Policy and Distribution Services

Novell® ZENworks® 7 Server Management Policy and Distribution Services is a software, configuration, and behavioral management system for servers. Through Policy and Distribution Services, you can:

- Control the versions of software installed on servers throughout your network
- Define and enforce a standard configuration on any given set of servers
- Control the behavior of servers in given situations, such as downing a server, backing up volumes, managing thresholds exceeded, and so on

Policy and Distribution Services has three components:

- **Tiered Electronic Distribution:** Simplifies data delivery and server policy implementation
- **Server Policies:** Simplifies configuration and management of your servers
- **Server Software Packages:** Simplifies the installation of software

You can administer Policy and Distribution Services by using the following:

- **ConsoleOne 1.3.6,** where you can create and configure Server Management objects and perform management tasks for Policy and Distribution Services.
- **ZENworks Server Management role in Novell iManager,** where you can perform management tasks for Policy and Distribution Services using iManager from any workstation where Internet Explorer 5.5 or later is available.

This Policy and Distribution Services documentation contains the following sections:

Post-Installation Setup

<div style="text-align: right; font-size: 3em;">1</div>

To use the Tiered Electronic Distribution capability of Novell® ZENworks® Server Management effectively, you must correctly install and configure its components on your network. You should have already performed a basic installation of Policy and Distribution Services (see "Installation on NetWare and Windows Servers" in the *Novell ZENworks 7 Server Management Installation Guide*.

For information on configuring policies, see Chapter 4, "Server Policies," on page 197.

This section provides you with the concepts, a planning worksheet, and instructions to help you further configure your Tiered Electronic Distribution system. For more detailed information, see Chapter 3, "Tiered Electronic Distribution," on page 85.

The information provided in the following sections will help you to add new Distributors as needed, finish installing the Subscriber software as needed, configure a Distributor's routing hierarchy, create some Distributions, and send those Distributions:

- Section 1.1, "Planning Your Distribution System," on page 29

  In this section, you can use the planning worksheet to keep track of the decisions you need to make. Then you can easily perform your planned configurations from the information on the planning worksheet.

- Section 1.2, "Configuring Your Distribution System," on page 50

  This section provides the steps for configuring your distribution system.

- Section 1.3, "Managing Your Distribution System," on page 61

  This section provides an overview of how you can manage your distribution system using Novell ConsoleOne® and Novell iManager.

- Appendix H, "Configuration Planning Worksheet," on page 437

  The planning worksheet contains basic information for each worksheet entry. It also contains links to where you can view more information to better understand a worksheet entry.

  The worksheet should not be used in place of the procedures in Section 1.2, "Configuring Your Distribution System," on page 50, because the worksheet contains only planning information; it does not contain information for the procedures that are not planned.

1.1 Planning Your Distribution System

Use these sections in the following order:

1. "Overview" on page 30
2. "Selecting Your Distributions" on page 32
3. "Understanding Your Network Topology" on page 37
4. "Are Additional Distributors Needed?" on page 38
5. "Other Subscribers To Be Installed?" on page 41
6. "Determining the Distribution Flow" on page 42
7. "Understanding Distribution Security" on page 44

1.1.1 Overview

Policy and Distribution Services contains three components:

- **Tiered Electronic Distribution** is a distribution system for your network.

    - It is a way to manage your network servers through the distribution of electronic data between servers.

    - It uses a tiered architecture for distribution efficiency. For example, workload sharing: one server can service many others, then each of those many servers can also service many more, and so on to any number of tiers.

    - It provides Distribution scheduling for efficient bandwidth usage, such as distributing during off-peak hours.

    - It provides security to prevent unauthorized tampering with the Distributions.

- **Server Policies** is a system for managing the configuration and behavior of your servers.

- **Server Software Packages** is a feature for automating the installation and upgrading of software on your servers.

Tiered Electronic Distribution is usually involved when you use any of these components, because most policies and all Server Software Packages are distributed. Therefore, in this section we will concentrate on understanding and configuring Tiered Electronic Distribution. See the following sections for more information on the other two components of Policy and Distribution Services:

- Chapter 4, "Server Policies," on page 197

- Chapter 5, "Server Software Packages," on page 243

The following sections provide basic information that will help you to understand Tiered Electronic Distribution and what you will need to know to configure it:

- "What Can You Distribute?" on page 30

- "How Is Data Distributed?" on page 31

- "What Do You Need to Know to Plan Your Distribution System?" on page 31

What Can You Distribute?

The types of electronic data you can distribute using Tiered Electronic Distribution include:

Table 1-1   *Distribution Types*

Distribution Type	Content Distributed
Desktop Application	Desktop Application objects and files created in ZENworks Desktop Management
File	Files and directories contained on the Distributor server's file system

Distribution Type	Content Distributed
FTP	Files and directories from an FTP source
HTTP	Content from an HTTP source
MSI	Contains software to be installed in a Windows* environment by the MSI engine
Policy Package	Policies for controlling servers
RPM	RPM packages for Linux and Solaris* servers (but only for Solaris if RPM is installed to the Solaris machine)
Software Package	Server Software Packages for automatically installing or upgrading software on your servers

From this list, you can see that there is a variety of electronic data types that you can distribute to your servers.

How Is Data Distributed?

Tiered Electronic Distribution sends Distribution files from Distributor servers to Subscriber servers. The basic distribution process is as follows:

1. Decide what you want to distribute.
2. Create the Distribution.
3. Create a Channel for the Distribution.
4. Determine which Subscriber servers need this Distribution.
5. Subscribe the Subscriber servers to the Distribution's Channel.
6. Make sure the applicable schedules are set (Build, Send, and Extract).
7. Send the Distribution by refreshing the Distributor, which causes the Distribution to be built according to the Distribution's Build schedule, and sent according to the Channel's Send schedule.
8. The Distribution is extracted on the Subscriber servers according to their Extract schedules.
9. The Distributions are used by the Subscriber servers according to the Distribution's type.

From this process, you can see that there are several components of Tiered Electronic Distribution that will need to be created and configured. For more information, see .

What Do You Need to Know to Plan Your Distribution System?

- The Distributions that you want, including:
  - Whether you want to distribute server files, HTTP content, FTP content, or RPM packages
  - If there are any desktop applications to be distributed (affects how you set up Subscriber objects when you have multiple trees)
  - Which policies you needed for managing your servers
  - What server software should have automated installation
- Whether you'll need additional Distributors

- Whether you have both Novell eDirectory™ 8.7.3 and NDS® 6.*x* or 7.*x* in your environment, which adversely affects Distributors (a workaround is available)
- How many databases you'll need for reporting purposes
- Whether you need to complete installation of the Subscriber software to your servers
- Which Subscribers need which Distributions
- Your network's topology (server platforms, slow WANs, firewalls, Network Address Translation [NAT], multiple trees, and so on)
- The system resource and server behavior issues that Tiered Electronic Distribution might create
- Whether you need to encrypt Distributions for certain servers
- Whether you can use Subscriber Groups for channeling Distributions
- How you want the Distributions to flow to the Subscriber servers (the tiered distribution model)
- How you want to schedule the distribution processes to minimize network traffic during business hours

To determine the above information, continue with .

1.1.2 Selecting Your Distributions

This section provides you with basic information for each Distribution type.

You can build your distribution system incrementally by adding Distributions a few at a time, then adding Distributors when needed. You can revisit this process at any time to add new Distributions.

