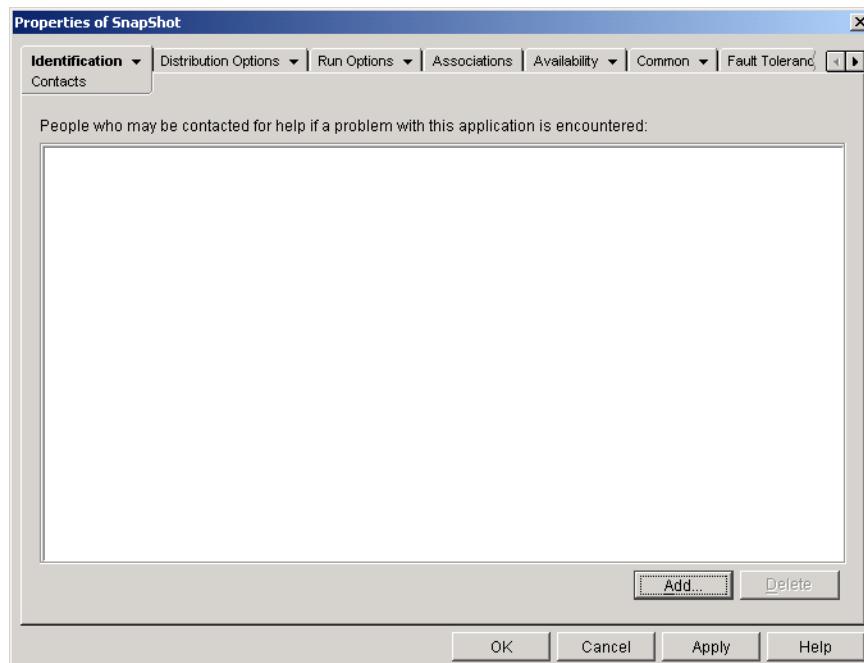
Use the Selected Folder in the Following Locations

Application Launcher/Explorer can display folders on the Windows Start menu and in the Application Launcher/Explorer window (provided these locations are enabled on the Application object's Associations page). Select a folder in the Folders list > check the boxes for the locations where you want to use the folder.

If you don't select either location, Application Launcher/Explorer will still display the Application object on the Start menu and in the Application Launcher/Explorer window (if these locations are enabled on the Associations page), but the object will not appear in the folders you've defined.

Contacts Page

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 > click Help Contacts.

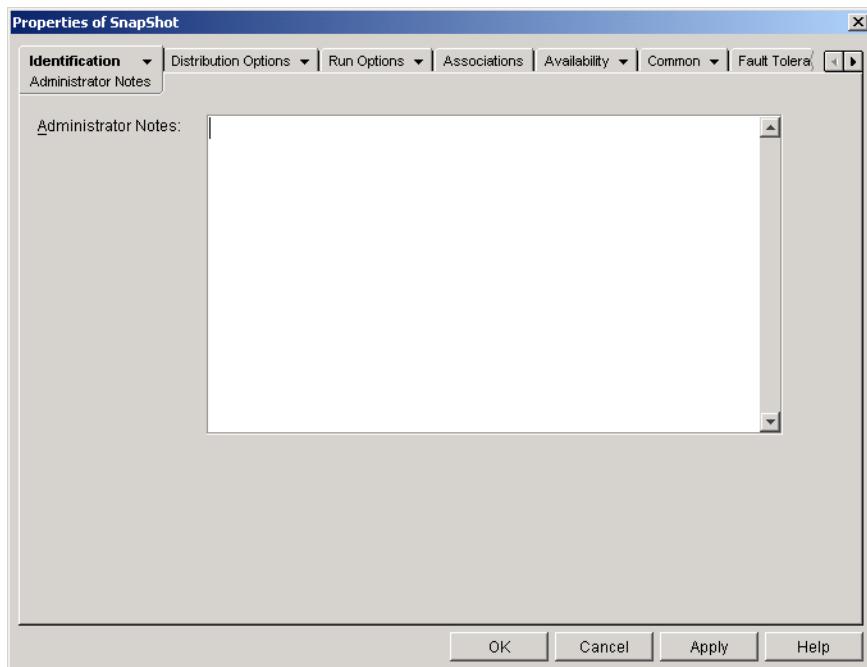


You can tailor the information to direct users to the support staff at their locations. If you enter the support contact's e-mail address, users will be able to send an e-mail message directly from the Help Contacts page of the Properties dialog box.

NOTE: Users must have the NDS rights required to read the E-Mail Address (MailboxID) and Telephone (Telephone Number) attributes of the users defined as contacts.

Administrator Notes Page

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.



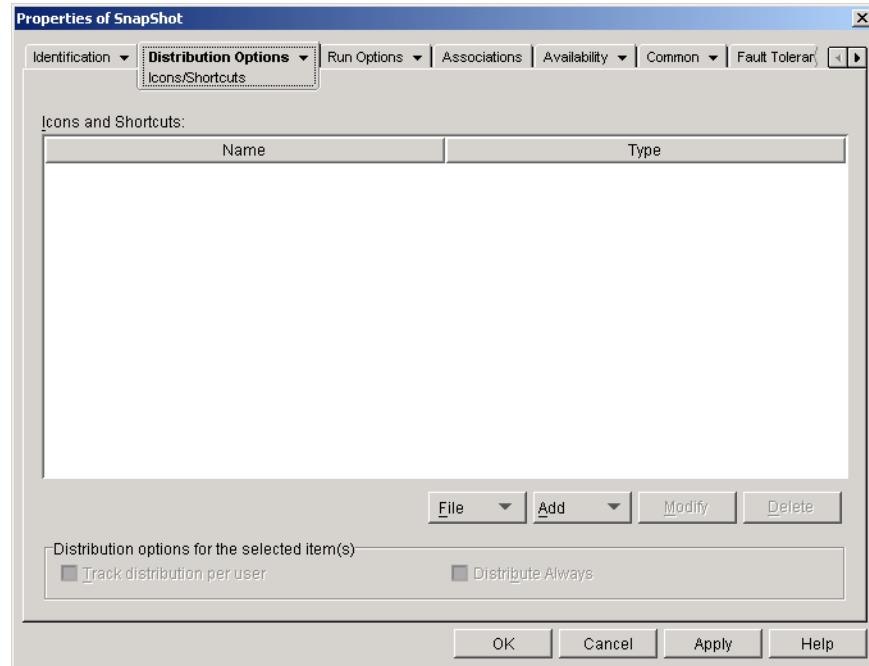
Distribution Options Tab

The Distribution Options tab includes the following pages to help you configure how the Application object is distributed to users:

- ◆ “Icons/Shortcuts Page (.AOT/.AXT Application Object Only)” on page 122
- ◆ “Registry Page” on page 124
- ◆ “Application Files Page (.AOT/.AXT Application Object Only)” on page 127
- ◆ “INI Settings Page” on page 130
- ◆ “Text Files Page (.AOT/.AXT Application Object Only)” on page 132
- ◆ “Distribution Scripts Page” on page 135
- ◆ “Pre-Install Schedule Page (.AOT/.AXT Application Object Only)” on page 136
- ◆ “Options Page” on page 138

Icons/Shortcuts Page (.AOT/.AXT Application Object Only)

The Icons/Shortcuts property page, shown below, determines the icons and shortcuts that Application Launcher/Explorer will create 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.



The icons and shortcuts you add with this page are in addition to the Application object's icon. Although the Application object's icon may 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 created and configure the Application object to run one time (Run Options > Applications). When a user selected the Application object, Application Launcher/Explorer would run the application one time, create the icons and shortcuts, perform any other tasks specified by the Application object's properties, and then remove the Application object's icon from the workstation. Thereafter, the user would need to select the icon or shortcut to launch the application.

IMPORTANT: Program groups and program group items are supported on Windows 95/98 workstations but not on Windows NT*/2000/XP workstations. Shortcuts are supported on all four Windows versions.

Icons and Shortcuts

This list displays the icons and shortcuts that will be created when the application is distributed to a workstation.

File

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.

Add

Click Add to add a new program group, program group item, or shortcut.

Modify

Select an icon or shortcut from the Icons and Shortcuts list > click Modify to change the information associated with it.

Delete

Select an icon or shortcut from the Icons and Shortcuts list > 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 > check Track Distribution Per User.

Distribute Always

By default, Application Launcher/Explorer will only create the icons and shortcuts defined in the Icons and Shortcuts list at the following times:

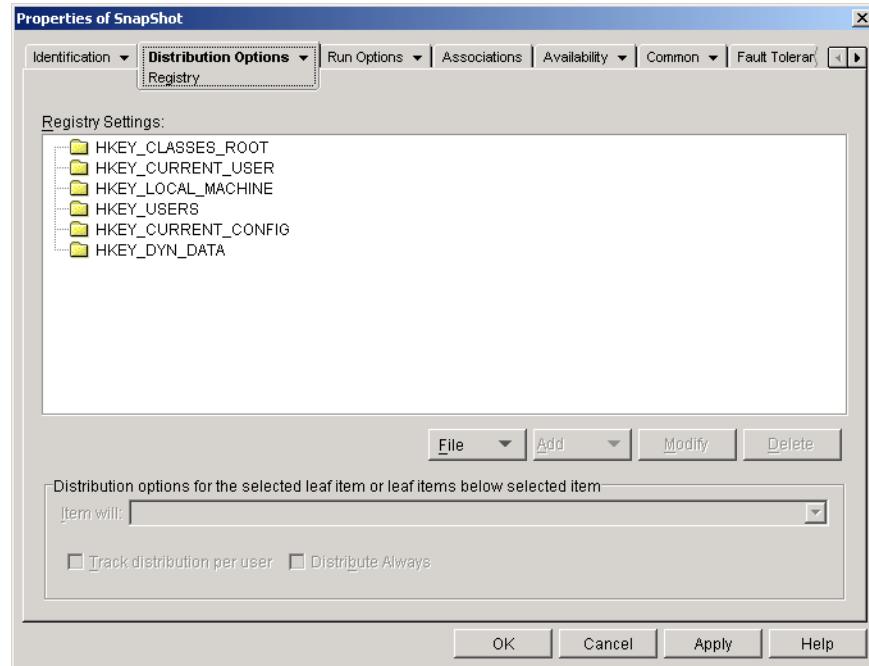
- The first time the application is launched on a workstation.
- The first time the application is launched after the application's version number (Distribution Options tab > Options page) has been changed.

To force Application Launcher/Explorer to create an icon or shortcut each time the application is launched, select the icon or shortcut in the Icons and Shortcuts list > check Distribute Always.

If the user has a NALCACHE directory on his or her local machine, Application Launcher/Explorer uses the information stored in the NALCACHE directory to create the icon or shortcut. If the user does not have a NALCACHE directory (for example, the user is running Application Launcher/Explorer through a terminal server client session) or if writing to the cache has been disabled for the user (User object > Application Launcher tab > Launcher Config tab > Enable Writing to Cache option), Application Launcher/Explorer uses the information stored in Novell eDirectory.

Registry Page

The Registry property page, shown below, determines the registry settings that Application Launcher/Explorer will create or delete when distributing the application to a workstation.



Registry Settings

The Registry Settings tree displays all settings that have been defined for the registry. When the application is distributed to a workstation, Application Launcher/Explorer will modify the workstation's registry according to the settings in this tree.

If you used a .AOT, .AXT, or .MSI file when creating the Application object, the tree automatically includes all registry settings that are defined in those templates.

File

Click File > Find to search for specific keys, value names, or value data in the registry.

Click File > Import to import registry settings from another Application object's .AOT or .AXT file, or from a registry file (.REG).

Click File > Export to export the registry settings to a registry file (.REG). To export the settings to a .AOT or .AXT file format, you must export the entire Application object using the Export Application Object option located on the Tools > 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 will be 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 > 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.
- **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.
- ◆ **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 15, “Macros,” on page 189](#).

Modify

Select the key or value you want to modify > click Modify.

Delete

Select the key or value you want to delete > click Delete. When you delete a key, everything subordinate to the key will also be deleted.

Distribution Options

Use these options to set individual distribution options for settings included in the Registry Settings tree.

Item Will

Use this option to specify whether the registry setting will be created or deleted during distribution.

In the Registry Settings tree, select the setting > select either Create or Delete from the Item Will list. If you delete a setting that has subordinate settings, the subordinate settings will also be deleted.

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 > check Track Distribution Per User.

Distribute Always

By default, Application Launcher/Explorer will only distribute the registry modifications defined in the Registry Settings list 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.

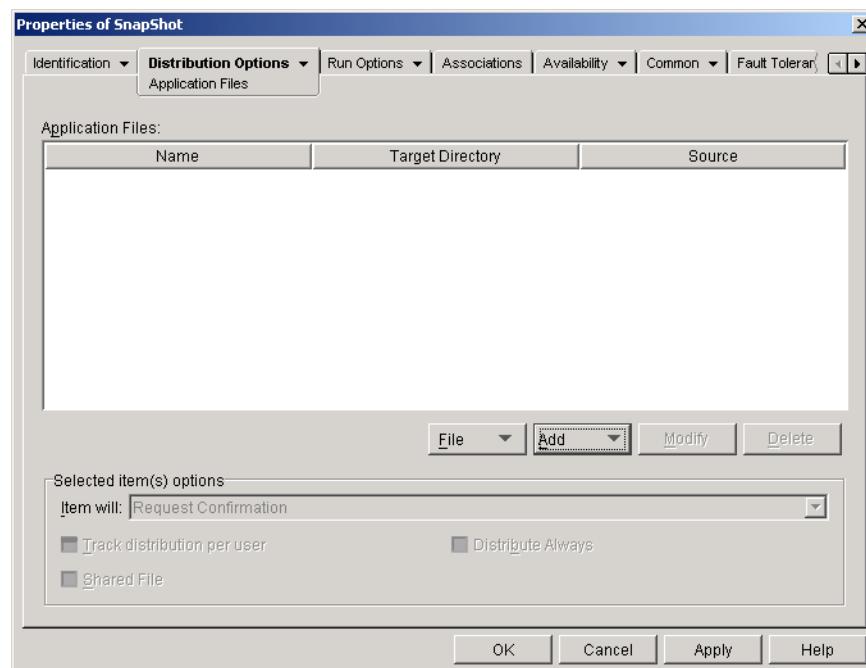
To force Application Launcher/Explorer to distribute a registry modification each time the application is launched, select the registry setting in the Registry Settings list > check Distribute Always.

If the user has a NALCACHE directory on his or her local machine, Application Launcher/Explorer uses the setting information stored in the NALCACHE directory to modify the registry.

If the user does not have a NALCACHE directory (for example, the user is running Application Launcher/Explorer through a terminal server client session) or if writing to the cache has been disabled for the user (User object > Application Launcher tab > Launcher Config tab > Enable Writing to Cache option), Application Launcher/Explorer uses the setting information stored in Novell eDirectory.

Application Files Page (.AOT/.AXT Application Object Only)

The Application Files property page, shown below, specifies the application files that Application Launcher/Explorer will install or remove when distributing the application to a workstation.



Application Files

The Application Files list displays all files and directories that will be installed or removed during distribution. The name, target directory (the location on the workstation where the file will be 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 a .AOT or .AXT file when creating the Application object, the list automatically includes all files and directories that are defined in those templates.

File

Click File > Find to search 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.

Click File > Import to import application files and directories from another Application object's .AOT or .AXT 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 will be installed to or removed from the workstation during distribution.

- ♦ **File:** To add a file to the list, click Add > 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 will be 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 will 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 > check the Target File to Be Deleted box.

- ♦ **Directory:** To add a directory to the list, click Add > Directory to display the Edit Directory dialog box.

In the Name field, specify the directory to create or delete on the workstation (for example, C:\PROGRAM FILES\NOVELL). You can also substitute a macro for the directory path (for example, %DIRECTORY_TARGET_PATH%\NOVELL).

Check the Directory to Be Deleted box if you want to delete the directory from the workstation. Otherwise, the directory will be created on the workstation.

Modify

Select the file or folder you want to modify > click Modify.

Delete

Select the file or folder you want to delete > click Delete to remove it from the Application Files list.

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 a .EXE or .DLL 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 > 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 file or folder you want to track > check Track Distribution Per User.

Distribute Always

By default, Application Launcher/Explorer will only distribute the file and folder modifications defined in the Application Files list 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/Explorer to distribute a file or folder modification each time the application is launched, select the file or folder in the Application Files list > check Distribute Always.