Print a copy of the . Worksheet fill-in instructions are given as you review the planning sections.

Review the following Distribution type sections to select which ones you want to create at this time. Planning worksheet entries are provided for each Distribution type.

-
-
-
-
-
-
-
-

Desktop Application

This Distribution type allows you to distribute Application objects and associated files to specified locations on the eDirectory tree and target Subscriber servers.

For information on integration with Desktop Management, see .

For information on the Desktop Application type of Distribution, see "Desktop Application" on page 119.

Determine whether you want to create a Desktop Application Distribution at this time:

CONFIGURATION PLANNING WORKSHEET

If you want to maintain trustee rights in the Distribution, under item 3 and item 20, indicate that you have Desktop Application Distributions, and therefore each server that receives Desktop Application Distributions must have its Subscriber object and NCP™ server object in the same tree.

Under item 19, enter Desktop Application as the type of Distribution to be created. Also indicate the following:

- ◆ A name for the Distribution that indicates its purpose
- ◆ Names of the servers that need a Desktop Application Distribution

---

File

With this Distribution type you can select files and/or directories from the Distributor server's file system to distribute to a selected location on the Subscriber server's file system.

A Distribution Wizard is available for automating the process of creating the File and FTP types of Distributions. For more information, see Section 3.4.12, "Using the Distribution Wizard," on page 144.

For information on the File type of Distribution, see "File" on page 119.

Determine whether you want to create a File Distribution at this time:

CONFIGURATION PLANNING WORKSHEET

Under item 19, enter File as the type of Distribution to be created. Also indicate the following:

- ◆ A name for the Distribution that indicates its purpose
- ◆ Names of the servers that need a File Distribution

FTP

With this Distribution type you can create a Distribution consisting of files from one or more FTP sources. Each source can contain one or more directories and/or files.

A Distribution Wizard is available for automating the process of creating the File and FTP types of Distributions. For more information, see Section 3.4.12, "Using the Distribution Wizard," on page 144.

For information on the FTP type of Distribution, see "FTP" on page 121.

Determine whether you want to create an FTP Distribution at this time:

CONFIGURATION PLANNING WORKSHEET

Under item 19, enter FTP as the type of Distribution to be created. Also indicate the following:

- ◆ A name for the Distribution that indicates its purpose
- ◆ Names of the servers that need an FTP Distribution

HTTP

With this Distribution type you can create a Distribution consisting of one or more HTTP sources. Each source can contain one or more target entries.

For information on the HTTP type of Distribution, see "HTTP" on page 121.

Determine whether you want to create an HTTP Distribution at this time:

CONFIGURATION PLANNING WORKSHEET

Under item 19, enter HTTP as the type of Distribution to be created. Also indicate the following:

- ◆ A name for the Distribution that indicates its purpose
- ◆ Names of the servers that need an HTTP Distribution

MSI

This is a Distribution of MSI packages that are installed by the MSI engine in a Windows environment.

For information on the MSI type of Distributions, see "MSI" on page 121.

Determine whether you want to create an MSI Distribution at this time:

CONFIGURATION PLANNING WORKSHEET

Under item 19, enter MSI as the type of Distribution to be created. Also indicate the following:

- ◆ A name for the Distribution that indicates its purpose
- ◆ Names of the servers that need an MSI Distribution

Policy Package

This Distribution type provides the mechanism for applying any of the policies in Table 1-2 to Subscriber servers:

Table 1-2   *Policies*

Policy	Description
Copy Files	Enables copying of files on a server from one location to another by using policy configurations.
NetWare Set Parameters	Specifies and optimizes selected NetWare® Set Parameters for a server or group of servers.
Prohibited File	Used to monitor and enforce the deletion or moving of unauthorized files from a specified volume/drive or directory/folder.
Scheduled Down	Schedules when a server should go down, and whether it should be brought back up automatically.
Scheduled Load/Unload	Automates the loading and unloading order of NLM™ and Java* Class processes for the selected servers, and for starting and stopping Windows services.
Search	Used in Server Management to enable the Distributor Agent to locate and use policies in the Service Location Package.
Server Down Process	Controls which processes to follow and which conditions to meet before downing a server.
Server Scripts	Automates script usage on your servers.
SMTP Host	Sets the TCP/IP address of the relay host that processes outbound Internet e-mail.
SNMP Community Strings	Allows you to receive and respond to SNMP requests.
SNMP Trap Targets	Sets SNMP trap targets for associated eDirectory objects for reporting purposes.
Text File Changes	Automates changes to text files.
Tiered Electronic Distribution	Sets defaults for the Distributor and Subscriber objects.
ZENworks Database	Sets the DN for locating a ZENworks Database object and the path to the database file. The database is used by Policy and Distribution Services for logging successes and failures that are used in creating reports. The database location specified during installation can be overridden by creating and enabling this policy.
ZENworks Server Management	Contains basic configuration parameters for Policy and Distribution Services, such as status logging, defining the server console prompt for the Policy/Package Agent, setting its working path, and setting a database purging limit.

For more information on each policy, see Section 4.1.6, "Server Policy Descriptions," on page 204.

For information on policies and policy packages, see Chapter 4, "Server Policies," on page 197.

For more information on the Policy Package type of Distribution, see "Policy Package" on page 122.

Determine whether you want to create a Policy Package Distribution at this time:

CONFIGURATION PLANNING WORKSHEET

Under item 19, enter Policy Package as the type of Distribution to be created. Also indicate the following:

 * Names of the policies
 * For each policy, names of servers that need the policy

RPM

This is a Linux or Solaris platform Distribution. You can distribute Red Hat* Package Manager (RPM) packages using the RPM Distribution.

For information on the RPM type of Distribution, see "RPM" on page 123.

Determine whether you want to create an RPM Distribution at this time:

CONFIGURATION PLANNING WORKSHEET

Under item 19, enter RPM as the type of Distribution to be created. Also indicate the following:

 * A name for the Distribution that indicates its purpose
 * Names of the servers that need an RPM Distribution

Software Package

This Distribution type allows you to distribute Server Software Packages that you create in ConsoleOne in the Server Software Package namespace. You first create a `.spk` file, then compile it into the `.cpk` file that is distributed.

For information on Server Software Packages, see Chapter 5, "Server Software Packages," on page 243.

For information on the Software Package Distribution type, see "Software Package" on page 123.

Determine the software packages you want to create at this time:

CONFIGURATION PLANNING WORKSHEET

Under item 19, enter Software Package as the type of Distribution to be created. Also indicate the following:

 * A name for the Distribution that indicates its purpose
 * Names of servers that need a Software Package Distribution

---

1.1.3 Understanding Your Network Topology

In order for you to efficiently manage your distribution system, you need to know your network's topology. For example:

- What are your server platforms?
- How many servers do you have per platform?
- Where are your servers located in relation to WAN links and firewalls?
- Is Network Address Translation (NAT) being used?
- Where are your slow network links?

This type of information is used to help you configure the best distribution management solution for your network.

To obtain information concerning your network:

1 Note the trees where you extended the schema for Server Management.

CONFIGURATION PLANNING WORKSHEET

Under item 1, provide the names of the trees in your network where you extended the schema for Server Management.

---

2 Draw a diagram of your network structure.

You will use this diagram later to determine distribution routes.

Indicate the following on your diagram:

- Where slow links exist
- The number of servers on each LAN
- The number of servers outside a firewall
- The number of servers using NAT

3 Draw tree diagrams that show how your trees are currently organized. Include the main containers, such as:

- The containers that represent geographic locations (a physical tree design)
- The containers that represent the corporate organization (a logical tree design)
- The containers where servers reside (for Distributors and Subscribers)

4 Indicate the following on your tree diagrams:

- Where servers are located that could be Distributors (NetWare, Windows, Linux, or Solaris servers that exceed the minimum Server Management requirements)
- Containers where there are slow network connections

  This should match where you indicated slow connections on your network diagram.

5 Indicate the following on your network diagram:

- Where the servers are located (as you just noted on the tree diagrams) that could be Distributors

1.1.4 Are Additional Distributors Needed?

When installing Policy and Distribution Services for the first time, you installed one Distributor with a database file. Generally, you'll need Distributors according to your corporate structure or geographic locations.

Distributor server workload, including the ability to complete Distribution building tasks, should also determine how many Distributors you need. For example, if you have a very large Distribution that you want built during off-peak hours, which does not need to be sent immediately, and also have virus pattern Distributions that do need to be sent immediately, you might need two different Distributors, one with a daily refresh schedule (because you are only going to be building the Distribution once per day), and another with a frequent refresh schedule for discovering new virus pattern changes, so that their Distributions can be built and sent on time.

Use your diagrams to determine whether you need to install additional Distributors.

CONFIGURATION PLANNING WORKSHEET

Under item 2, provide the names of the servers where you want to install the Distributor software.

You can always add Distributors later after you've seen how your Distributor servers handle their Distribution building and sending workload, you can determine whether to add additional Distributors for spreading that workload.

You also need to determine the following information for each Distributor:

- "Distributor Properties" on page 38
- "Software Installation Paths" on page 39
- "Whether a Distributor Server Will Host a Server Management Database" on page 39
- "Whether Distributors Might Exist in a Mixed eDirectory Environment" on page 40

Distributor Properties

You can change the following Distributor properties from the defaults during installation:

- **Object name:** If you want to rename the Distributor object, we recommend that you maintain the server's identity in the name, including the fact that it is a Distributor.
- **Container:** Plan on using the container where you previously installed Distributor objects.

  If eDirectory is not installed on the Windows 2000/2003 server that you want to be a Distributor, a default container object is not displayed for that server during installation. Therefore, determine the container for that Distributor object.
- **Working directory:** You can use a different volume, drive, or directory path for the Distributor's working files than the default path.

  Because the working directory has the potential to be quite large (depending on the size of the Distributions), make sure you have enough disk space.

  The default volume on a NetWare server is sys:. For NetWare servers we strongly recommend that you specify a different volume.

  The default working directory path for NetWare and Windows servers is:
  ```
  \zenworks\pds\ted\dist
  ```

For Linux or Solaris servers the path is:

`/var/opt/novell/zenworks/zfs/pds/ted/dist`

The Distributor's working directory is also used whenever a Distribution is created. A directory is created under the working directory using the DN of the Distribution object.

For more information on the working directory, see .

CONFIGURATION PLANNING WORKSHEET

Under item 7, provide the property information for the Distributor that you want to be different than the defaults. This includes object names, containers for the object, and working directories.

Software Installation Paths

Server Management uses the following default installation paths:

- **NetWare:** `sys:`

  You can select a different volume.

- **Windows:** `C:`

  You can select a different drive.

The Linux or Solaris path cannot be changed.

CONFIGURATION PLANNING WORKSHEET

Under item 5, provide the installation path information for the Distributor if it is different from the default path. Include the identities of the Distributors where you have different Distributor installation paths.

Under item 6, provide the installation path information for the Subscriber if it is different from the default path. Include the identities of the Subscribers where you have different Subscriber installation paths.

Whether a Distributor Server Will Host a Server Management Database

You can have multiple Server Management databases in the tree, and you can install the database to both NetWare and Windows servers.

The database is used by Policy and Distribution Services to log successes and failures for the Server Policies or Tiered Electronic Distribution components. Policy and Distribution Services can function normally without a database, because it uses the `zfslog.db` file to only log information for reports. `Zfslog.db` for Policy and Distribution Services does not contain any configuration information.

To determine whether you want each Distributor to have its own database, or have all Distributors share the same database, you need to determine how you want information reported. Consider the following to determine how many databases to have in the tree:

- **WAN traffic:** Tiered Electronic Distribution does not perform a large number of database updates, so the actual impact on system resources should be minimal. The greatest impact could be the time it takes to perform the transaction. However, if you have slow WAN connections, you might not want database logging to occur over the WAN.

- **Multiple Distributors:** If you have multiple Distributors in the tree, you can have one database for each, or have them share one or more databases. The type of Distributor reporting you want should determine whether to have a separate database for each. For example, are your Distributors specialized in the types of Distributions they'll send?

- **Consolidated reporting:** To have only one report for all of your Tiered Electronic Distribution information, install only one database object and file and have all Tiered Electronic Distribution Distributors log to that one file, regardless of WAN traffic considerations. Use the ZENworks Database policy (Service Location Package) to direct all Distributors to that database file.

- **Specialized reporting:** You might want reports that are specific to a region or group of servers. You can install a database object and file for each region and have the Distributors in those regions or server groups log to that database. Use a separate ZENworks Database policy (Service Location Package) to direct each Distributor to its desired database file.

For more information, see Chapter 10, "ZENworks Database," on page 359.

IMPORTANT: Make sure you select a server for the database where you are installing the Subscriber/Policies option. The Purge Database option in the ZENworks Server Management policy (Distributed Server Package) works only if the Policy/Package Agent software and the `zfslog.db` file are located on the same server.

CONFIGURATION PLANNING WORKSHEET

Provide the following information for each Database object to be created:

- Under item 4, provide the name of the Distributor server that hosts the Server Management database file.
- Under item 9, provide the installation path information that is different from the default path.
- Under item 10, provide a name for the Database object, if different from the default.
- Under item 11, provide the eDirectory container where the Database object should be created.

Whether Distributors Might Exist in a Mixed eDirectory Environment

Server Management can run in a mixed eDirectory environment. For example, your network might have both eDirectory 8.*x* and NDS® 6.*x* or 7.*x* installed.

However, eDirectory 8.*x* (only 8.6.2, 8.7.1, or 8.7.3 or later) is required for Server Management so that its objects can be placed in the tree during installation of the product. eDirectory must be installed with the master replica somewhere in your network, but not necessarily on a server where you are installing the Server Management software.

Also, ZENworks 7 Distributor servers must be running eDirectory 8.*x*.

The only requirement for any Server Management server is that it can communicate with the server where the eDirectory master replica (of the partition where its NCP Server object resides) is installed. Therefore, you do not need to install eDirectory on each server where you will install Server Management.

Select an IP address of any server in your tree that is using eDirectory 8.*x*. This can even be the IP address of the Distributor server itself, if the server is running eDirectory 8.*x*.

CONFIGURATION PLANNING WORKSHEET

Under item 12, provide the IP address of a server using eDirectory 8.*x*.

1.1.5 Other Subscribers To Be Installed?

When you first installed Policy and Distribution Services, you might not have installed the software to all of your servers. If you determined that you wanted to install the Subscriber software incrementally to your servers, you can complete another stage at this time.

You can change the following Subscriber properties from the defaults during installation:

- ◆ **Object name:** If you want to rename the Subscriber object, we recommend that you maintain the server's identity in the name, including the fact that it is a Subscriber.