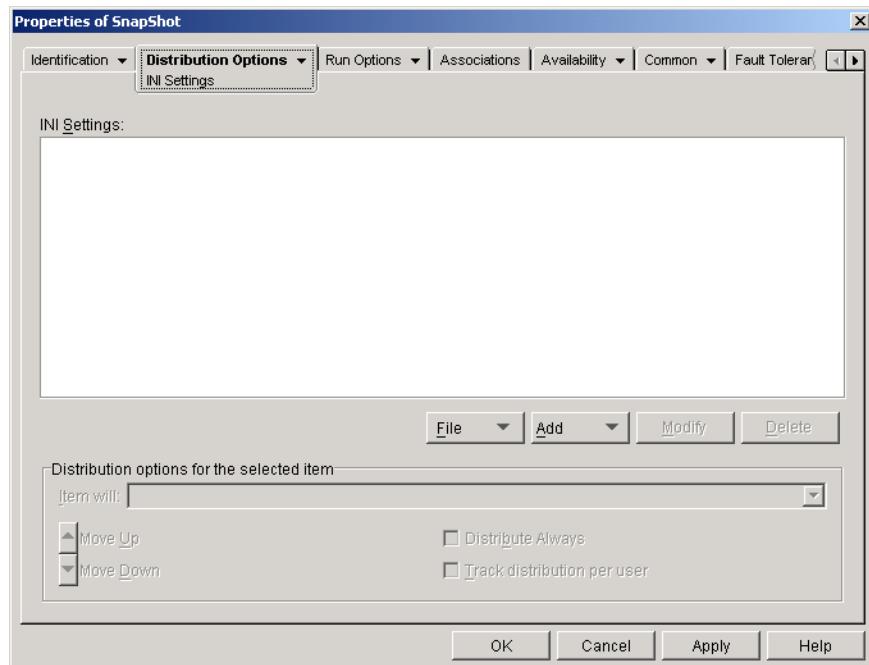
If the user has a NALCACHE directory on his or her local machine, Application Launcher/Explorer uses the information stored in the NALCACHE directory to install or remove the file or folder. If the user does not have a NALCACHE directory (for example, the user is running Application Launcher/Explorer through a terminal server client session) or if writing to the cache has been disabled for the user (User object > Application Launcher tab > Launcher Config tab > Enable Writing to Cache option), Application Launcher/Explorer uses the information stored in Novell eDirectory™.

Shared File

Use this option to mark a file as a shared file (that is, one that will be 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.

INI Settings Page

The INI Settings property page, shown below, determines the .INI settings that Application Launcher/Explorer will create or delete when distributing the application to a workstation.



INI Settings

The INI Settings tree displays the .INI files that have been defined for the application, as well as all sections and values that have been added to each .INI file. When the application is distributed to a workstation, the .INI files are created if necessary, or existing .INI files are modified as needed.

If you used a .AOT, .AXT, or .MSI file when creating the Application object, the tree automatically includes all .INI settings that are defined in those templates.

File

Click File > Find to search for specific files, sections, or values.

Click File > Import to import .INI settings from another Application object's .AOT or .AXT file, or from a .INI file.

Click File > Export to export the settings to a .INI file. To export the settings to a .AOT or .AXT file, you must export the entire Application object using the Export Application Object option located on the Tools > Application Launcher Tools menu.

Add

This option lets you add INI settings to the INI Settings tree. Only settings displayed in the INI Settings tree will be 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 > choose one of the following options:

- ◆ **File:** Adds a file to the INI Settings tree. 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/Explorer will create the file or section before adding the value.

You can use a macro for a section name, value name, or value data.

Modify

Select the file, section, or value you want to modify > click Modify.

Delete

Select the file, section, or value you want to delete from the INI Settings tree > click Delete. When deleting a file or section, everything subordinate to it will also be deleted.

Distribution Options

Use these options to set individual distribution options for value settings included in the INI Settings tree.

Item Will

Select a value in the INI Settings tree > select one of the following options from the Item Will list:

- ◆ **Create Always:** 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.

Track Distribution Per User

If you have implemented roaming user profiles, use this option to ensure that particular INI settings are distributed to each workstation a user logs in to. You should enable this option for all INI settings that are not saved as part of roaming user profiles.

In the INI Settings list, select the desired INI modification > check Track Distribution Per User.

Distribute Always

By default, Application Launcher/Explorer will only distribute the .INI modifications defined in the INI Settings list 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/Explorer to distribute an INI modification each time the application is launched, select the INI setting in the INI Settings list > check Distribute Always.

If the user has a NALCACHE directory on his or her local machine, Application Launcher/Explorer uses the information stored in the NALCACHE directory to make the INI modification. If the user does not have a NALCACHE directory (for example, the user is running Application Launcher/Explorer through a terminal server client session) or if writing to the cache has been disabled for the user (User object > Application Launcher tab > Launcher Config tab > Enable Writing to Cache option), Application Launcher/Explorer uses the information stored in Novell eDirectory™.

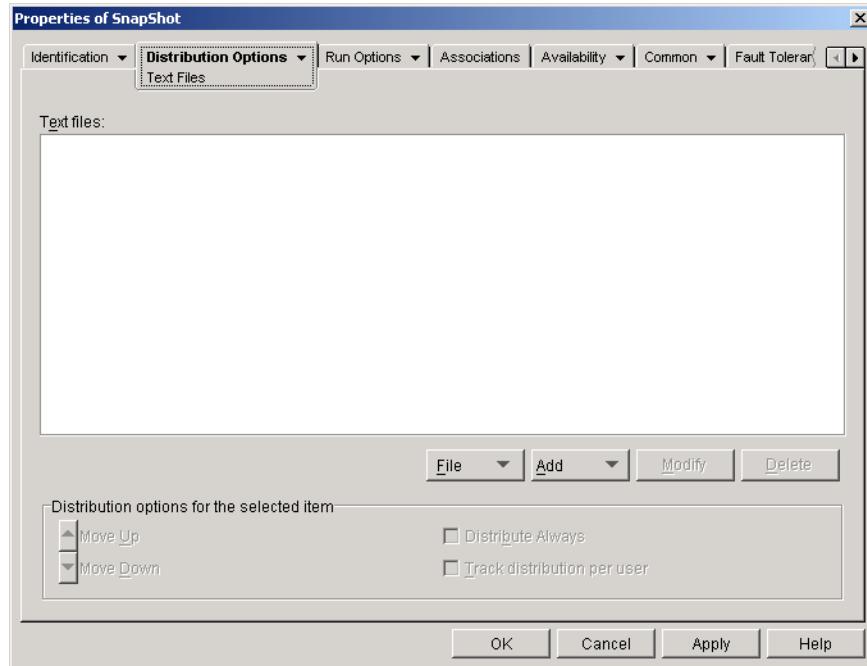
Move Up / Move Down

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 > click Move Up or Move Down.

Text Files Page (.AOT/.AXT Application Object Only)

The Text Files property page, shown below, determines the modifications that Application Launcher/Explorer will make to text files (such as CONFIG.SYS and AUTOEXEC.BAT) when distributing the application to a workstation.



Text Files

The Text Files tree shows all the text files that will be modified or added when the application is distributed. Each modification to a file is displayed subordinate to the file.

File

Click File > Find to search for specific files or information in the Text Files tree.

Click File > Import to import text files from another Application object's .AOT or .AXT file.

Add

This option lets you add text file modifications to the Text Files tree. Only the modifications displayed in the Text Files tree will be 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.

In general, you should specify a path if possible. If you enter the filename only, Application Launcher/Explorer will search all directories specified in the workstation's PATH environment variable. If it does not find a matching filename, it will assume the file doesn't exist and create 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 may 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.

Modify

To change the name of a text file, select the file in the Text Files tree > click Modify > enter 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 > 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

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. Check 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/Explorer will distribute the text file modifications defined in the Text Files list 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/Explorer to distribute a text file modification each time the application is launched, select the modification in the Text Files list > check Distribute Always.

If the user has a NALCACHE directory on his or her local machine, Application Launcher/Explorer uses the information stored in the NALCACHE directory to make the modification. If the user does not have a NALCACHE directory (for example, the user is running Application Launcher/Explorer through a terminal server client session) or if writing to the cache has been disabled for the user (User object > Application Launcher tab > Launcher Config tab > Enable Writing to Cache option), Application Launcher/Explorer uses the information stored in Novell eDirectoryTM.

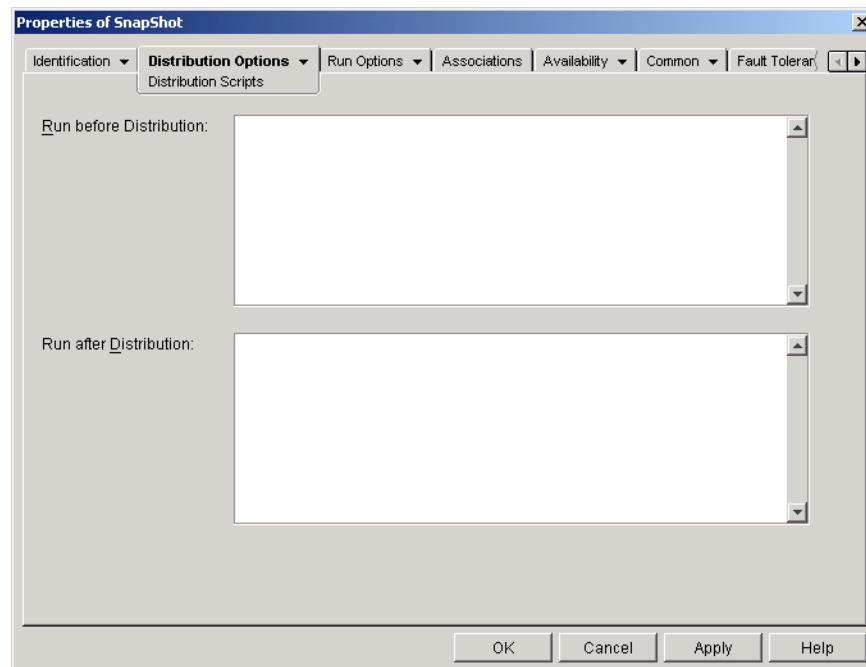
Track Distribution Per User

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 > check Track Distribution Per User.

Distribution Scripts Page

The Distribution Scripts property page, shown below, defines the scripts Application Launcher/Explorer will execute when distributing the application.



The scripts support the same commands and syntax as the Novell Client™, with the exception of the ones listed below. 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>).

Application Launcher/Explorer will not output anything to the screen, display errors, or support the following commands:

CLS
DISPLAY
EXIT
FDISPLAY
INCLUDE
LASTLOGINTIME
NO_DEFAULT
NOSWAP
PAUSE
PCOMPATIBLE
SCRIPT_SERVER
SET_TIME
SWAP
WRITE

IMPORTANT: On Windows NT/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 will be halted 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 will be run in secure system space, will never be displayed to the user.

Run Before Distribution

Use this text window to enter any script commands you want executed before the application is distributed. The script is executed in the order shown below:

1. Run Before Launching script executed (Run Options > Launch Scripts page)
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 executed (Run Options > Launch Scripts page)

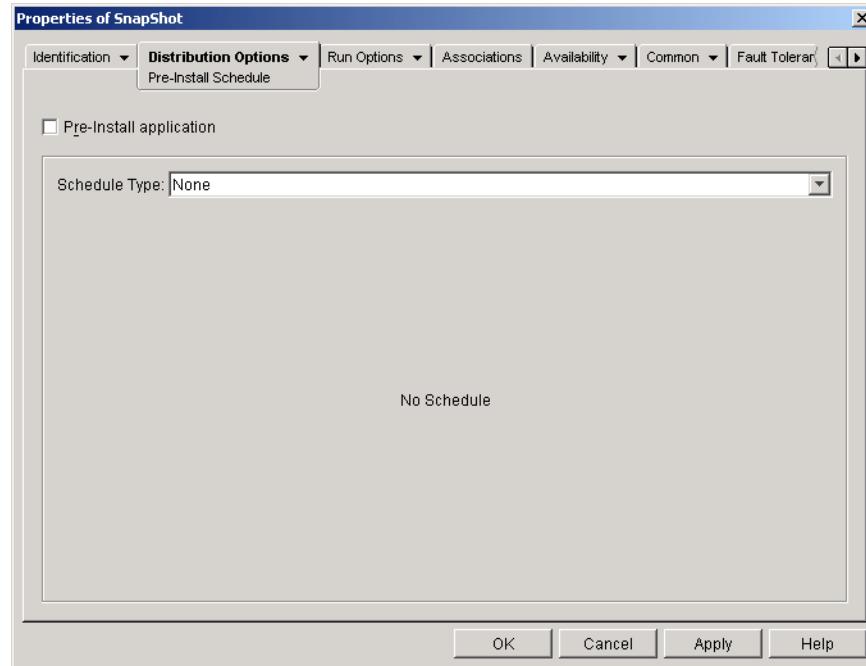
Run After Distribution

Use this text window to enter any script commands you want executed after the application is distributed. The script is executed in the order shown below:

1. Run Before Launching script executed (Run Options > Launch Scripts page)
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 executed (Run Options > Launch Scripts page)

Pre-Install Schedule Page (.AOT/.AXT Application Object Only)

The Pre-Install Schedule property page, shown below, enables Application Launcher/Explorer to install the application to a workstation before the user launches the Application object the first time. Because you can specify when the installation will take place, this enables an off-line, or “lights-out,” distribution of the application and saves the user the wait associated with installing the application. For example, you could pre-install the application after work hours so the application is ready to use when the user comes to work.



With a pre-install, Application Launcher/Explorer performs all workstation-related distribution processes (file copying, modifying text files, .INI files, and workstation registry settings). Then, when the user double-clicks the Application object, Application Launcher/Explorer finishes the installation by performing any user-specific distribution processes (modifying user registry keys, etc.).

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/Explorer must be running. For workstation-associated applications, a user does not need to be logged in.

IMPORTANT: On Windows NT/2000/XP workstations, if a user is not logged in, the user-specific macros will 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 (i.e. %*PROGRAMS% to %*COMMONPROGRAMS%).

Pre-Install Application

Check this option to enable the application to be pre-installed. If you don't check this option, the application will not be 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 will be pre-installed as soon as the 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 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 will be pre-installed. To add a date, click Add > select the date you want > click OK to display it in the list.
- **Times for Selected Dates:** Select the availability start time and end time. The times apply to all dates in the Date Range list.
- **Spread from Start Time (in Minutes):** Enter a number of minutes between available times if you don't want multiple workstations installing the application at once for fear of overloading the network. The spread option spreads out access times over the number of minutes specified so that all pre-installations don't occur at the same time.

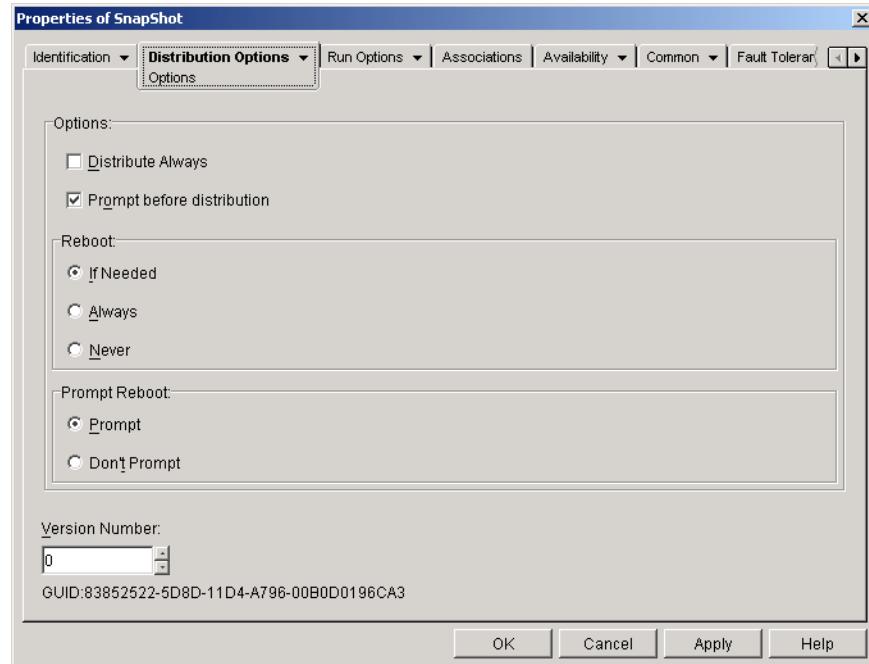
Range of Days

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 is depressed.
- **Times for Selected Range:** Select the availability start time and end time. This option works differently depending on whether you selected one day or multiple days. If you selected one day, the application will be available between the start and end times on that day. For example, if you make the application available on Monday between 8:00 and 5:00, it will run between 8:00 and 5:00. However, if you make the application available Monday through Saturday between 8:00 and 5:00, the times are ignored. The application will be available Monday through Saturday, 24 hours a day.
- **Spread from Start Time (in Minutes):** Enter a number of minutes between available times if you don't want multiple workstations installing the application at once for fear of overloading the network. The spread option spreads out access times over the number of minutes specified so that all pre-installations don't occur at the same time.
- **Use the 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 11:00 p.m. Los Angeles time).

Options Page

The Options property page, shown below, determines general options to be used by Application Launcher/Explorer when distributing the application to a workstation.



Options

These options let you determine if the application should always be distributed and if the user should be prompted to accept or reject the distribution.

Distribute Always

By default, Application Launcher/Explorer will make 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/Explorer to redistribute the application each time it is launched, check 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 NALCACHE directory on his or her local machine, the files and settings are distributed from the NALCACHE directory. If the user does not have a NALCACHE directory (for example, the user is running Application Launcher/Explorer through a terminal server client session) or if writing to the cache has been disabled for the user (User object > Application Launcher tab > Launcher Config tab > Enable Writing to Cache option), the application files and settings are updated from Novell eDirectory. To force a distribution from eDirectory even if the user has a NALCACHE 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 Settings, INI Settings, Application Files, Icons/Shortcuts, and Text Files pages (Distribution Options tab).

Prompt Before Distribution

Check this option to prompt users whether or not they want the application distributed to their workstation. 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).

Reboot

Select how a workstation reboot should occur. The available options are:

- ♦ **If Needed:** Application Launcher/Explorer reboots the workstation if changes need to be made that cannot occur while Windows is running (such as replacing open DLLs).
- ♦ **Always:** Application Launcher/Explorer always reboots the workstation after distributing the application.
- ♦ **Never:** Application Launcher/Explorer does not reboot the workstation. The changes take effect the next time the workstation reboots.

Prompt Reboot

Select whether or not the user is prompted to reboot the workstation.

Version Number

The version number is a unique number between 0 and 65535 that you can increment as you make revisions to the Application object. If you increment the version number, the application is distributed again the next time the user launches the Application object.

GUID

The GUID (global unique identifier) is a randomly generated string that provides a unique identifier for the application. When Application Launcher/Explorer distributes an application to a workstation, it adds the GUID to the workstation's Windows registry for tracking purposes.

You can use the GUID and version number to track and troubleshoot distributed applications. For example, if you want to ensure that a particular application or version has been distributed to a workstation, you can compare the GUID and version number as recorded in the Application object's Distribution page with the GUIDs that are currently included in the workstation's registry. Using the Synchronize Distributed GUIDs option (Tools > Application Launcher Tools menu), you can make several Application objects share the same GUID. This is useful if you are distributing a suite of applications. Using the Generate New GUIDs option (Tools > Application Launcher Tools menu), you can also ensure that two or more applications have different GUIDs.

Run Options Tab

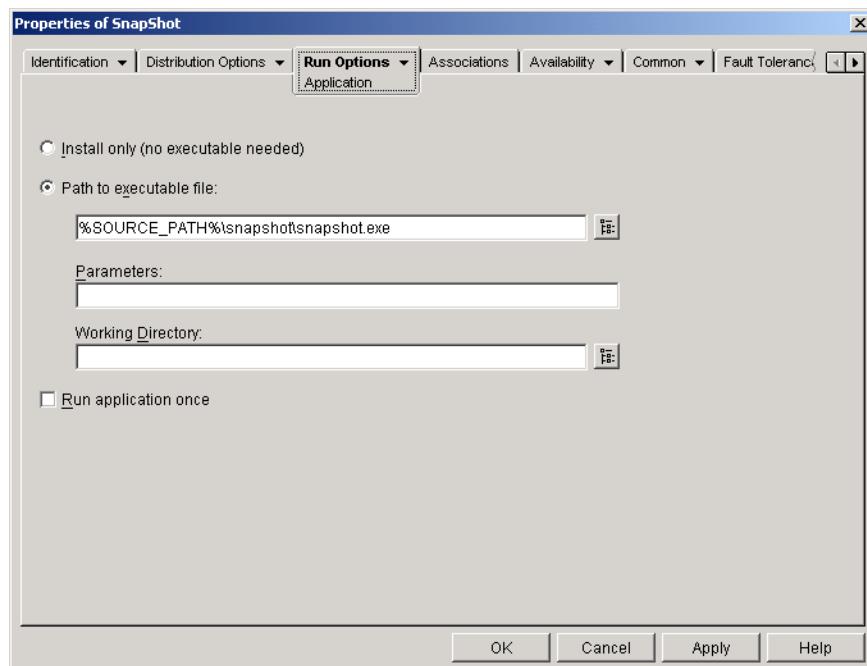
The Run Options tab includes the following pages to help you configure how the Application object runs on the workstation:

- ♦ “[Application Page](#)” on page 141

- ◆ “Environment Page” on page 143
- ◆ “Launch Scripts Page” on page 145
- ◆ “Environment Variables Page” on page 146
- ◆ “License/Metering Page” on page 147

Application Page

The Application property page, shown below, determines general options to be used by Application Launcher/Explorer when running the application on a workstation.



Install Only (No Executable Needed)

Check this option if the Application object does not include an application executable that Application Launcher/Explorer should run after distributing the Application object. 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 Executable File

Enter, or browse for, the path to the executable that Application Launcher/Explorer will run 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 will be located on the user's workstation after distribution.

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 15, “Macros,” on page 189](#).

Parameters

Specify any command line parameters that need to be passed to the executable specified in the Path to Executable File field. See your Windows documentation for more information. The field limit is 250 characters.

You can also use macros in this field. For information about macros, see [Chapter 15, “Macros,” on page 189](#).

Working Directory

Specify the working directory of the executable you specified in the Path to Executable 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 15, “Macros,” on page 189](#).

Run Application Once

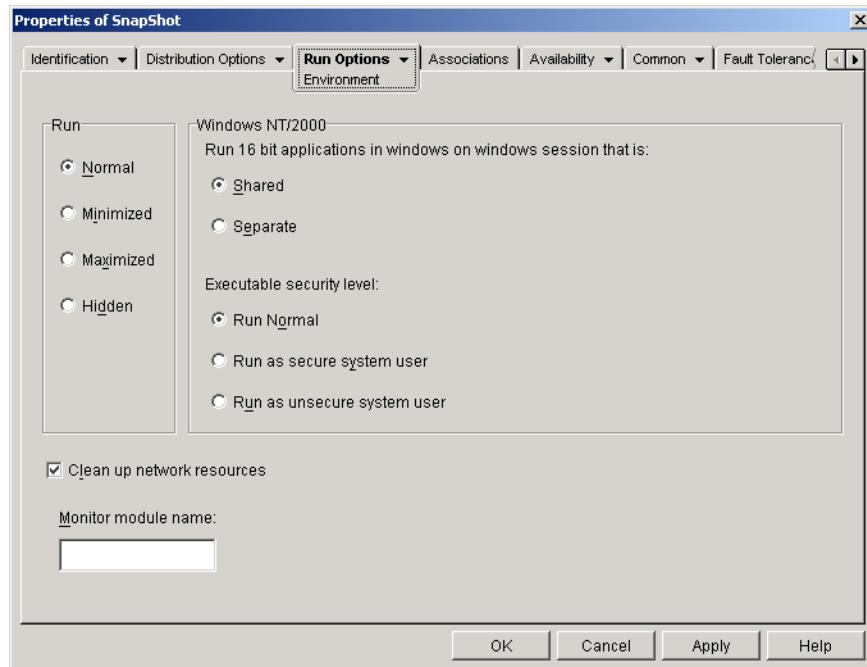
Check this option to have Application Launcher/Explorer run the application one time only and then remove the Application object's icon from the workstation. If the Application object's version number (Distribution Options > Options page) changes, the Application object icon will appear again so that 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 will run when it becomes available to Application Launcher/Explorer), the application is forced to run only one time.

Environment Page

The Environment property page, shown below, specifies the workstation environment that Application Launcher/Explorer will create before running the application on the workstation.



Run

Select the application's initial window size: Normal, Minimized, Maximized, or Hidden. In Hidden mode, the application will run 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.

Windows NT/2000/XP

Use these options to specify how applications run on Windows NT/2000/XP workstations.

Run 16-Bit Applications in Windows on Windows Session

If you are setting up a 16-bit application to run on Windows NT/2000/XP, specify either a shared or separate Windows on Windows (WOW) session. To run on Windows NT/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

Select one of the following security levels for the application:

- ♦ **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:** Running under the System user, the application inherits the workstation's credentials and is run as a Windows NT/2000/XP process. 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 distributing 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 will have the ability to do whatever that application offers on the workstation. For example, if you specify REGEDIT as an unsecure system user, the workstation's user will be able to edit any registry values.

On a Windows 2000 Advanced Server functioning as a Windows terminal server, applications that run as Secure System User or Unsecure System User are displayed on the server's console and not in the user's terminal session. This occurs because the Application Launcher/Explorer Service for Windows NT/2000/XP, which launches and runs the application, is running as a console user, not as a terminal server user.

Clean Up Network Resources

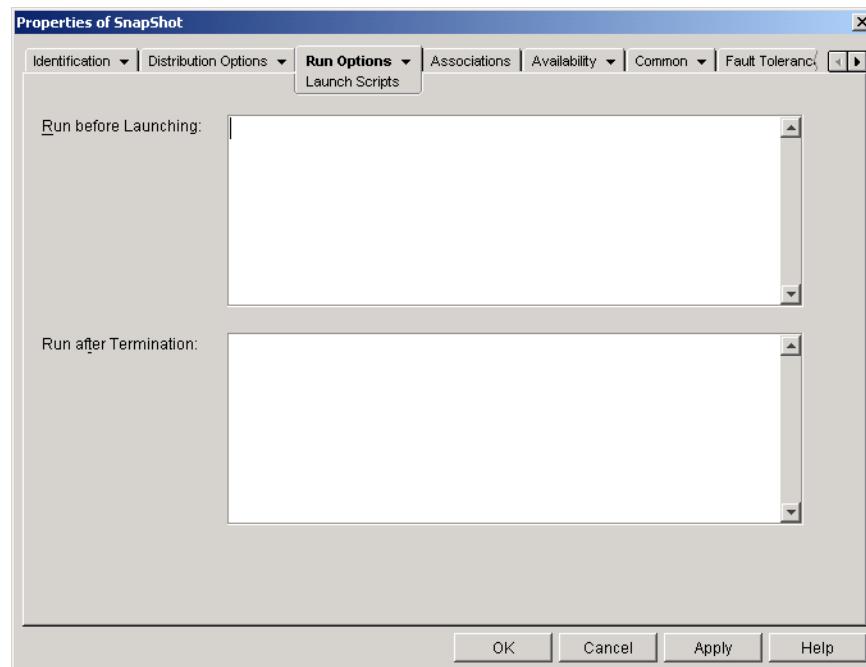
Check this option to remove network connections, drive mappings, and port captures that were established for running the application. If this option is not selected, the network resources remain in effect. If the resource is already in use when Application Launcher/Explorer starts the application, it is not cleaned up until all other distributed applications are finished using it. If distributed applications are still running when Application Launcher/Explorer is terminated, the allocated resources remain intact.

Monitor Module Name

When an application is launched, Application Launcher/Explorer 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/Explorer 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.

Launch Scripts Page

The Launch Scripts property page, shown below, defines the scripts Application Launcher/Explorer will execute when running the application.



The scripts support the same commands and syntax as the Novell Client, with the exception of the ones listed below. 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>).

Application Launcher/Explorer will not output anything to the screen, display errors, or support the following commands:

CLS
DISPLAY
EXIT
FDISPLAY
INCLUDE
LASTLOGINTIME
NO_DEFAULT
NOSWAP
PAUSE
PCOMPATIBLE
SCRIPT_SERVER
SET_TIME
SWAP
WRITE

Run Before Launching

Use this text window to enter any script commands you want Application Launcher/Explorer to execute before launching the application. The script is executed in the order shown below:

1. Run Before Launching script executed
2. Run Before Distribution script executed (Distribution Options > Distribution Scripts page)
3. Application distributed (files copied, settings modified, etc.)
4. Run After Distribution script executed (Distribution Options > Distribution Scripts page)
5. Application launched
6. Application closed (by user)
7. Run After Termination executed

Run After Termination

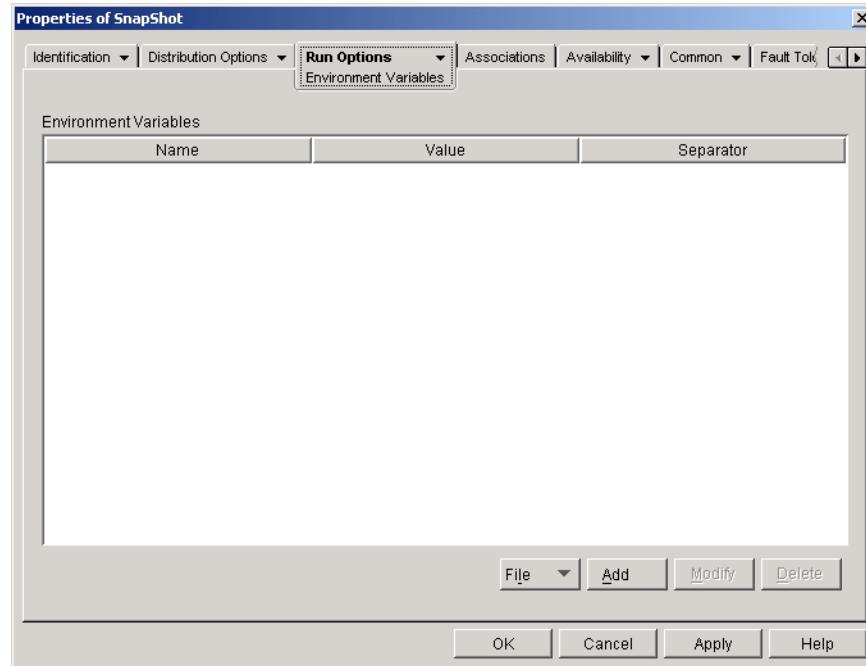
Use this text window to enter any script commands you want Application Launcher/Explorer to execute after the user closes the application. The script is executed in the order shown below:

1. Run Before Launching script executed
2. Run Before Distribution script executed (Distribution Options > Distribution Scripts page)
3. Application distributed (files copied, settings modified, etc.)
4. Run After Distribution script executed (Distribution Options > Distribution Scripts page)
5. Application launched
6. Application closed (by user)
7. Run After Termination script executed

IMPORTANT: 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 will 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.

Environment Variables Page

The Environment Variables property page, shown below, specifies the environment variables that Application Launcher/Explorer will set before running the application. For example, you can specify the application's working directory in the PATH environment variable.



Environment Variables

The Environment Variables list displays all variables that will be modified before the application is run. Application Launcher/Explorer modifies only the variables in this list.

File

Click File > Find to search for variables or variable data in the Environment Variables list.

Click File > Import to import variables from another Application object's .AOT or .AXT file.

Add

To add a variable to the Environment Variables list, click Add to display the Edit Environment Variables dialog box > enter the variable name and data > check 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 > click OK.

Modify

In the Environment Variables list, select the variable you want to modify > click Modify.

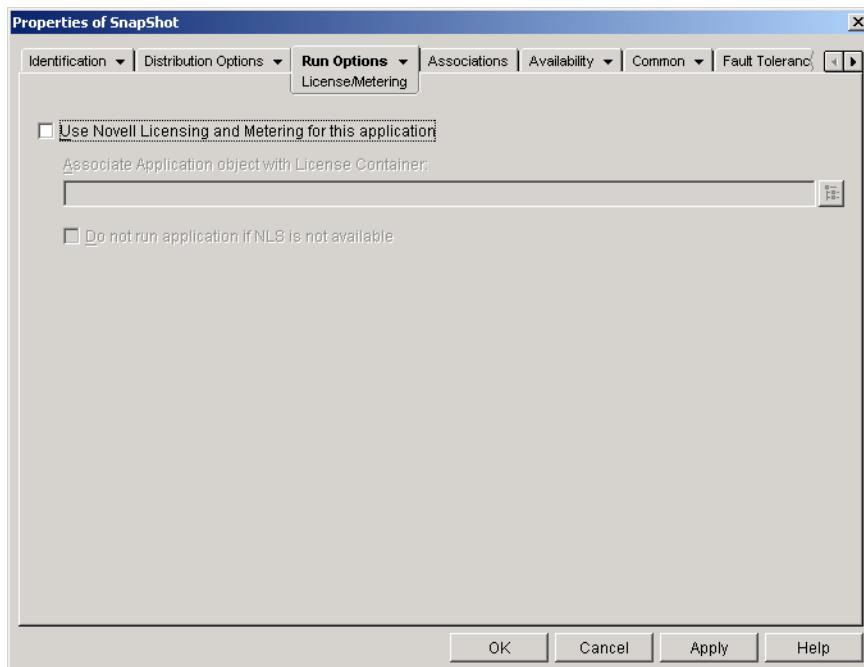
Delete

In the Environment Variables list, select the variable you want to delete > click Delete to remove it from the list.