- ◆ **Container:** Plan on using the container where you previously installed Subscriber objects.

    You should place Subscriber server objects in containers matching their operating systems. For example, a NetWare container for NetWare servers, and a Windows container for Windows servers.

    If eDirectory is not installed on the Windows 2000/2003 server that you want to be a Subscriber, a default container object is not displayed for that server during installation. Therefore, determine the container for that Subscriber object.

- ◆ **Working directory:** You can use a different volume, drive, or directory path for the Subscriber's working files than the default path.

    Because the working directory has the potential to be quite large (depending on the size of the Distributions), make sure you have enough disk space. The default volume on a NetWare server is sys:. For NetWare servers we strongly recommend that you specify a different volume.

    You might need to provide different paths for your Subscriber servers. For example, sys: for NetWare servers and D: for Windows servers. You can use variables for path data, such as the volume/drive designation. For more information, see Chapter 9, "Variables," on page 349.

    The default working directory path for NetWare and Windows servers is:

    `\zenworks\pds\ted\sub`

    For Linux and Solaris servers, the path is:

    `/var/opt/novell/zenworks/zfs/pds/ted/sub`

    For more information on working directories, see Section 3.12, "Working Directories," on page 189.

CONFIGURATION PLANNING WORKSHEET

Under item 3, provide the names of the servers where you want to install the Subscriber software at this time.

For each Subscriber to be installed, under item 8, provide the property information that you want to be different than the defaults. This includes object names, containers for the object, and working directories.

1.1.6 Determining the Distribution Flow

The following sections provide information for determining distribution routes:

- "Understanding Distribution Routes" on page 42
- "Selecting Subscribers for the Distribution Routes" on page 43
- "Configuring the Distribution Routes" on page 44

For more detailed information, see Section 3.3.2, "Understanding Distribution Routing," on page 97.

Understanding Distribution Routes

Each Distributor has a routing hierarchy that provides it with a hierarchical path for sending its Distributions. The routing hierarchy contains a list of Subscribers. The hierarchy of Subscribers can be many levels deep.

Subscribers in a Distributor's routing hierarchy do not need to also be recipients of the Distributions from that Distributor. A Subscriber can merely act as a proxy for the Distributor to pass Distributions to other Subscribers.

Not all Subscribers are needed in a routing hierarchy; only the ones used to pass Distributions on to other Subscriber servers. Most of your network's Subscriber servers will likely be end-node Subscribers; meaning, Subscribers that only receive and extract the Distributions.

The Distributor determines the most efficient route to any given Subscriber as follows:

1. The Distributor identifies the Subscriber that is to receive the Distribution.
2. The Distributor determines whether that Subscriber has a parent Subscriber.
3. If the Subscriber has a parent Subscriber, the Distributor checks its routing hierarchy for that parent Subscriber:
   a. If the parent Subscriber is in the routing hierarchy, the Distributor uses that route to send the Distribution to the Subscriber.
   b. If the parent Subscriber is not in the routing hierarchy, the Distributor sends the Distribution directly to the parent Subscriber of the end-node target Subscriber.
4. If the Subscriber does not have a parent Subscriber, the Distributor checks its routing hierarchy for the Subscriber:
   a. If the Subscriber is in the routing hierarchy, the Distributor uses that route to send the Distribution to the Subscriber.
   b. If the Subscriber is not in the routing hierarchy, the Distributor sends the Distribution directly to the Subscriber.

In other words, if the Distributor can find a way to send the Distribution using its routing hierarchy, it uses the path in that hierarchy to get the Distribution to the Subscriber. Otherwise, it sends the Distribution directly to the Subscriber (or its parent Subscriber).

For that reason, you should make sure every Subscriber that regularly receives Distributions from a Distributor has some connection to the Distributor's routing hierarchy. You can make this connection by listing a Subscriber in the hierarchy or by having one of the Subscribers in the hierarchy be its parent Subscriber.

You should generally not allow the Distributor to send Distributions over WAN links, except to such Subscribers that might be in the first tier of its routing hierarchy.

Consider the following in designing your Distributor's routing hierarchy:

◆ **End-node Subscribers:** The only Subscribers that you need to add to the routing hierarchy are those you want to be used to pass on Distributions. End-node Subscribers that only receive Distributions and not pass them on do not need to be added to the routing hierarchy.

◆ **Configuring distribution routes:** To create the distribution routes, consider your network design and the number of Subscribers on each LAN. Then design the routing hierarchy to mimic your network topology.

◆ **Selecting multiple Subscribers:** During hierarchy creation, you can place multiple Subscribers at the same tier under a single Distributor or Subscriber.

IMPORTANT: The most efficient routing hierarchy is to have more tiers and fewer Subscribers per tier, than just a few tiers with many Subscribers per tier. Therefore, select only a few Subscriber servers per tier. This minimizes the workload for the Distributor or Subscriber server that is sending Distributions to other Subscriber servers. Tiering helps to share the workload of sending Distributions throughout the network.

◆ **Using multiple Distributors:** Multiple Distributors can use the same routing hierarchy of Subscribers, so that the same distribution route can be used by each Distributor.

◆ **Reusing Subscribers:** You should consider whether you might overload a Subscriber server if it should be a parent Subscriber in a routing hierarchy that services multiple Distributors.

Selecting Subscribers for the Distribution Routes

The purpose of the Distributor's routing hierarchy is to create the most efficient method for distributing to Subscribers. You need to determine which servers are best suited to be Subscribers in a routing hierarchy, and how many servers to include in the hierarchy.

Select a server that is robust in its physical configuration. For example, a fast CPU, plenty of RAM, and plenty of free hard disk space (especially on volumes other than sys: on NetWare servers).

Use the following criteria to determine which Subscribers to include in a Distributor's routing hierarchy:

◆ Is the Subscriber needed to minimize the Distributor's workload?

◆ Do you need other Subscribers to share the workload of a parent Subscriber on a given LAN?

◆ Is the Subscriber needed to minimize network traffic (such as through WANs or firewalls)?

To identify the Subscriber servers to use in a Distributor's routing hierarchy, create a list of the servers in your network that you want to use as parent Subscribers in a Distributor's routing hierarchy.

To help minimize network traffic, select at least one server on each LAN.

Identify the server objects that you want to be parent Subscribers in the Distributors' routing hierarchies:

CONFIGURATION PLANNING WORKSHEET

Under item 16. provide the names (including full context) for your parent Subscriber servers.

Configuring the Distribution Routes

Specify the following information on your network diagram:

CONFIGURATION PLANNING DIAGRAM

Write "parent=1" next to every location on the diagram that is separated from the Distributor's location by a WAN link or firewall (unless there is only one Subscriber at that location).

For every location on the diagram that requires additional parent Subscribers because of the high number of Subscribers, change "parent=1" to "parent=#" where # is the number of parent Subscribers the site needs for load-balancing.

Also note whether you want to use one parent Subscriber in a given location as the primary parent Subscriber (the only one at that location in the Distributor's routing hierarchy) for receiving Distributions and passing them on to other parent Subscribers in that location.

Be sure to include parent Subscribers at the Distributor's location, if needed.

Using the information from your network diagram, design your Distributors' routing hierarchies using the Subscribers you have selected:

CONFIGURATION PLANNING WORKSHEET

Under item 15, create a hierarchy for each Distributor's routing hierarchy. You can reuse Subscriber servers in different Distributor's hierarchies.