License/Metering Page

The License/Metering property page, shown below, configures Application Launcher/Explorer to use Novell Licensing Services (NLS) to track application usage and comply with the application's licensing agreement. NLS must already be installed, and you must have created a License

Container and a Metered Certificate for the application. For detailed information about setting up license metering, see [Chapter 12, “Metering Software Licenses,” on page 109](#).



Use Novell Licensing and Metering for this Application

Check this option to enable metering of the application.

Associate Application Object with License Container

Once 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

Check this option to prevent users from running the application if NLS is not running.

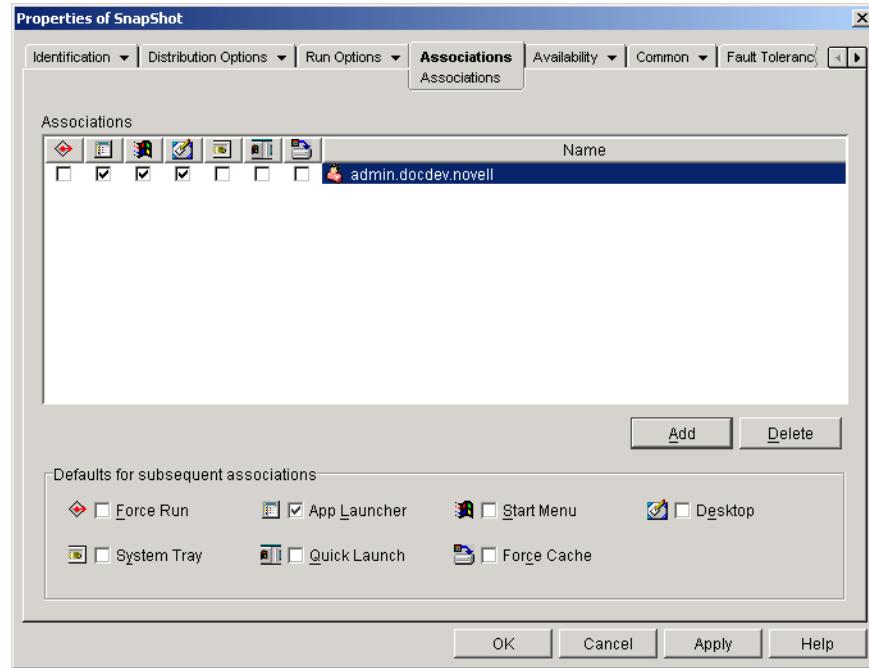
Associations Tab

The Associations tab includes one page only. This page lets you associate the Application object with users and workstations.

- ◆ [“Associations Page” on page 148](#)

Associations Page

The Associations property page, shown below, lists the objects that have been associated with the Application object. Objects that are associated with the Application object can see and use the application. You can grant rights to User, Workstation, Group, Workstation Group, Organizational Unit, Organization, and Country objects.



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 > 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 checking or unchecking the appropriate boxes.

Delete

To disassociate an object from the Application object, select the object in the Associations list > 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. If you change the characteristics that are selected, any objects you add after that will have the new characteristics.

Force Run

Automatically runs the application. With a user-associated application, the application is run immediately after Application Launcher/Explorer 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 will be

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 associating an Application object with a workstation, Force Run will not work if the Application object uses a Microsoft Windows Installer (.MSI) package. This is true only when the Application object is associated with a workstation. Force Run works with .MSI Application objects associated with users.

App Launcher

Adds the Application object's icon to the Application Launcher/Explorer window.

Start Menu

If the workstation is running Application Explorer, this option adds the Application object to the Windows Start menu. The Application object will be 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 toolbar.

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 NDS. The files are compressed to save space on the workstation's local drive. For more information about caching applications, see [Chapter 8, “Running in Disconnected Mode,” on page 73](#).

IMPORTANT: This option is required only if you want to ensure that the user can install or repair the application while disconnected from NDS. Without this option selected, the user will still be able to launch the application in disconnected mode, provided the application has already been distributed (installed) to the workstation.

Availability Tab

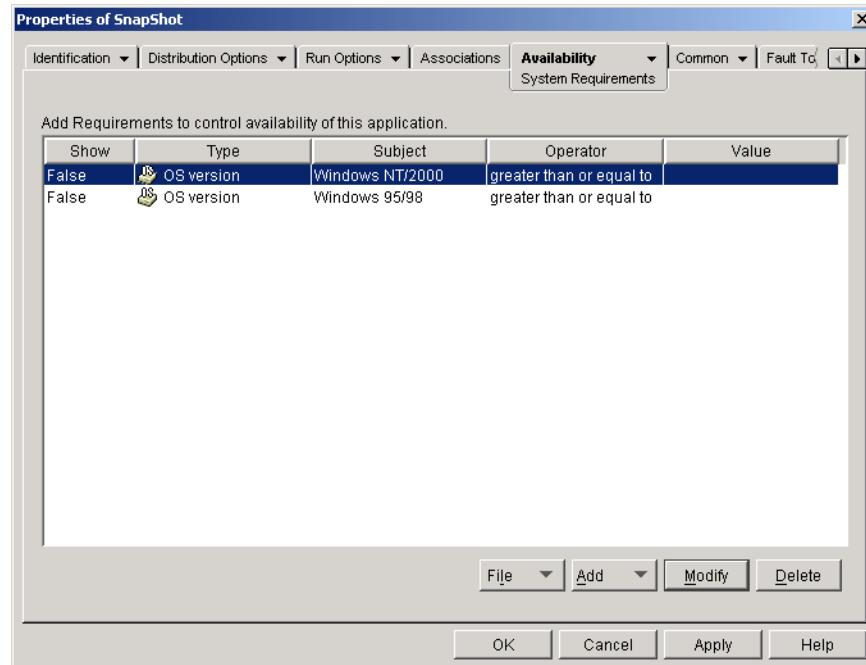
The Availability tab includes the following pages to help you determine the system requirements a workstation must meet before the application will be distributed to the workstation, the days and times the application will be available to users, and the messages and warnings Application

Launcher/Explorer will display to users when terminating an application that has become unavailable to users:

- ◆ “[System Requirements Page](#)” on page 151
- ◆ “[Schedule Page](#)” on page 162
- ◆ “[Termination Page](#)” on page 164

System Requirements Page

The System Requirements property page, shown below, determines the system requirements a workstation must meet before Application Launcher/Explorer will distribute the application to it.



Add Requirements to Control Availability of This Application

This list displays all the requirements associated with the distribution of the application. For each requirement, the following information is displayed:

- ◆ **Show:** This column indicates if the Application object's icon will be displayed on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met.
If an icon is displayed in a disabled state, users can right-click the icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.
- ◆ **Type:** This column shows the category of the system requirement. Categories are explained under Add below.
- ◆ **Subject:** This column shows the subject of the system requirement. For example, if the type is OS Version, the subject could be Windows 95/98 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 version may need to be greater than or equal to 4.00.950 in order for the application to be distributed.

File

Click File > Find to search for requirements that include certain information. For example, you could search for requirements that include "Windows," "greater than," or "system memory."

Click File > Import to import requirements from another Application object.

Add

Click this option to add a new system requirement, based on one of the following categories:

- ♦ **Applications:** Require a specific application to be installed or not be installed. The application must have been distributed through an Application object. For more information, see "[Applications](#)" on page 152.
- ♦ **Disk Space:** Require a specific amount of disk space on the workstation. For more information, see "[Disk Space](#)" on page 153.
- ♦ **Environment Variables:** Require specific environment variables to be present or not present. For more information, see "[Environment Variables](#)" on page 155.
- ♦ **Memory:** Require a specific amount of memory on the workstation. For more information, see "[Memory](#)" on page 155.
- ♦ **Operating System:** Require a specific operating system. For more information, see "[Operating System](#)" on page 156.
- ♦ **Processor:** Require a specific processor type. For more information, see "[Processor](#)" on page 157.
- ♦ **Registry:** Require specific registry values to be present or not present. For more information, see "[Registry](#)" on page 158.
- ♦ **File Existence:** Require a file to be present or not present. For more information, see "[File Existence](#)" on page 159.
- ♦ **File Version:** Require a file to be a specific version. For more information, see "[File Version](#)" on page 160.
- ♦ **File Date:** Require a file to be a specific date. For more information, see "[File Date](#)" on page 160.
- ♦ **File Size:** Require a file to be a specific size. For more information, see "[File Size](#)" on page 161.

Modify

Select a requirement in the list > click Modify to edit the information.

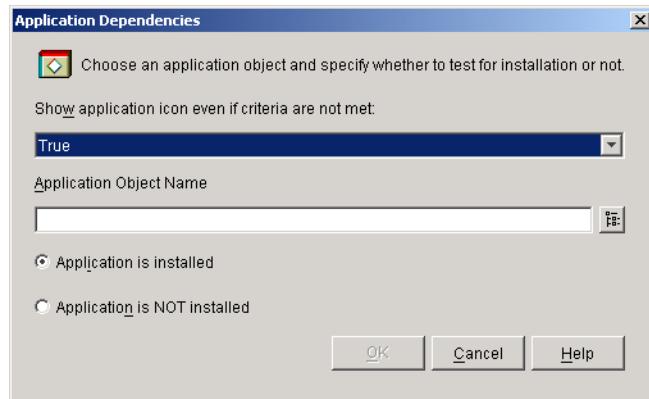
Delete

Select a requirement in the list > click Delete to remove the requirement.

Applications

The Application Dependencies dialog box (System Requirements page > Add > Application) lets you create a requirement based on the existence, or nonexistence, of another application.

Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

Application Object Name

Browse and select the appropriate Application object.

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.

Disk Space

The Disk Requirements dialog box (System Requirements page > Add > Disk Space) lets you create a requirement based on the amount of disk space required by the application. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

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, WINDOWS\SYSTEM) resides.
- **Windows Directory Drive:** Allows you to select the disk where the Windows directory (for example, WINDOWS) resides.
- **Windows Temp Directory Drive:** Allows you to select the disk where the Windows temporary directory (for example, WINDOWS\TEMP) resides.

If you create multiple disk space requirements for the application, Application Launcher/Explorer 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 drive (which is also the C: drive), and 50 MB free on the D: drive, Application Launcher/Explorer 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)

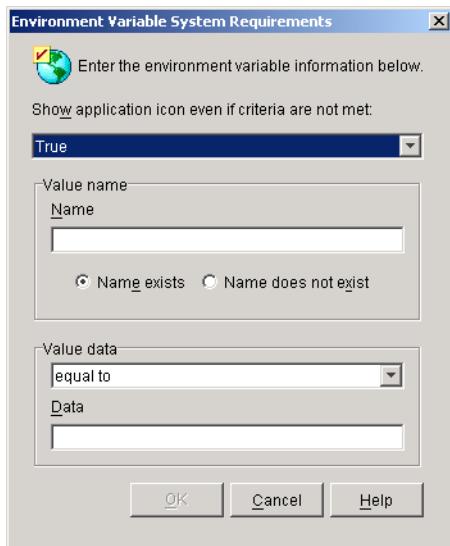
Select whether the available disk space must be less than, less than or equal to, equal to, greater than, greater than or equal to the amount listed in the MB field. For example, if you want to require at least 50 MB of free disk space, select Greater Than or Equal To and then enter 50 in the MB field.

MB (Condition)

Enter the disk space requirement.

Environment Variables

The Environment Variable System Requirements dialog box (System Requirements page > Add > Environment Variables) lets you create a requirement based on the existence, or nonexistence, of a specific environment variable. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

Value Name

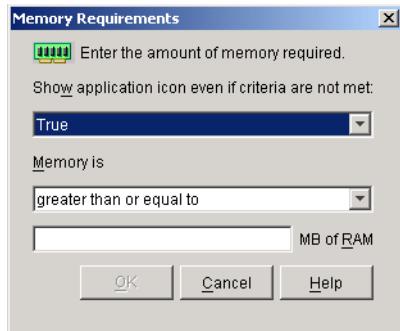
Enter the name of the environment variable. Select the Name Exists option to require the variable to already exist on the workstation, or select the Name Does Not Exist option to require the variable to not exist on the workstation.

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.

Memory

The Memory Requirements dialog box (System Requirements page > Add > Memory) lets you create a requirement based on the amount of memory (RAM) required to run the application. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

Memory Is (Conditional Statement)

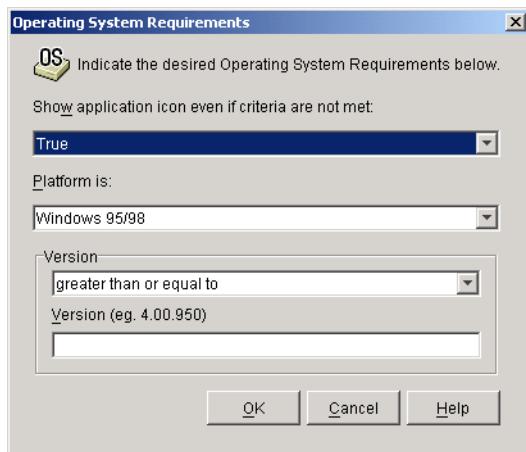
Select whether the total memory must be less than, less than or equal to, equal to, greater than, greater than or equal to the memory entered in the MB of RAM field. For example, if you want to require at least 32 MB of free RAM, select Greater Than or Equal To and then enter 32 in the MB of RAM field.

MB of RAM (Condition)

Enter the required memory amount.

Operating System

The Operating System Requirements dialog box (System Requirements page > Add > Operating System) lets you create a requirement based on the operating system required to run the application. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

IMPORTANT: With this option set to True, the application icon will be shown only if the operating system platform is the same as the one selected but the version does not meet the version requirement. For example, if you select Windows95/98 as the platform and enter greater than or equal to 4.02.1998 as the version, the icon will not display on a Windows NT/2000/XP workstation, will display in an active state on a Windows 98 workstation running version 4.02.1998 or newer, and will display in a disabled state on a Windows 95 workstation or a Windows 98 workstation running version 4.01.1998 or older.

Platform Is

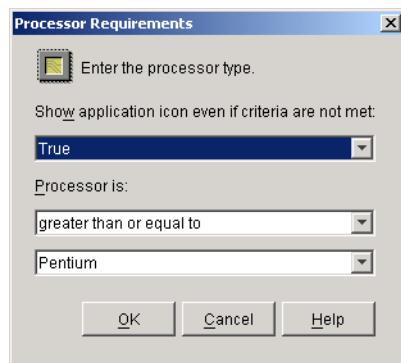
Select the operating system (Windows95/98 or Windows NT/2000/XP) the workstation must be running.

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. For example, if you want to require at least version 4.00.95, select Greater Than or Equal To and then enter 4.00.95 in the Version field.

Processor

The Processor Requirements dialog box lets you create a requirement based on the processor required to run the application. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

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. 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 Requirements dialog box lets you create a requirement based on a specific registry key or value. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

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

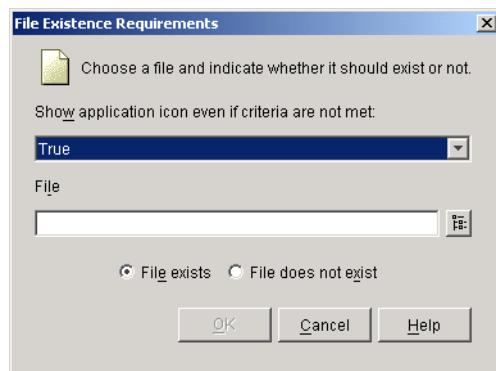
If the key has a specific value you want to check for, browse and select the value. Select Value Exists if the value must exist for the distribution to take place, or select Value Does Not Exist if the value cannot exist for the distribution to take place.

Value

If you specified a value in the Name field, select a condition (less than, less than or equal to, equal to, not equal to, greater than or equal to, greater than) for the value's data > enter the data in the Value field > select whether the value data is a string or DWORD type.

File Existence

The File Existence Requirements dialog box lets you create a requirement based on the existence, or nonexistence, of a specific file. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

File

Browse to select the file you want to search for.

File Exists

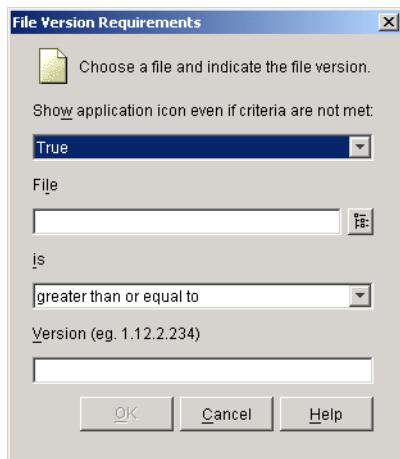
Select this option to require the file to exist on the workstation.

File Does Not Exist

Select this option to require the file to not exist on the workstation.

File Version

The File Version Requirements dialog box lets you create a requirement based on the version of a specific file. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

File

Browse to select the file whose version you want to check.

Is (Conditional Statement)

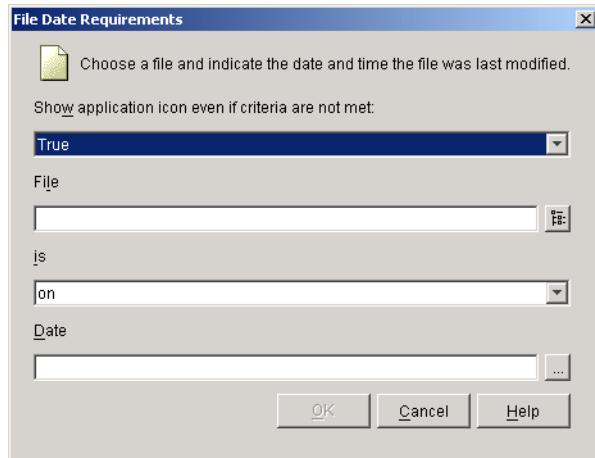
Select whether the version must be less than, less than or equal to, equal to, greater than, greater than or equal to the version entered in the Version field. For example, if you want to require at least version 1.12.2.234, select Greater Than or Equal To and then enter 1.12.2.234 in the Version field.

Version

Enter the file version.

File Date

The File Date Requirements dialog box lets you create a requirement based on the date of a specific file. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

File

Browse to select the file whose date you want to check.

Is (Conditional Statement)

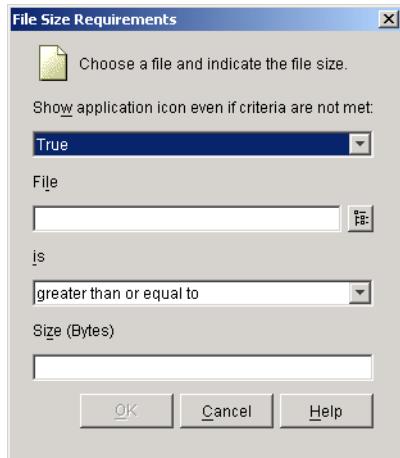
Select whether the file date must be before, on or before, on, on or after, or after the date entered in the Date field. For example, if you want to make sure the application is distributed only if the file is dated 15 January 2000 or older, you would select On or Before and then select the date.

Date

Use the Calendar to select the appropriate date and time.

File Size

The File Size Requirements dialog box lets you create a requirement based on the size of a specific file. Application Launcher/Explorer will not distribute the application to the workstation unless the requirement is met.



Show Application Icon Even If Criteria Are Not Met

Select True to display the application icon on the workstation regardless of whether or not the requirement is met. The icon will display in a disabled state if the requirement is not met. Users can right-click the disabled application's icon, select Details, then select Requirements in order to see which requirements were not met. They can then perform any necessary upgrades to the workstation configuration to meet the application requirements.

File

Browse to select the file whose size you want to check.

Is (Conditional Statement)

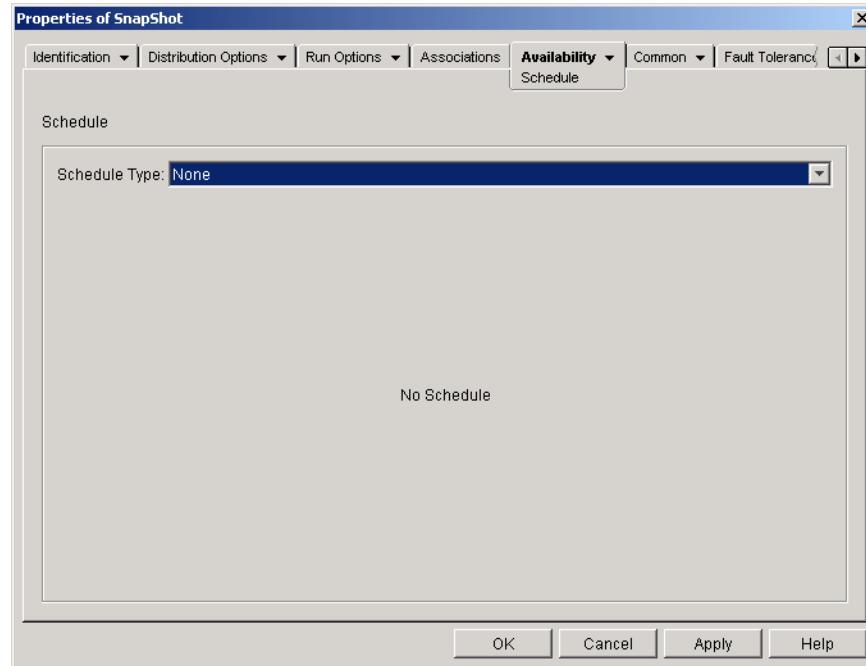
Select whether the size must be less than, less than or equal to, equal to, greater than, or greater than or equal to the size entered in the Size field. For example, if you want to require the file to be at least 200 KB, select Greater Than or Equal To and then enter 200 in the Size field.

Size

Enter the file size in kilobytes.

Schedule Page

The Schedule property page, shown below, defines the dates and times when Application Launcher/Explorer will make the Application object available to users.



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. You cannot select more than 350 specific dates.

Date Range: The Date Range list displays all dates when the Application object will be available. To add a date, click Add > select the date you want > click OK to display it in the list.

Times for Selected Dates: Select the availability start time and end time. The times apply to all dates in the Date Range list.

Spread from Start Time (in Minutes): Enter a number of minutes between available times if you don't want all users to run the application at the same time for fear of overloading the network. The spread option spreads out user access times over the number of minutes specified so they don't all run the application at once.

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.

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 will be available. By default, all days are selected; a day is selected when the button is depressed.

Times for Selected Range: Select the availability start time and end time. This option works differently depending on whether you selected one day or multiple days. If you selected one day, the application will be available between the start and end times on that day. For example, if you make the application available on Monday between 8:00 and 5:00, it will run between 8:00 and 5:00. However, if you make the application available Monday through Saturday between 8:00 and 5:00, the times are ignored. The application will be available Monday through Saturday, 24 hours a day.

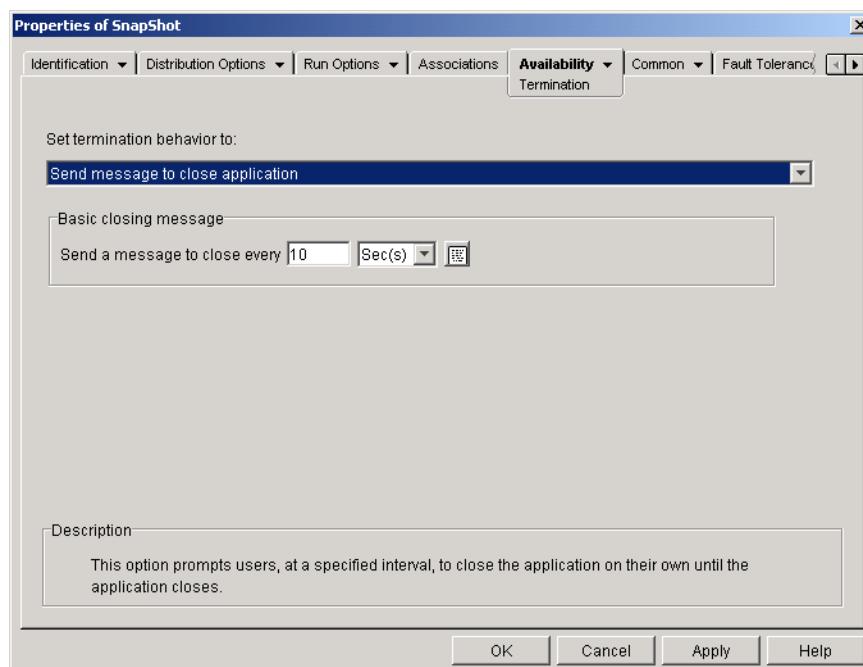
Spread from Start Time (in Minutes): Enter a number of minutes between available times if you don't want all users to run the application at the same time for fear of bringing down the network because of the load and traffic.

Use the 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 11:00 p.m. Los Angeles time).

Termination Page

The Termination property page, shown below, determines how Application Launcher/Explorer 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 system requirements change and the user's workstation no longer complies.



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/Explorer 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/Explorer to display the message. The following default message will be 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/Explorer to first display a message instructing the user to close the application. If the user does not close the application, Application Launcher/Explorer 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.

Warning Message (Optional): Check the Send Warning option if you want to warn the user to close the application, then enter the number of times to display the warning message and the interval between messages. The following default message will be 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, enter 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/Explorer attempts to close the application. The following default message will be 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 to Close

This option causes Application Launcher/Explorer to first display a message instructing the user to close the application. If the user does not close the application, Application Launcher/Explorer 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.

Warning Message (Optional): Check the Send Warning option if you want to warn the user to close the application, then enter the number of times to display the warning and the interval between warnings. The following default message will be 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, enter 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/Explorer attempts to close the application. The following default message will be 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/Explorer 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/Explorer closes the application automatically and displays a termination message.

Warning Message (Optional): Check the Send Warning option if you want to warn the user to close the application, then enter the number of times to display the warning and the interval between warnings. If the user does not close the application, Application Launcher/Explorer closes the application automatically after the last warning message. The following default message will be 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/Explorer 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.

Common Tab

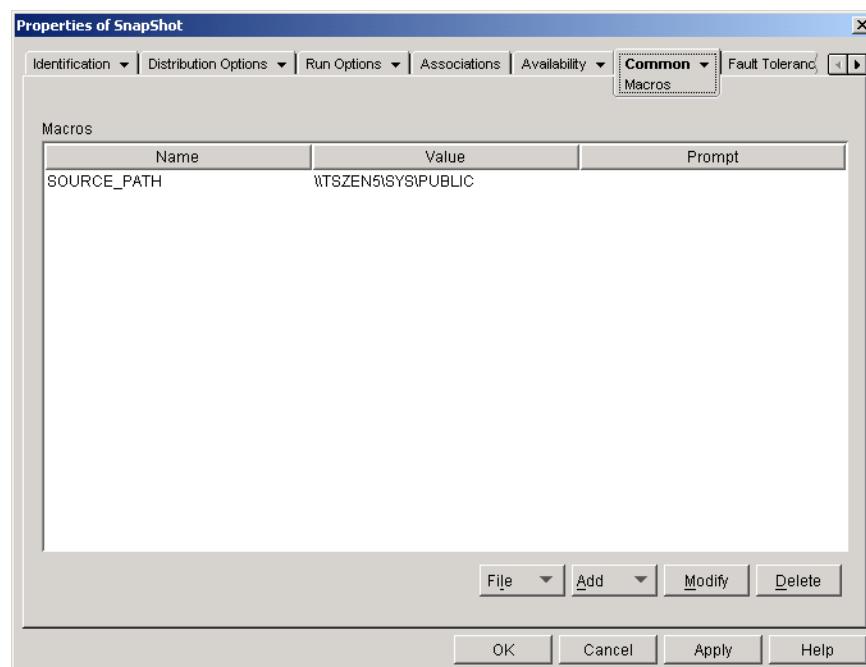
The Common tab includes the following pages:

- ◆ “[Macros Page](#)” on page 167
- ◆ “[Drives/Ports Page](#)” on page 169

- ◆ “File Rights Page” on page 172
- ◆ “Reporting Page” on page 174
- ◆ “Properties Page (.MSI Application Object Only)” on page 175
- ◆ “Transforms Page (.MSI Application Object Only)” on page 176
- ◆ “Imaging Page” on page 177
- ◆ “Sources Page” on page 179
- ◆ “Uninstall Page” on page 181

Macros Page

The Macros property page, shown below, displays the macros that have been defined for use with the Application object.



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 will be 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.

You can use Application object macros when defining the following Application object properties:

- ◆ Path to Executable (Run Options > Applications page)
- ◆ Command Line (Run Options > Applications page)
- ◆ Working Directory (Run Options > Applications page)
- ◆ Drive Mapping Path (Common > Drives/Ports page)

- ◆ Capture Port Path (Common > Drives/Ports page)
- ◆ Key, Value Name, and Value Data strings (Distribution Options > Registry Settings page)
- ◆ Section, Value Name, and Value Data strings (Distribution Options > INI Settings page)
- ◆ Source File, Target File, Directory Name (Distribution Options > Application Files page)
- ◆ Find File, Find Next File, Add string (Distribution Options > Text Files page)
- ◆ All properties (Distribution Options > Icons/Shortcuts page)

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

Click File > Find to search for specific information in the Name, Value, and Prompt fields.

Click File > Import to import macros from another Application object's .AOT or .AXT 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 > 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, NDS attribute macros, environment variable macros, login script macros, and other Application object macros. For information about these types of macros, see [Chapter 15, “Macros,” on page 189](#).

Prompted

A Prompted macro is used to create a macro that prompts the user to accept the default value (either a drive or string) or enter a new value.

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:** Enter 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 will be used if the user does not enter another drive letter.
- **Minimum Disk Space in MB:** Enter 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:** Enter the text you want presented to users.
- **Default Value:** Select the value that will be used if the user does not enter another value in response to the prompt text.
- **Minimum Disk Space in MB:** Enter the minimum amount of free disk space required to install the application.
- **Maximum String Length in Chars:** Enter the maximum number of characters the user can enter as a response to the prompt.

Modify

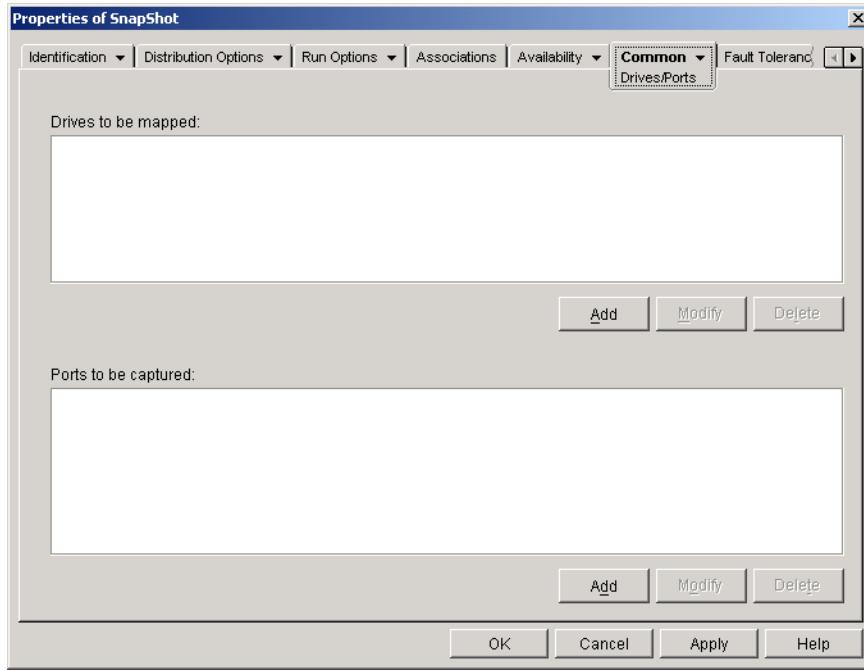
Select the macro whose information you want to modify > click Modify.

Delete

Select the macro you want to delete > click Delete.