1.1.7 Understanding Distribution Security

Server Management provides adequate security for Distributions that are sent within a secured network using certificates. However, Distributions could require additional security measures that are available in Server Management.

For more information about security, see Chapter 7, "Security in Policy and Distribution Services," on page 307.

Review the following to determine whether you need any additional security for your Distributions:

- ◆ "Determining Whether You Need Inter-Server Communications Security" on page 44
- ◆ "Determining Whether You Need Encryption Security for Windows Servers" on page 45

Determining Whether You Need Inter-Server Communications Security

Policy and Distribution Services uses XMLRPC (Extensible Markup Language Remote Procedure Call) for its normal inter-server communications. XMLRPC optionally provides security for communicating securely across non-secured connections.

Policy and Distribution Services can use this security for inter-server communications between servers across non-secured connections, or between a management workstation and servers across non-secured connections. For example, firewalls, intranets, NAT configurations, and so on.

This inter-server communications security ensures that data received across a non-secured connection is from a trusted source, that it has not been tampered with en route, and that the data received can be trusted by other machines. This is accomplished through the use of signed security certificates and digital signatures.

This security requires modifications to certain text files, and is installed using a Server Management wizard.

The following are instances when you could want inter-server communication security:

- **ConsoleOne administration:** When you use a workstation to manage a Distributor server across a non-secured connection.

- **SET parameters:** When you create a SET Parameter policy or a software package for SET parameters, inter-server communication takes place to provide the target server's SET parameter information. This communication could cross a non-secured connection.

- **Server Down policy:** When you use this policy to down a server, the communication between the downed server and another server watching for it to come back up could cross a non-secured connection.

For more information, see Section 7.3, "Security for Inter-Server Communication Across Non-Secured Connections," on page 321.

CONFIGURATION PLANNING WORKSHEET

Under item 13, provide the NetWare and Windows server names where you need to install the inter-server communications security software.

Determining Whether You Need Encryption Security for Windows Servers

You normally do not need to encrypt Distributions that are sent within your secured network. However, you can use encryption to provide security for when you send Distributions outside your network. The NICI software is used for encrypting Distributions.

For some NetWare servers, NICI 2.6 is automatically installed with the operating system. However, version 2.6.4 is supported in ZENworks 7 Server Management. You may need to upgrade your NetWare version of NICI. Version 2.6.4 is shipped with ZENworks 7, and is also shipped with ZENworks for Servers 3.0.2 (including version 3 SP2).

For Windows, Linux, and Solaris servers, you must install NICI 2.6.4 on the Distributor and Subscriber servers where you expect encrypted Distributions to be built and extracted.

IMPORTANT: If you have NICI 2.4.6 running on your network, it is optional whether you upgrade to NICI 2.6.4, because these versions are compatible with each other.

If you need to install the NICI software on a Windows, Linux, and Solaris server, you must also install that same version on all Distributor and Subscriber servers in your network. Encryption does not work correctly if there are two different versions of NICI installed in your network.

For information on Distribution encryption, see Section 7.2, "Distribution Security Using Encryption," on page 317.

CONFIGURATION PLANNING WORKSHEET

Under item 14, provide the Windows, Linux, and Solaris server names where you need to install the NICI software.

1.1.8 Determining the Channels for the Distributions

Channels are used to group Distributions, to establish a schedule for passing a Distributor's Distributions on to Subscribers, and to list the Subscribers that are subscribed to the Channel so that the Distributor knows where to physically send the Distribution files.

You can create a Channel for a specific Distribution usage (such as virus pattern files, operating system support packs, or policy packages), or for a specific Distribution time (such as off-peak Distributions).

You can associate a Channel with Distributions from many Distributors. A Channel can be subscribed to by many Subscribers.

Subscribers subscribe to Channels in order to receive certain Distributions. Distributors associate their Distributions with the Channels so that the subscribed Subscribers can receive those Distributions.

If you are installing multiple Distributors, they can share Channels for their Distributions. For example, if Distributor A and Distributor B both want to send some of their Distributions to the same set of Subscribers, one Channel can be used by both Distributors.

Channels are used in providing Distributions to Subscribers. Consider the following:

- A Channel is not owned by any particular Distributor
- Distributors associate their Distributions with the Channels
- A Channel can have Distributions from multiple Distributors
- A Channel can be used to group related Distributions
- A Channel's schedule determines when the listed Distributions are sent
- A Subscriber subscribes to one or more Channels to receive all of the Distributions listed in those Channels
- A Subscriber cannot select an individual Distribution from the several that could be listed in a Channel (it must receive all of the Channel's Distributions)

In naming Channels, use a descriptive method. For example:

```
VirusProtect
VProtectPatterns
VirusProtection
NW51patch4
NW6patch1
AUTOEXECNCF000326
```

You can manage your Channels more easily by:

- Using names that are purpose oriented
- Using a similar name for the Channel and its Distributions

CONFIGURATION PLANNING WORKSHEET

Under item 21, provide your Channel names. Make the names unique to help identify which Distributions they will hold.

You generally create a Channel for one or more related Distributions. However, for distribution flexibility, you can create one Channel for each application to be distributed.

CONFIGURATION PLANNING WORKSHEET

Under item 22, provide the Distributions that belong to each Channel.

For ease of management, plan to create the Channel objects in the same context as your other Tiered Electronic Distribution objects, especially the Distribution objects.

CONFIGURATION PLANNING WORKSHEET

Under item 20, provide the eDirectory context where the Channel object should be created.

1.1.9 Determining Subscribers' Subscriptions

You need to subscribe your Subscribers to Channels before they can receive their Distributions. This is done by subscribing a Subscriber or Subscriber Group to the Channel that is associated with the Distribution it needs:

- "Subscribers" on page 47
- "Subscriber Groups" on page 47

Subscribers

Because Subscribers do not access eDirectory, all configuration information in the Subscriber object's properties is pushed down to it from the configuring Distributor, if it is needed. This includes such information as working directory, log file level and location, console messaging level, variables, and so on.

Changes to a Subscriber object's properties are not in effect until the Distributor reads eDirectory again and sends a new Distribution with the configuration information down to the Subscriber.

For each Distribution, determine which Subscriber servers need a particular Distribution.

CONFIGURATION PLANNING WORKSHEET

Under item 24, provide the Channel name for a Distribution (see item 22) and list the Subscribers that need that Distribution. Repeat for each Channel you provided in item 21.

Subscriber Groups

A Subscriber Group is used for grouping Subscribers that have the same Distribution needs.

Subscriber Groups are useful when you are sending several different Distributions to the same set of Subscribers. There is no need to create a Subscriber Group if it is only associated with one Channel.

For example, Distribution A is in Channel A, Distribution B is in Channel B, and so on. Then, if you are not using a Subscriber Group, you need to subscribe each of your Subscribers to Channel A, then each to Channel B, and so on, which could be a very long process. However, by using a Subscriber Group, you only need to create the group, add the Subscribers to it, then subscribe that one group to each Channel.

Another use of a Subscriber Group is that when the group is associated with two or more Channels, you can edit the group's membership more easily than making the same changes in multiple Channels. For example, to remove a Subscriber from one Subscriber Group, you just edit that one group's properties. To remove that same Subscriber from several Channels, you need to edit each Channel's properties.

CONFIGURATION PLANNING WORKSHEET

Under item 17, provide a unique name for the Subscriber Group.