Drives/Ports Page

The Drives/Ports property page, shown below, determines the drive mappings and port captures for the application. When a user launches the application (through the Application object), Application Launcher/Explorer establishes the drive mappings and port captures before launching the application.



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/Explorer 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 not released when the user closes the application. For Application Launcher/Explorer to release the drives and ports, you need to select the Clean Up Network Resources option (Run Options > Environment page). With this option selected, Application Launcher/Explorer releases any drives it mapped and ports it captured as long as the drives and ports are not being used by another application. If several applications use the same drive or port, then the drive mappings and port captures aren't cleaned up until the last application terminates.

Drives to Be Mapped

The Drives to Be Mapped list displays all drives that will be mapped before the application is launched.

Add

Click Add to display the Drive to Be Mapped dialog box. Fill in the following fields:

- ◆ **Root:** Check 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
 - ◆ S2: Select this option to assign the drive to the last search drive.

- **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 15, “Macros,” on page 189](#).

- **Map If Drive Mapping Doesn’t Exist or Same as Existing:** Select this option if you want Application Launcher/Explorer 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/Explorer 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 > 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 > click Delete to remove the mapped drive.

Ports to Be Captured

The Ports to Be Captured list displays all ports that will be 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 15, “Macros,” on page 189](#).

- ◆ **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 will be 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 unchecked 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

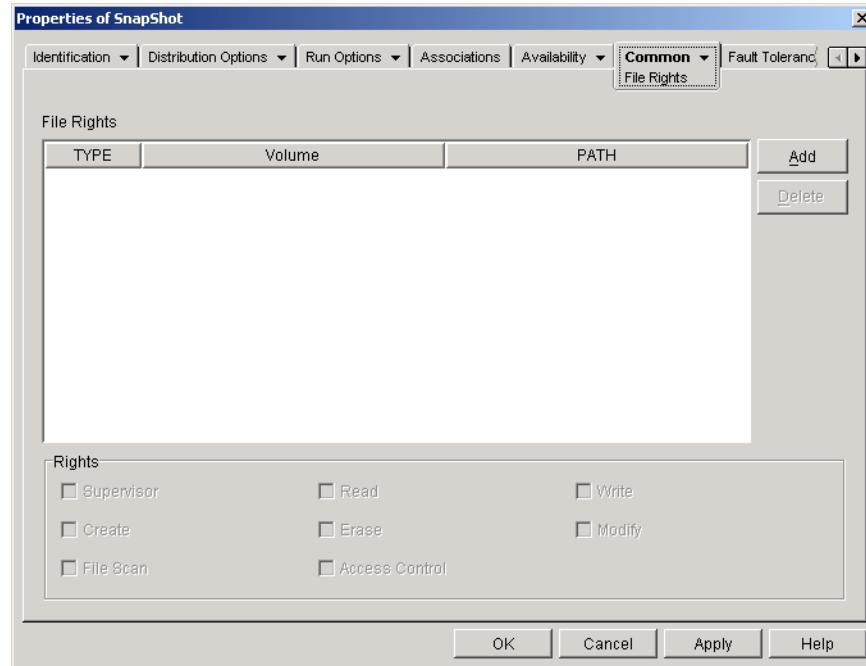
Select a port capture in the Ports to Be Captured list > click Modify to display the Ports to be Captured dialog box. Change the capture information as needed. See the field descriptions above or click the Help button in the dialog box.

Delete

Select a port capture in the Ports to Be Captured list > click Delete to remove the captured port.

File Rights Page

The File Rights property page, shown below, specifies the file, directory, and volume rights a user must have to run the application. A user receives these 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 will be 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 checking or unchecking the appropriate boxes.

Add

Click Add > browse for and select the file, directory, or volume you want to add > click OK to display it in the list.

Delete

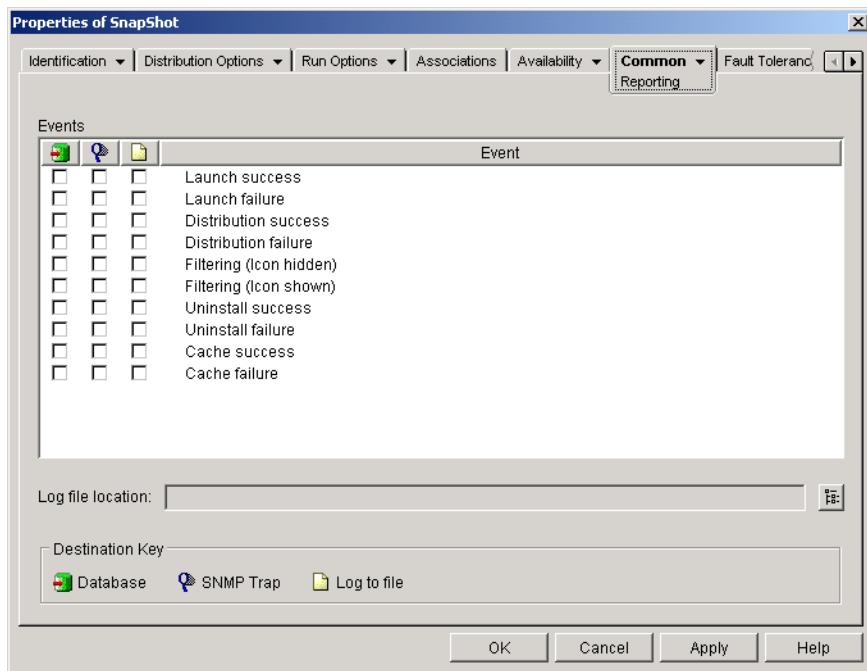
Select the file, directory, or volume you want to delete from the list > click Delete. Any users associated with the Application object will lose rights to the file, directory, or volume.

Rights

Select the file, directory, or volume (in the File Rights list) for which you are establishing users' rights > check the appropriate boxes.

Reporting Page

The Reporting property page, shown below, specifies the application events that you want Application Launcher/Explorer to report on and lets you specify the method of reporting.



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.

For information about setting up reporting, see [Chapter 11, “Reporting on Application Management Events,” on page 95](#).

Events

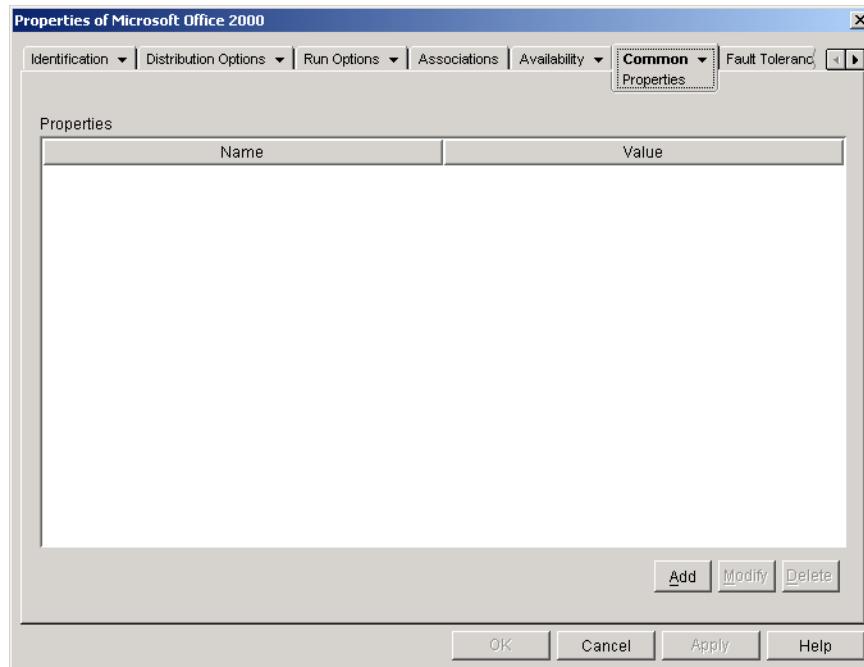
Select the events (distribution, launch, filtering, and uninstallation) that you want reported and the destination where you want the events sent. You can send events to a database, to a management console via an SNMP trap, or to a text log file. If necessary, you can send events to multiple destinations (for example, both a database and a text log file).

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.

Properties Page (.MSI Application Object Only)

The Properties property page, shown below, lets you override the public property values contained in the .MSI package and add new public properties.



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 may want to change one or more of the property values. For example, a property value may 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/Explorer 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 > enter the new value in the Value Data field > 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, enter the new property's name > enter the property's value in the Value Data field > click OK to add the property to the Properties list.

Modify

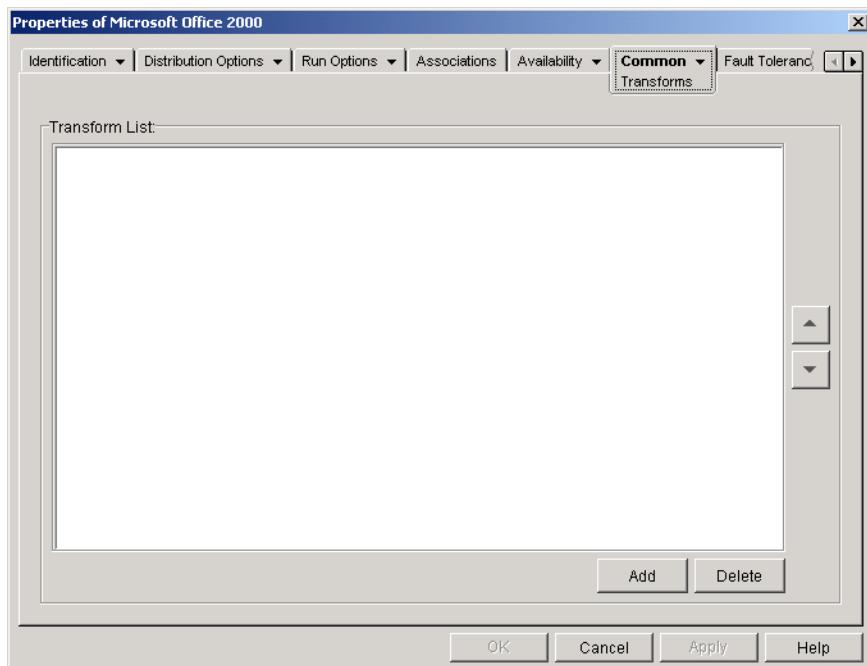
To modify a property that is in the Properties list, select the property > click Modify to display the Edit String Value dialog box > modify the value data > click OK.

Delete

To delete a property from the Properties list, select the property > click Delete. Deleting the property causes future installations of the application to use the property value defined in the .MSI package.

Transforms Page (.MSI Application Object Only)

The Transforms property page lists the transforms that Microsoft Windows Installer will apply 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.



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 > click OK to add it to the list. If the transform file is in the same location as the .MSI package, you can simply enter the filename in the Transform Path field.

Delete

Select a transform from the Transforms List > click Delete to remove it from the list.

Up-Arrow and Down-Arrow

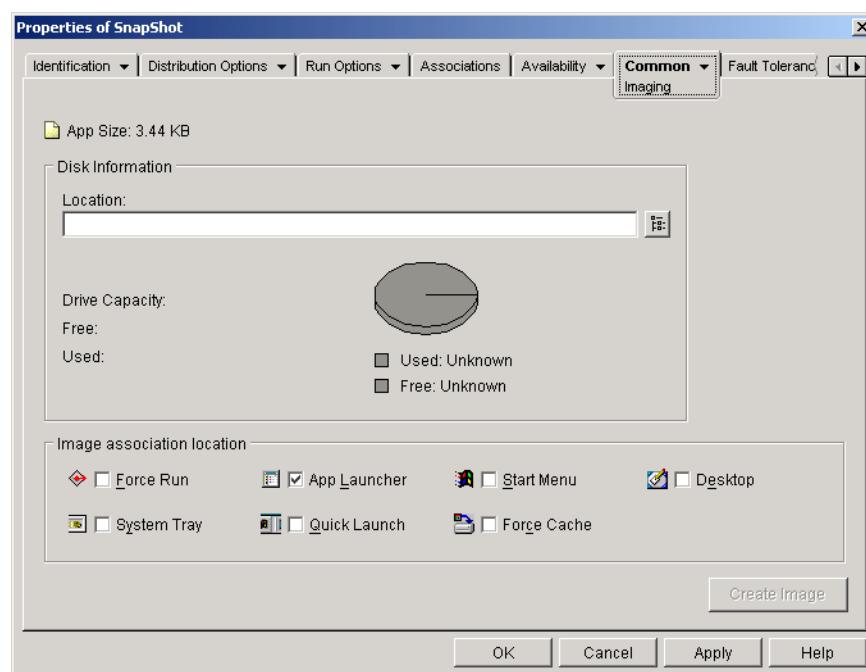
Select a transform from the Transform List > click the up-arrow to move the transform up in the list or click the down-arrow to move it down in the list. Application Launcher/Explorer applies the transforms in the order they are listed, from top to bottom.

Imaging Page

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 so that the application will be delivered to a workstation during imaging of the workstation.

During the imaging process, the add-on image is applied after the base image. Applying the add-on image results in the application files being added to the imaged workstation's NALCACHE directory, located on the workstation's Windows drive (typically C:). When Novell Application Launcher/Explorer starts, it reads the NALCACHE directory and displays the application's icon. After the user double-clicks the icon, the distribution process occurs and then the application is launched.

IMPORTANT: On Windows NT/2000/XP, users who do not have administrative rights to their workstation must be given Full Control access to NALCACHE directory. Otherwise, Application Launcher/Explorer will not be able to read the cache and display the application.



App Size

This field displays the size of the Application object. This includes all Application object information (.AOT or .AXT file) as well as application source files (.FIL files).

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 may want to save it in the same location as the base image file. The filename does not need to have a particular filename extension, although .ZMG is used as the default.

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 will fit in the amount of free disk space.

IMPORTANT: In addition to the image location, the ConsoleOne workstation's NALCACHE directory, located on the workstation's Windows drive (typically C:), 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 will vary; 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.

Create Image

After you've defined the image location and selected the image associations (see “[Image Association Location](#)” on page 178), click the Create Image button to create the image. A dialog box will appear that lets you select from the following compression options:

- **None:** Creates the image file but does not compress it. This will result 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.

Image Association Location

Specify the association characteristics for the Application object. These will be enforced when the image is applied to the workstation.

Force Run

Runs the Application object as soon as Application Launcher/Explorer or Workstation Helper starts and the Application object is available.

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 > Applications page), as soon as the Application object is distributed it will run one time and then be 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).

App Launcher

Adds the Application object's icon to the Application Launcher/Explorer window.

Start Menu

If the workstation is running Application Explorer, this option adds the Application object to the Windows Start menu. The Application object will be 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

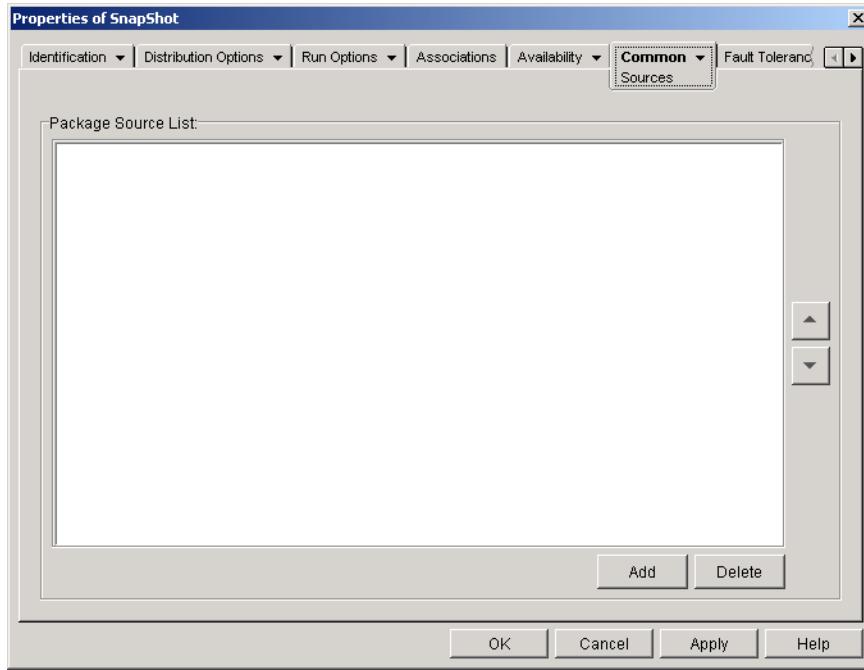
Displays the Application object's icon on the Windows Quick Launch toolbar.

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 NDS. The files are compressed to save space on the workstation's local drive.

Sources Page

The Sources property page, shown below, specifies additional network locations that contain installation packages from which Application Launcher/Explorer can distribute the application. If the Application object was created from a .AOT or .AXT file, the installation packages must be snAppShot packages (.AOT, .AXT, and .FIL files). If the Application object was created from a .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/Explorer begins with the first location in the list and continues down the list until an accessible package is found.



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 a .MSI file rather than a .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 will fail. However, if you create additional source locations and define them in the source list, the on-demand installation will have 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/Explorer cannot access the location, it will use 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/Explorer will use these locations, in the order listed, if the original source location is unavailable.

When distributing applications that use the Windows Installer, Application Launcher/Explorer passes the source list to the Windows Installer, which then uses the list for the locations from which to install the application files.

Add

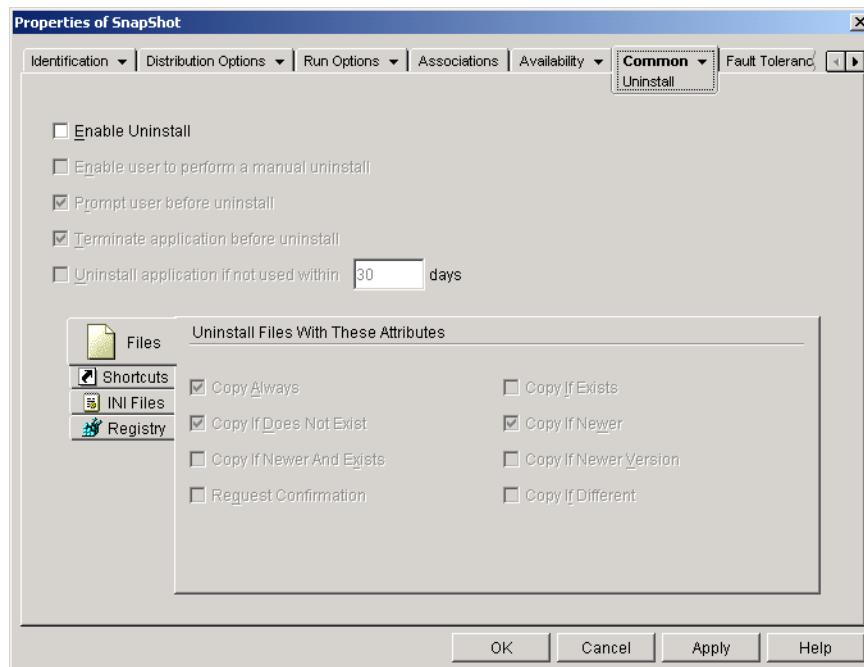
To add a source location to the list, click Add > browse to and select the directory that contains the installation package > click OK.

Delete

To remove a source location from the list, select the location > click Delete.

Uninstall Page

The Uninstall property page, shown below, enables the application to be uninstalled and determines the behaviors associated with uninstalling the application.



Enable Uninstall

Check 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

Check 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

Check this option to prompt users before removing the application from their workstations. If users answer "No" to the prompt, the application will not be removed.

Terminate Application Before Uninstall

Check this option to ensure that Application Launcher/Explorer terminates the application (if the application is running) before it begins uninstalling the application files.

Uninstall Applications If Not Used Within XX Days

Check 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 > check 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). By default, the options are configured (Copy Always, Copy if Does Not Exist, Copy if Newer) to 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.

Shortcuts

Click Shortcuts > check 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 > check 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). By default, the options are configured (Create Always, Create if Exists, Create or Add to Existing Section) to 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 > check 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). By default, the options are configured (Create Always, Create if Exists) to 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/Explorer during distribution of the application to the workstation. The default settings cause Application Launcher/Explorer 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 may 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.

NOTE: 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).

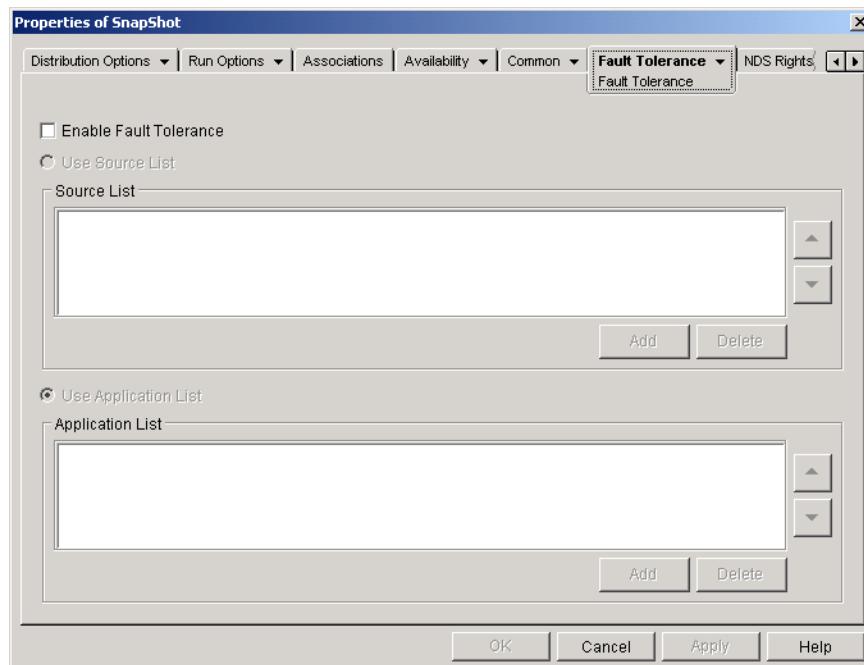
Fault Tolerance Tab

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:

- ◆ “Fault Tolerance Page” on page 183
- ◆ “Load Balancing Page” on page 184
- ◆ “Site List Page” on page 186

Fault Tolerance Page

The Fault Tolerance property page, shown below, specifies the installation package sources or Application objects that Application Launcher/Explorer will use as backups if this Application object becomes unavailable. Application Launcher/Explorer tries the package sources or Application objects in the order shown in the Source List or Application List, from top to bottom.



We recommend you use fault tolerance, rather than load balancing (Fault Tolerance > Load Balancing page) when servers are located at various sites throughout a large wide area network. However, the servers must reside in the same NDS tree. You can implement both methods if necessary. Application Launcher/Explorer will try load balancing first, then fault tolerance.

IMPORTANT: The Fault Tolerance page does not apply to Application objects that use Microsoft Windows Installer (.MSI) packages. For a .MSI Application object, you can provide source resiliency (fault tolerance) through the use of additional source locations for the installation package (Common > Sources page).

Enable Fault Tolerance

Check 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/Explorer use a list of Package sources as the backup. You must have already created at least one installation package source (Common > Sources page).

Add

Click Add > browse to and select a package source > click OK to add it to the Source List.

Delete

Select a package source from the Source List > click Delete to remove it from the Source List.

Up-Arrow and Down-Arrow

Select a package source from the Source List > 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/Explorer tries the sources in the order they are listed, from top to bottom.

Use Application List

Select this option to have Application Launcher/Explorer 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 too busy or is not available, then Application Launcher/Explorer tries each Application object in the specified order.

Add

Click Add > browse to and select an Application object > click OK to add it to the Application List.

Delete

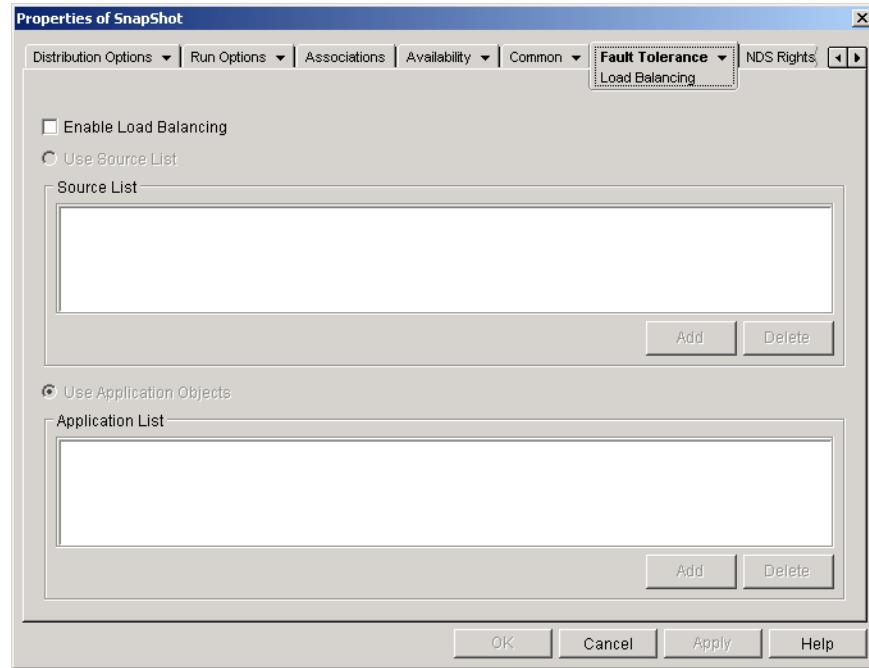
Select an Application object from the Application List > click Delete to remove it from the Application List.

Up-Arrow and Down-Arrow

Select an Application object from the Source List > 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/Explorer tries the Application objects in the order they are listed, from top to bottom.

Load Balancing Page

The Load Balancing property page, shown below, lets you balance the workload required to support the application. With load balancing enabled, Application Launcher/Explorer will pull the application files from all servers offering the application, ensuring a more balanced workload across servers.



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/Explorer selects one of the package sources or Application objects to use. If the package source or Application object is unavailable, Application Launcher/Explorer selects another one.

We recommend you use load balancing when all the servers for the Application objects are located at a single site. If the servers are located at various sites throughout a large wide area network, we recommend you use fault tolerance (Fault Tolerance > Fault Tolerance page). You can implement both methods if necessary. Application Launcher/Explorer will try load balancing first, then fault tolerance.

Enable Load Balancing

Check this option to enable load balancing. The other load balancing options are available only if this option is enabled.

IMPORTANT: 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 a .MSI Application object, check 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/Explorer will randomly select source locations from the list to balance the workload.

Use Source List

Select this option to have Application Launcher/Explorer 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 > click OK to add it to the Source List.

Delete

Select a package source from the Source List > click Delete to remove it from the Source List.

Use Application List

Select this option to have Application Launcher/Explorer 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

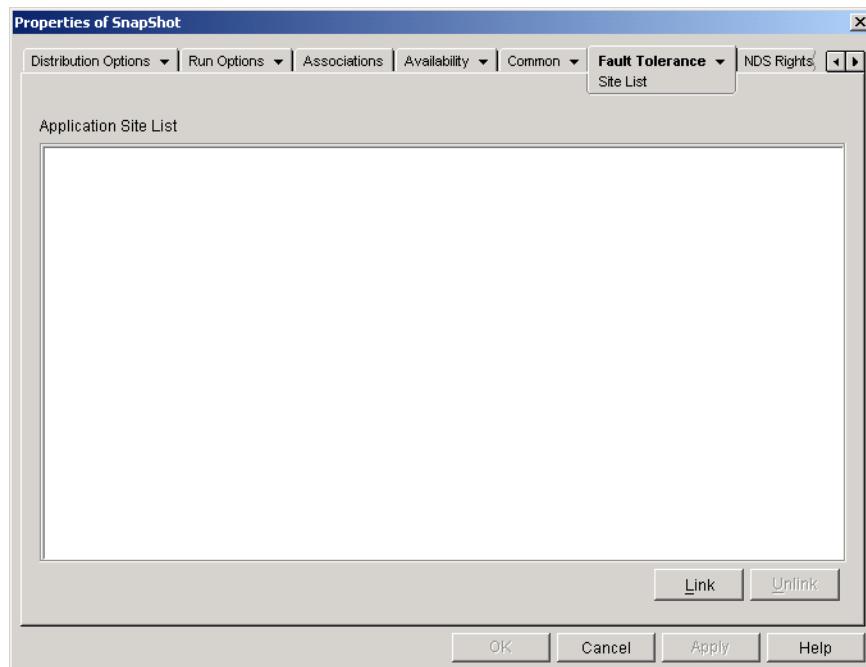
Click Add > browse to and select an Application object > click OK to add it to the Application List.

Delete

Select an Application object from the Application List > click Delete to remove it from the Application List.

Site List Page

The Site List property page, shown below, lists the other Application objects to which this Application object is linked. By linking identical (or similar) Application objects whose application source files are located on servers at different sites, you can create an application site list that gives users who travel from site to site fast access to their applications while reducing WAN traffic and associated costs.



For example, users at two different sites run a spreadsheet application from servers that are local to their sites. Different Application objects are used to distribute and launch the spreadsheet application from the two servers. If you link the two Application objects, any Site1 users who travel to Site2 will be able to run the application from the Site2 server the same way they would run it from their own Site1 server. The same would be true for Site2 users who travel to Site1.

You can link to one other Application object only. 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 linked to: App2  
App2 linked to: App1  
App3 linked to: (nothing)
```

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 linked to: App2, App3  
App2 linked to: App1, App3  
App3 linked to: App1, App2
```

Link

Click Link > browse for and select the Application object you want to link to > click OK to add it to the Application Site List. The list will also include any other Application objects that the selected Application object is linked to.

Unlink

Click Unlink to remove the Application object's link to the Application objects displayed in the Application Site List.

15 Macros

You can use macros with many of the properties available on the Application object pages in ConsoleOne®. You can use any of the five types of macros listed below.

- ◆ Application object macros
- ◆ Special Windows* macros
- ◆ Login script macros
- ◆ NDS® attribute macros
- ◆ Environment 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 may vary from workstation to workstation.

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.

- ◆ “Application Object Macros” on page 189
- ◆ “Special Windows Macros” on page 191
- ◆ “Login Script Macros” on page 194
- ◆ “NDS Attribute Macros” on page 196
- ◆ “Environment Variable Macros” on page 197
- ◆ “Macro Precedence” on page 197
- ◆ “Application Object Properties Where Macros Can Be Used” on page 198
- ◆ “Refreshing Macro Information” on page 198

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™/Explorer will distribute the application to the user's workstation, and TARGET_PATH defines the workstation location to which Application Launcher/Explorer will copy 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:

- ◆ “[Nesting Macros](#)” on page 190
- ◆ “[Defining an Application Object Macro](#)” on page 190

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/Explorer would substitute 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, NDS attribute macros, environment variable macros, login script macros, and other Application object macros.

Defining an Application Object Macro

- 1 In ConsoleOne, right-click the Application object for which you want to define the macro > 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 190.

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 191.

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 “[Application Object Properties Where Macros Can Be Used](#)” on page 198.

Defining a Prompted Macro

- 1** On the Macros page, click Add > Prompted > Drive to create a macro that prompts the user for a drive or click String to create a macro that prompts the user for text input.
- 2** Fill in the following fields:
 - Macro Name:** Enter any name that describes the macro. Do not use spaces in the name.
 - Prompt Text:** Enter 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 that will be displayed as the default drive. For a prompted string macro, enter the text that will be displayed as the default text.
 - Minimum Disk Space in MB:** Enter the minimum amount of free disk space required. The user will not be allowed to install to any drive that does not meet the minimum disk space requirement.
 - Maximum String Length in Chars (Prompted String Macro Only):** Enter 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 “[Application Object Properties Where Macros Can Be Used](#)” on page 198.

Special Windows Macros

A special Windows macro is one that defines Windows 95/98 and Windows NT*/2000/XP directories. The typical paths listed below are based on default installations and may or may not match your specific setup. On Windows 95/98 workstations, macros behave differently if User Profiles are enabled.

Suppose that you have installed Windows to drive D: (for example, D:\WINDOWS). However, an application installation expects Windows to be on drive C: (for example, C:\WINDOWS). Using the *WinDisk macro, you can substitute drive C: with the macro for the files that require it.

The following two sections describe the special Windows macros you can use.

- ◆ “[Windows 95/98 and Windows NT/2000/XP Macros](#)” on page 192
- ◆ “[Windows 2000 Macros](#)” on page 193

For information about using the macros in a valid Application object property, see “[Application Object Properties Where Macros Can Be Used](#)” on page 198.

Windows 95/98 and Windows NT/2000/XP Macros

The following macros apply to Windows 95/98 and Windows NT/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%).