Under item 18, provide a list of Subscribers that need the same Distributions from the Channel (see item 21 and item 22) where the group is subscribed.

Under item 24, provide the Channel names for the Distributions that you want all of the Subscribers in the group to receive.

1.1.10 Determining the Distribution Schedules

Tiered Electronic Distribution has different schedules so that you can coordinate the various distribution processes. For more detailed information, see Chapter 8, "Scheduling," on page 325.

Review the following to plan your Tiered Electronic Distribution schedules:

- "Understanding Scheduling in Tiered Electronic Distribution" on page 48
- "Determining the Distributor's Refresh Schedule" on page 49
- "Determining the Distribution's Build Schedule" on page 49
- "Determining the Channel's Send Schedule" on page 49
- "Determining the Subscriber's Extract Schedule" on page 50

Understanding Scheduling in Tiered Electronic Distribution

Both Tiered Electronic Distribution objects and individual Server Policies can be scheduled.

Tiered Electronic Distribution uses schedules to control when Distributors are refreshed and Distributions are built, sent, and extracted. Schedules do not affect the total resources used by a Distribution, but rather *when* the resources are used.

Some policies must be scheduled before they can be enforced. If you enable a policy, but do not schedule it, it is activated according to the schedule currently specified in the Default Package Schedule, which provides a default for scheduled policies. The default schedule is Run At System Startup.

If you configure several policies with the same schedule, the order they are run depends on the time stamps created when you created the policies. Therefore, when you view a list of policies, the order they are listed is the order that they are run.

If you want to control the order that certain policies are run, you should stagger their schedules, rather than rely on the time stamps to determine when they run. Therefore, consider the Tiered Electronic Distribution schedules you select when scheduling your policies, so that you do not have undesirable overlap, or out-of-sequence events that could cause some scheduled items to fail.

Other issues you might need to understand:

- How time zones can affect scheduling
- How policy schedules are affected by distribution schedules
- How distribution schedules can be affected by Distributor and Subscriber servers' non-Server Management software usage
- How the Randomly Dispatch option can affect scheduling
- How the Active and Inactive object options for the Tiered Electronic Distribution objects can affect scheduling and distribution flow

Determining the Distributor's Refresh Schedule

The Distributor's Refresh schedule determines when the Distributor should read eDirectory for new Distribution and Channel objects, or for configuration changes to existing Distribution and Channel objects. Upon a Distributor refresh, when the Build schedule starts the Distributor rebuilds the Distributions that it discovers to be new or changed, then sends them when the Send schedule starts.

The Refresh schedule is set to Never by default, which is recommended because an infinite loop could be encountered if the Refresh frequency is shorter than the time it takes to complete the building or sending of a Distribuiton. Therefore, you should normally refresh a Distributor manually.

If you want to use a different schedule than Never for Refresh, be certain that when the Distributor is refreshed it is not going to be in the middle of building or sending a Distribution.

As an example of when you might want to change the Refresh schedule from Never, if you create or change your Distributions daily and do not need to build and send them immediately, you can set the Refresh schedule to 1:00 AM daily to have your new Distribution objects or changes found by the Distributor so that it can build and send them during off-peak hours according to the Build and Send schedules.

Determining the Distribution's Build Schedule

The Build schedule determines when a Distributor is requested to build the individual pieces that comprise the Distribution.

During configuration, you are instructed to set each Distribution's Build schedule to allow the Distribution to be sent immediately after building it.

Determining the Channel's Send Schedule

The Send schedule provides a window of time for when a Distributor can send its Distributions to the Subscribers.

During configuration, you set each Channel's Send schedule to an interval of every 5 minutes, meaning that the Distributor can send its Distributions at any of the 5-minute intervals when the Channel's schedule fires.

Determining the Subscriber's Extract Schedule

The Extract schedule determines when a Subscriber can extract its received Distribution.

Before a Subscriber can use a Distribution that is sent to it, it must first extract the Distribution. Therefore, you should set the Subscriber's Extract schedule before you send the Distributions.

Determine when you want the various Subscriber servers to be active extracting Distributions. Depending on a Distribution's size, it could be best to have Distributions extracted during off-peak hours. For information on scheduling issues involving time zones, see Section 8.2.5, "Scheduling Issues," on page 339, especially "Calculating Time Differences" on page 342.

CONFIGURATION PLANNING WORKSHEET

Under item 23, provide the Subscribers' extract schedules.

1.2 Configuring Your Distribution System

Use these sections in the following order:

1. "Installing Additional Distributors, Databases, and Subscribers" on page 50
2. "Setting Up Additional Distribution Security" on page 54
3. "Configuring the Distribution Flow" on page 55
4. "Creating the Distributions and Related Channels" on page 57
5. "Subscribing to the Distributions" on page 59
6. "Sending the Distributions" on page 60

1.2.1 Installing Additional Distributors, Databases, and Subscribers

When installing Policy and Distribution Services for the first time, you installed one Distributor with a database file. If you planned to install more Distributors or databases (see "Understanding Distributors" on page 95 and Section 10.2, "Determining How Many Databases You Need," on page 361), you should perform this installation now.

When installing Policy and Distribution Services for the first time, you might not have installed the Subscriber software to all of your servers. If you want to install the Subscriber software to more servers at this time, you should perform this installation now.

IMPORTANT: Any servers where you do not have the Subscriber software installed are not eligible to receive the Distributions you have planned to create and distribute at this time. However, when you install the Subscriber software to servers at a later date, you can subscribe them to existing Channels for receiving their Distributions.

To install additional Distributors, databases, and Subscriber software to more servers, do the following in order:

Preparing to Install

1 Make sure you have fulfilled all of the necessary requirements for your target Distributor and Subscriber servers.

For more information, see "Server Requirements" in the *Novell ZENworks 7 Server Management Installation Guide*.

2 Select the workstation you will use to install the Distributors and Subscribers.

3 If you have not already done so, log in to the eDirectory tree where you want to create the Server Management objects (worksheet item 1).

This should be the same tree where you extended the schema for ZENworks 7 Server Management.

You are automatically authenticated to all of the NetWare target servers in this tree during installation. You can select those servers, as well as servers in other trees or domains, for installing the Policy and Distribution Services software. However, this is the tree where all of the Server Management objects are installed for each of the selected servers.

4 Continue with "Starting the Installation Program" on page 51.

Starting the Installation Program

1 On the installation workstation, insert the *ZENworks 7 Server Management with Support Pack 1 Program* CD.

The startup screen is displayed. If the startup screen is not automatically displayed after inserting the CD, you can start it by running `winsetup.exe` at the root of the CD.

IMPORTANT: Installation from a CD in a remote server is not supported unless there is a drive mapped on the workstation to that remote server. For example, if you place the CD in a Windows server CD drive, then run the installation from a workstation, you must have a drive mapped on the workstation to the CD drive of that Windows server.

2 Select *Server Management*, then select *Policy-Enabled Server Management*.

This begins the installation program.

3 If you agree with the Software License Agreement, click *Accept > Next*.

4 On the Installation Type page, click *New Installation*, then click *Next*.

5 On the Installation Options page, make sure all three check boxes are selected.

6 On the eDirectory Tree for Creating Objects page, select the tree (worksheet item 1).

This is the tree where you initially created Server Management objects.