Macro	Description
%*AppData %	File system directory that serves as a common repository for application specific data. Typically C:\WINNT\PROFILES\user name\APPLICATION DATA. 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 C:\WINNT\PROFILES\ALL USERS\DESKTOP. If not available, the *DESKTOP value will be used.
%*CommonPrograms%	File system directory that contains the directories for the common program groups that appear on the Start menu for all users. Typically C:\WINNT\PROFILES\ALL USERS\START MENU\PROGRAMS. If not available, the *PROGRAMS value will be used.
%*CommonStartMenu%	File system directory that contains the programs and folders that appear on the Start menu for all users. Typically C:\WINNT\PROFILES\ALL USERS\START MENU. If not available, the *STARTMENU value will be 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 or starts Windows 95/98. Typically this directory is C:\WINNT\PROFILES\ALL USERS\START MENU\PROGRAMS\STARTUP. If not available, the *STARTUP value will be used.
%*CommonWinDesktop%	Windows NT common desktop directory (C:\WINNT\PROFILES\ALL USERS\DESKTOP). This is a Novell Application Launcher 2.0 macro that is allowed for backward compatibility.
%*Cookies%	File system directory that contains the user's cookies. Typically C:\WINDOWS\COOKIES or C:\WINNT\PROFILES\username\COOKIES.
%*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 C:\WINDOWS\DESKTOP or C:\WINNT\PROFILES\username\DESKTOP.
%*Favorites%	File system directory that serves as a common repository for the user's favorite items. Typically this directory is C:\WINDOWS\FAVORITES or C:\WINNT\PROFILES\username\FAVORITES.
%*Fonts%	Virtual folder containing fonts. Typically C:\WINDOWS\FONTS or C:\WINNT\FONTS.
%*History%	File system directory that contains the user's history of visited Internet addresses. Typically C:\WINDOWS\HISTORY or C:\WINNT\PROFILES\username\HISTORY.
%*NetHood%	File system directory containing objects that appear in the network neighborhood. Typically C:\WINDOWS\NETHOOD or C:\WINNT\PROFILES\username\NETHOOD.
%*Personal%	File system directory that serves as a common repository for documents. Typically C:\MYFILES or C:\WINNT\PROFILES\username\PERSONAL.
%*PrintHood%	File system directory that serves as a common repository for printer links. Typically C:\WINNT\PROFILES\username\PRINTHOOD. Only used by Windows NT.
%*Programs%	File system directory that contains the user's program groups (which are also file system directories). Typically C:\WINDOWS\START MENU\PROGRAMS or C:\WINNT\PROFILES\username\START MENU\PROGRAMS.

Macro	Description
%*Recent%	File system directory that contains the user's most recently used documents. Typically C:\WINDOWS\RECENT or C:\WINNT\PROFILES\username\RECENT.
%*SendTo%	File system directory that contains Send To menu items. Typically C:\WINDOWS\SENTO or C:\WINNT\PROFILES\username\SENDTO.
%*StartMenu%	File system directory containing Start menu items. Typically C:\WINDOWS\START MENU or C:\WINNT\PROFILES\username\START MENU.
%*Startup%	File system directory that corresponds to the user's Startup program group. Typically C:\WINDOWS\START MENU\PROGRAMS\STARTUP or C:\WINNT\PROFILES\username\START MENU\PROGRAMS\STARTUP.
%*TempDir%	Windows temporary directory (C:\WINDOWS\TEMP).
%*Templates%	File system directory that serves as a common repository for document templates. Typically C:\WINDOWS\SHELLNEW or C:\WINNT\SHELLNEW.
%*WinDesktop%	Windows desktop directory (C:\WINDOWS\Desktop or C:\WINNT\PROFILES\username\Desktop for Windows NT). This is an Application Launcher 2.0 macro that is allowed for backward compatibility.
%*WinDir%	Windows directory. Typically C:\WINDOWS or C:\WINNT.
%*WinDisk%	Drive letter (plus colon) for the Windows directory. Typically C:.
%*WinSysDir%	Windows system directory. Typically C:\WINDOWS\SYSTEM or C:\WINNT\SYSTEM32.
%*WinSysDisk%	Drive letter (plus colon) for the Windows system directory. Typically C:.

Windows 2000 Macros

The following macros apply to Windows 2000 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%).

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. Typically C:\DOCUMENTS AND SETTINGS\username\START MENU\PROGRAMS\ADMINISTRATIVE TOOLS.
%*CommonAdminTools%	File system directory that contains the administrative tools that appear in the Control Panel for all users who log in to Windows 2000. Typically C:\DOCUMENTS AND SETTINGS\ALL USERS\START MENU\PROGRAMS\ADMINISTRATIVE TOOLS.
%*CommonAppData%	File system directory that contains the application-specific data for all users who log in to Windows 2000. Typically C:\DOCUMENTS AND SETTINGS\ALL USERS\APPLICATION DATA.
%*CommonDocuments%	File system directory that contains the documents shared by all users who log in to Windows 2000. Typically C:\DOCUMENTS AND SETTINGS\ALL USERS\DOCUMENTS.
%*CommonTemplates%	File system directory that contains the document templates shared by all users who log in to Windows 2000. Typically C:\DOCUMENTS AND SETTINGS\ALL USERS\TEMPLATES.

Macro	Description
%*MyPictures%	File system directory that contains a specific user's graphic files. Typically C:\DOCUMENTS AND SETTINGS\username\MY DOCUMENTS\MY PICTURES.
%*ProgramFiles%	File system directory that contains the program files. Typically C:\PROGRAM FILES.
%*ProgramFilesCommon%	File system directory that contains the program files shared by multiple applications. Typically C:\PROGRAM FILES\COMMON.
%*UserProfile%	File system directory that contains the logged-in user's profile. Typically C:\DOCUMENTS AND SETTINGS\username.

Login Script Macros

Application Launcher/Explorer supports many, but not all, login script variables.

- ◆ “[Supported Login Script Macros](#)” on page 194
- ◆ “[Unsupported Login Script Macros](#)” on page 195

For information about using the macros in a valid Application object property, see “[Application Object Properties Where Macros Can Be Used](#)” on page 198.

Supported Login Script Macros

The following is a list of supported login script variables and what they mean. Alternate variable names are shown in parentheses.

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 NDS monitored connection. For example: APPS_PROD.
%FULL_NAME%	Full name attribute of the User object. This is supported by virtue of Application Launcher/Explorer's support for NDS 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 NDS Surname attribute). For example: Jones.
%LOGIN_NAME%	First eight bytes of the user's NDS 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.

Macro	Description
%OS_VERSION%	Version of the OS. For example: v5.00 (Win3x shows the DOS version, Win 95/98 and Win NT/2000 shows the Windows version).
%OS	OS type. For example: MSDOS, WIN95/98, WINNT, WIN2000. (Win3 shows MSDOS.)
%PLATFORM%	Platform running. For example: WIN, W95/98, WNT, W2000.
%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.
%WORKSTATION_DN%	The distinguished name and context of the workstation object.
%YEAR%	Full year number. For example: 2000, 2001.

Unsupported Login Script Macros

The following is a list of login script macros that Application Launcher/Explorer does not support:

ACCESS_SERVER
 AM_PM
 DAY_OF_WEEK
 DIALUP
 ERROR_LEVEL
 GREETING_TIME
 LOCATION
 LOGIN_ALIAS_CONTEXT
 LOGIN_CONTEXT
 MACHINE
 MONTH_NAME
 NEW_MAIL
 OFFLINE
 PASSWORD_EXPIRES
 REQUESTER_VERSION
 SHELL_VERSION
 SMACHINE
 STATION
 USERID

NDS Attribute Macros

Application Launcher/Explorer supports macros that pull information from the attributes of the currently logged-in user, the current Application object, or from the attributes of other NDS objects.

The following sections explain the macro syntax and provide examples:

- ◆ “[Syntax](#)” on page 196
- ◆ “[Examples](#)” on page 196

For information about using the macros in a valid Application object property, see “[Application Object Properties Where Macros Can Be Used](#)” on page 198.

Syntax

NDS attribute macros use the following syntax:

`%object_name.container;NDS_attribute%`

Element	Description
%	Flags the text as a macro. The entire macro must be enclosed in % characters.
.object_name.container;	Indicates the NDS object from which to read the attribute. To use an attribute from the currently logged in user, leave this element out. For example, %CN% would return the common name of the currently logged in user.
NDS_attribute	To use an attribute from the Application object, replace .object_name.container with an asterisk (*). For example, %*;DN% would return the distinguished name of the Application object. Defines the attribute to be read. You can use the ConsoleOne Schema Manager (available from the Tools menu) to view an NDS object's available attributes.

Examples

The following table provides examples of NDS attribute macros.

Macro	Description
%CN%	Returns the common name of the currently logged-in user.
%DN%	Returns the distinguished name of the currently logged-in user.
%Given Name%	Returns the first name of the currently logged-in user.
%Surname%	Returns the last name of the currently logged-in user.
%Full Name%	Returns the full name of the currently logged-in user.

Macro	Description
%Email Address%	Returns the e-mail address of the currently logged-in user.
%*,DN%	Returns the distinguished name of the Application object (the one in which the macro is being used).
%*,App:Caption%	Returns the application icon title given the Application object.
*;App:Path%	Returns the path to the application's executable as defined for the Application object.
%*,App:GUID%	Returns the Application object's global unique identifier (GUID).
%JSmith.Novell;Description%	Returns the description for the JSmtih User object located in the Novell container.
%ABurrough.Novell;Email Address%	Returns the e-mail address for the ABurrough User object located in the Novell container.

Environment Variable Macros

Application Launcher/Explorer supports all valid Windows 95/98 and Windows NT/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 “[Application Object Properties Where Macros Can Be Used](#)” on page 198.

Macro Precedence

To resolve conflicts with macro names, Application Launcher/Explorer uses the following order of precedence:

- ◆ Application object macros
- ◆ Special Windows macros
- ◆ Login script macros
- ◆ NDS 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.

Application Object Properties Where Macros Can Be Used

You can use macros when defining the following Application object properties:

- ◆ Path to Executable (Run Options tab > Applications page)
- ◆ Command Line (Run Options tab > Applications page)
- ◆ Working Directory (Run Options tab > Applications page)
- ◆ Drive Mapping Path (Common tab > Drives/Ports page)
- ◆ Capture Port Path (Common tab > Drives/Ports page)
- ◆ Key, Value Name, and Value Data strings (Distribution Options tab > Registry Settings page)
- ◆ Section, Value Name, and Value Data strings (Distribution Options tab > INI Settings page)
- ◆ Source File, Target File, Directory Name (Distribution Options tab > Application Files page)
- ◆ Find File, Find Next File, Add string (Distribution Options tab > Text Files page)
- ◆ All properties (Distribution Options tab > Icons/Shortcuts page)

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/Explorer to recognize and treat the macro as a macro.

Refreshing Macro Information

When Application Launcher/Explorer starts, it caches the macro information defined for each Application object associated with the logged in user or workstation. The Novell Client does this as well. As a result of this, if you change information associated with a macro, the change will not be reflected on a user's workstation until he or she logs out of NDS and logs in again.

For example, if you are using the %DN% (distinguished name) NDS attribute macro in a folder name and you move a User object from one NDS container to another, the change will not be reflected on the user's workstation until he or she logs out of the Novell Client™ and logs in again.

16 Application Launcher Tools

ZENworks® for Desktops (ZfD) provides several tools that may be helpful as you manage applications for your users. These tools, located on the Tools > Application Launcher Tools menu in ConsoleOne®, are explained in the following sections:

- ◆ “Export Application Object” on page 199
- ◆ “Show Inherited Applications” on page 199
- ◆ “Search and Replace” on page 199
- ◆ “Sync Distribution GUIDs” on page 200
- ◆ “Generate New GUIDs” on page 200
- ◆ “AOT/AXT File Tools” on page 201
- ◆ “Create Virtual CD” on page 201

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 > Application Launcher Tools > Export Application Object.
- 3 Select the export option you want > click Next > follow the prompts to complete the export.

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 > Application Launcher Tools > Show Inherited Applications.

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 > 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** Check 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 > 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.

Sync Distribution GUIDs

This tool enables you to synchronize the GUIDs (global unique identifiers) of two or more Application objects. Synchronizing GUIDs causes all Application objects to be distributed and installed whenever one Application object is. This can be useful, for example, when you want to distribute a suite of applications to users.

- 1** In ConsoleOne, select the Application objects whose GUIDs you want to synchronize. Control-click to select multiple objects.
- 2** Click the Tools menu > Application Launcher Tools > Sync Distribution GUIDs.

Generate New GUIDs

This tool generates a new GUID for an Application object. If you've synchronized the GUIDs of several Application objects, you can use this tool unsynchronize the GUIDs so that distributing one of the Application objects will not distribute all of them.

This tool requires careful use. When an Application object is distributed to a workstation, its GUID is stored in the Windows registry. Novell® Application Launcher™/Explorer uses the GUID to determine whether or not the application has been distributed to the workstation. After the new GUID is generated, Application Launcher/Explorer may re-distribute the application because the GUID in the Windows registry will not match the new GUID for Application object.

- 1 In ConsoleOne, select the Application objects whose GUIDs you want to change. Control-click to select multiple objects.
- 2 Click the Tools menu > Application Launcher Tools > Generate New GUIDs.

AOT/AXT File Tools

This tool converts a .AOT file to a .AXT file or a .AXT file to a .AOT file.

- 1 In ConsoleOne, click the Tools menu > Application Launcher Tools > AOT/AXT File Tools.
- 2 To convert a .AOT file to a .AXT file, click AOT -> AXT.

or

To convert a .AXT file to a .AOT file, click AXT -> AOT.
- 3 In the Open dialog box, browse for and select the file you want to convert > click Open.
- 4 In the Save As dialog box, type a name for the new file > click Save As. The correct extension (.AOT or .AXT) will be added automatically.

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 media can then be given to users who are disconnected from the network so that they can install the applications from the CD. For detailed information and instructions, see “[Distributing Applications to Disconnected Workstations](#)” on page 80.

A Documentation Updates

This section lists updates to the Application Management part of the *Administration* guide that have been made since the initial release of Novell® ZENworks® for Desktops 3.2. The information will help you to keep current on documentation updates and, in some cases, software updates (such as a Support Pack release).

The information is grouped according to the date when the *Administration* guide was updated and republished:

- ◆ “October 31, 2002 (ZfD 3.2 Support Pack 1)” on page 203
- ◆ “May 15, 2003” on page 204
- ◆ “August 15, 2003” on page 205

October 31, 2002 (ZfD 3.2 Support Pack 1)

Location	Change
“Setting Configuration Options” on page 37	Under step 5, added information about the new OnDemand UsageServer setting for Novell Application Launcher™/Explorer.
Chapter 9, “Configuring Application Fault Tolerance and Load Balancing,” on page 83	Added a new section about using application fault tolerance, load balancing, and site lists.

Location	Change
“Icons/Shortcuts Page (.AOT/.AXT Application Object Only)” on page 122	Clarified the information about the Distribute Always and Track Distribution Per User settings found on the various Distribution pages of an Application object.
and	
“Registry Page” on page 124	
and	
“Application Files Page (.AOT/.AXT Application Object Only)” on page 127	
and	
“INI Settings Page” on page 130	
and	
“Text Files Page (.AOT/.AXT Application Object Only)” on page 132	
and	
“Options Page” on page 138	
“Pre-Install Schedule Page (.AOT/.AXT Application Object Only)” on page 136	Clarified information about how the Range of Days > Time for Selected Range setting works. The selected times are used for one day or multiple days. If you select all seven days, the times are ignored and the application is available 24 hours a day for all seven days.
and	
“Schedule Page” on page 162	

May 15, 2003

Location	Change
Global	Changed Windows NT/2000 to Windows NT/2000/XP to reflect support for Windows XP.
“Application Launcher/Explorer Command Line Switches” on page 46	Corrected the syntax for using the colon (nal : or nalexpld :) startup switch to suppress the initial splash screen. There must be a space between the executable name and the colon.
“Order Icon Display (Also Determines Force Run Priority)” on page 118	Corrected information regarding this field. It does not determine icon display order in Application Launcher/Explorer. It only determines the execution order of applications marked as Force Run.
GUID under “Options Page” on page 138	Corrected information about the GUID setting located on an Application object’s Options page (Distribution Options tab > Options page). The GUID does not change when the version number changes.
“Path to Executable File” on page 141	Under the Parameters heading, noted that the Parameter field’s character limit is 250.

August 15, 2003

Location	Change
"Imaging Page" on page 177	Provided more information about how application add-on images are applied during imaging of a workstation. Also explained the NALCACHE directory rights that non-administrative users need in order for Novell Application Launcher/Explorer to access application's delivered as add-on images.