7 Continue with "Selecting and Configuring the Distributor and Subscriber Servers" on page 52.

Selecting and Configuring the Distributor and Subscriber Servers

1 On the Server Selection page, click *Add Server*.

2 Browse for and select the Distributor (worksheet item 2) and Subscriber (worksheet item 3) servers and click *OK*.

3 Configure each server listed on this page, then click *Next* to continue with the File Locations and Options page:

TIP: To quickly configure a specific role or set of roles for one or more servers, select the servers, right-click the selection, then select the role for the server. The options that apply to that role are automatically selected. Repeat for additional roles.

ZENworks Policy-Enabled Management Services

The following three options are all selected by default. If you want to install the Inventory Agent, you must also select to install the Policy and Distribution Server.

- ◆ **Policy and Distribution Services Server:** For each server that you want to be a Subscriber, select this check box.

For Tiered Electronic Distribution purposes, you can deselect the following:

Inventory Agents
Remote Management

Additional Options

The installation program detects whether these options are already installed on a target server and dims the option label. You can still select the check box to reinstall the component.

- ◆ **Distributor:** The Subscriber service is installed automatically to all target servers. Select this check box if you planned to make a Distributor server.

- ◆ **Server Management database:** This is the Policy and Distribution Services database that the Distributor logs to server (worksheet item 4). You should install it on the same server as the Distributor in order to minimize network traffic for database logging.

IMPORTANT: You can install the database to multiple servers per run of the installation program; however, you can only install one database per server. On the Database Settings page, you will be able to individually configure each database that is being installed. On the Database Logging page, you will identify which of the databases being installed is to be the one database for initial logging.

For Tiered Electronic Distribution purposes, you can deselect the following:

Inventory Database
Inventory Server
Inventory Proxy Server
ConsoleOne Snap-Ins

TIP: You can configure a group of selected servers with the same options by selecting the group and right-clicking the group. This displays the Custom Selection dialog box.

---

4 On the File Locations and Options page, do the following:

    4a For each Distributor server, edit the installation path if you do not want to use the default (worksheet item 5).

        If you want all Distributor servers to have the same installation path, select all of the servers, then edit the path.

    4b For each Subscriber server, edit the installation path if you do not want to use the default (worksheet item 6).

        If you want all Subscriber servers to have the same installation path, select all of the servers, then edit the path.

    4c To launch Policy and Distribution Services components on server startup, select the check box.

    4d To start services when the installation is finished, select the check box, then click Next.

5 On the Distributor Object Properties page, edit the properties as necessary (worksheet item 7), then click Next.

6 On the Subscriber Object Properties page, edit the properties as necessary (worksheet item 8), then click Next.

7 On the Database Settings page, do the following:

    7a Edit the database file's path if you do not want to use the default (worksheet item 9).

        Because the database file can become very large, we recommend that you change the default NetWare volume from sys: to another volume on that server.

    7b Edit the Database object's name, if desired (worksheet item 10).

    7c Change the Database object's container, if desired (worksheet item 11).

8 If you chose to install the Policy and Distribution Services database, the Log to a Server Management Database That Will Be Installed option is selected; click Next to display the Summary page.

9 Continue with .

Completing the Installation

1 To save the current installation configuration for future use in installing Distributors, on the Summary page select the *Save the following* check box.

2 Provide a path and filename for the template file.

If you attempt to quit the installation program without clicking Finish, you are prompted to save your current installation configuration to an installation template file.

You can reuse this template to speed up filling in installation pages in subsequent installations of Distributors or Subscribers.

3 Click *Finish* to begin the installation process.

4 After the installation program has finished, review the installation log file to determine whether any components failed to install.

The log file is located at:

`%TEMP%\_res`*number*`.txt`

where *number* is a three-digit number that is increased incrementally each time a new installation log is created.

5 If necessary, rerun the installation program.

Select only the components that failed to install.

6 Rerun the installation program once for each additional database that needs to be installed (worksheet item 4).

On the Server Selection page add only one of the Distributors where you planned to have a database installed, but have not installed it yet. Then, click only the Database column for that database's Distributor server and fill in the applicable information on the remaining installation pages.

7 To set up additional distribution security, continue with Section 1.2.2, "Setting Up Additional Distribution Security," on page 54; otherwise, continue with Section 1.2.3, "Configuring the Distribution Flow," on page 55.

1.2.2 Setting Up Additional Distribution Security

To ensure that you have the proper security for your Distributions, do the following tasks that are applicable:

- "Installing NICI 2.6.4" on page 54
- "Setting Up Inter-Server Communications Security" on page 55

Installing NICI 2.6.4

If you need Distribution encryption support for certain NetWare, Windows, Linux, or Solaris Subscriber servers, NICI 2.6.4 is supported in ZENworks 7 Server Management. If not, skip to "Configuring the Distribution Flow" on page 55.

If you previously updated your servers to NICI 2.6.4 using ZENworks for Servers 3 SP2, skip to "Configuring the Distribution Flow" on page 55.

IMPORTANT: All servers that are sending or receiving encrypted Distributions must be running the same version of NICI. Otherwise, encrypted Distributions to any of those servers will fail.

You must install NICI 2.6.4 to all Subscribers subscribed to the Channel that you select for the software package used to distribute NICI. NICI 2.6.4 must also be running on any Distributor server that creates encrypted Distributions.

However, if you already have NICI 2.4.6 installed, it is optional whether you upgrade to NICI 2.6.4, because these versions are compatible with each other.

A NICI update is contained on the *ZENworks 7 with Support Pack 1 Companion 2* CD, which is installed to Windows servers using the Novell International Cryptographic Infrastructure (NICI) menu option.

A software package update for NICI 2.6.4 is also provided on the *ZENworks 7 with Support Pack 1 Companion 2* CD.

When you install NICI 2.6.4, the installation program does not check to see if NICI is already installed.

Select the appropriate installation method:

- "Installing NICI on Windows Servers" on page 55
- "Installing NICI Using the Server Software Package" on page 55

Installing NICI on Windows Servers

To install NICI 2.6.4 on Windows servers:

1 On a Windows workstation, insert the *ZENworks 7 with Support Pack 1 Companion 2* CD.

2 Select the *Companion Programs and Files* option, then click *more >>* to access the *Companion 2* CD menu.

3 Select the *Novell International Cryptographic Infrastructure (NICI)* menu option.

4 Follow the installation instructions.

5 Continue with Section 1.2.3, "Configuring the Distribution Flow," on page 55.

Installing NICI Using the Server Software Package

To install NICI 2.6.4 on any supported server:

1 On a Windows workstation, insert the *ZENworks 7 with Support Pack 1 Companion 2* CD.

2 Copy the `nici265.exe` file from the `\NICI` directory on the CD to a location on your workstation, then extract the file.

3 Copy the `nici264.cpk` file that was extracted to a location on the Distributor server where you create the Software Package Distribution for installing NICI 2.6.4.

4 Create and send the Distribution to each Subscriber server where encrypted Distributions are received.

For information on creating and sending Software Package Distributions, see Section 3.4.4, "Creating a Distribution," on page 124.

5 Continue with Section 1.2.3, "Configuring the Distribution Flow," on page 55.

Setting Up Inter-Server Communications Security

If you are distributing to servers outside your secured network (worksheet item 13), see Section 7.3, "Security for Inter-Server Communication Across Non-Secured Connections," on page 321 for detailed instructions on setting up security for inter-server communications.

1.2.3 Configuring the Distribution Flow

You need to configure your distribution system to ensure the most efficient use of your network in sending Distributions by setting up the Distributors' routing hierarchies. This was not done for any Distributor when you installed Policy and Distribution Services.

To configure your distribution system:

- "Configuring the Distributor Routing Hierarchies" on page 56
- "Configuring Parent Subscribers" on page 56
- "Configuring Subscriber Groups" on page 57

Configuring the Distributor Routing Hierarchies

1 In ConsoleOne, right-click a Distributor object (worksheet item 2), then click *Properties*.

2 Select the *Routing* tab and do the following:

2a Click *Add* and browse for your first tier Subscriber servers (worksheet item 15), then click *Select > OK*.

This sets up your first tier of Subscriber servers. These receive Distributions directly from the Distributor.

2b Select one of the Subscriber servers in the first tier of the routing tree, click *Add* and browse for your next tier of Subscriber servers to go under that fist tier Subscriber (worksheet item 15), then click *Select > OK*.

This sets up a second tier of Subscriber servers for the one Subscriber that you selected. These second-tier Subscribers receive Distributions indirectly from the Distributor via the Subscriber server above them in the hierarchy.

2c Repeat Step 2b for each of the first-tier Subscribers until you have selected all of the second-tier Subscribers for this part of the hierarchy.

2d Select one of the Subscriber servers in the second tier of the routing tree, click *Add* and browse for your next tier of Subscriber servers to go under that Subscriber (worksheet item 15), then click *Select > OK*.

2e Repeat Step 2d for each of the second tier Subscribers until you have selected all of the third-tier Subscribers for this part of the hierarchy.

2f Repeat this process, tier by tier, until you have completed your planned routing hierarchy for the current Distributor.

3 Repeat Step 1 through Step 2 for your other Distributors.

4 When you have finished building the routing hierarchy, click *OK*.

5 Continue with "Configuring Parent Subscribers" on page 56.

Configuring Parent Subscribers

All Subscribers should not receive their Distributions directly from a Distributor. The Distributor's routing hierarchy provides a way to minimize the Distributor's workload in sending Distributions.

For Subscriber servers to receive their Distributions using the routing hierarchy, you need to identify a parent Subscriber that is in the routing hierarchy for each end-node Subscriber (the Subscriber to receive the Distribution). This allows an end-node Subscriber to receive its Distributions through the routing hierarchy, rather than directly from a Distributor.

A Subscriber that is in the Distributor's routing hierarchy does not need to have a parent Subscriber in order to receive a Distribution from that Distributor. Distributors check their routing hierarchies first, then check for parent Subscribers second.

To associate Subscribers with parent Subscribers:

1 In ConsoleOne, select a group of Subscriber objects for servers that you planned to have serviced by a particular parent Subscriber (worksheet item 16), right-click the selected group, click *Properties of multiple objects*, in the *Parent Subscriber* field browse for the parent Subscriber object, then click *OK > OK*.

Because you can do multiple editing of eDirectory objects, you can select all of the Subscribers that are serviced by one parent Subscriber and edit the Parent Subscriber field once for all of them.

2 Repeat this process for all end-node Subscribers.

3 Continue with "Configuring Subscriber Groups" on page 57.

Configuring Subscriber Groups

To create and populate a Subscriber Group:

1 In ConsoleOne, select the container to hold the Subscriber Group object, click *File > New > Object*, then select *TED Subscriber Group*.

2 In the New TED Subscriber Group dialog box, specify a *Subscribe Group* name (worksheet item 17), click *Define additional properties*, then click *OK*.

3 Click *General > Settings* and provide a description.

4 To populate the group with Subscribers, select the *Members* tab, then do the following:

> **4a** Click *Add*, browse for and select the Subscriber objects (worksheet item 18), then click *OK*.

> **4b** To remove any Subscribers from the list, select the Subscribers and click *Delete*.

> **4c** To view the properties of any Subscriber, select the Subscriber and click *Details*.

5 Click *OK* when you have finished configuring the Subscriber Group object.

6 Continue with Section 1.2.4, "Creating the Distributions and Related Channels," on page 57.

1.2.4 Creating the Distributions and Related Channels

The following are generic instructions for creating a Distribution. For more detailed instructions for most Distribution types, see Chapter 3, "Tiered Electronic Distribution," on page 85. For steps on using the Distribution Wizard to create a File or FTP Distribution, see Section 3.4.12, "Using the Distribution Wizard," on page 144.

You first need to create the Distribution, then create the Channel (if you don't use an existing Channel):

* "Creating and Configuring the Distribution" on page 57
* "Creating and Configuring the Channel" on page 59

Creating and Configuring the Distribution

1 In ConsoleOne, locate the containers where the Tiered Electronic Distribution objects were installed.

2 Right-click the container for Distributions, click *New > Object*, then select *TED Distribution*.

3 Specify a Distribution name (worksheet item 19).

Name the Distribution so you can identify what it contains.

4 Browse to and select the Distributor object to own this Distribution (worksheet item 19).

Each Distribution is associated with a single Distributor. That Distributor is responsible for building and sending the Distribution.

5 Select the *Define additional properties* check box.

6 Click *OK* to create the object.

The properties for the Distribution are now displayed.

7 Select the *Type* tab; in the Select Type drop-down box, select a Distribution type (worksheet item 19).

8 Configure the Distribution.

For information on configuring the different Distribution types, see Section 3.4, "Distributions," on page 111.

Use the up-arrow and down-arrow buttons to change the distribution order.

9 Select the *Schedule* tab.

The Distribution's schedule determines how often the Distributor attempts to build a new version of the Distribution. A new version is built only if there have been changes since the last version was built.

10 Select *Run Immediate* from the drop-down list.

This causes the Distributor to build the Distribution as soon as it reads eDirectory for the Distribution information.

11 Click *OK* at the bottom of the Distribution Properties dialog box to save all changes.

12 If you have not previously resolved certificates, for NetWare and Windows servers, select *Yes* when prompted to copy security certificates.

For Linux and Solaris servers, certificates must be resolved manually if you do not have a drive mapped to them. For more information, see Section 7.1.6, "Resolving Certificates," on page 311.

The Distributor needs to have been run at least once so that its certificates can be minted (created).

A Distributor needs to resolve its certificates only once per Subscriber.

The Subscriber software does not need to be running on the server for security certificates to be resolved. The server only needs to be up.

ConsoleOne sends security certificates to each Subscriber server that subscribes to the Channel that was selected in the Channel Tab. Each Subscriber must have a security certificate from the Distributor before it can receive Distributions from that Distributor.

It can take several minutes to copy a security certificate to each Subscriber.

IMPORTANT: Certificate copying only needs to be done once for each Distributor/Subscriber relationship.

13 If you receive an error when the Distributor tries to copy to a Windows Subscriber, enter the following for the path:

`\\IP_Address\zen$\pds\ted\security`

where *IP_Address* is the IP address of that Windows Subscriber.

14 If you receive an error when the Distributor tries to copy to a Linux or Solaris Subscriber, or you cannot browse for the Server to select it for resolving certificates, you must map a drive to the server (such as through using Samba), and then repeat resolving certificates.

15 Repeat these steps for any other Distributions you want to create at this time (worksheet item 19).

16 Continue with "Creating and Configuring the Channel" on page 59.

Creating and Configuring the Channel

Channel objects are used to associate Subscribers with Distributions. When Subscribers subscribe to a Channel, they receive all of the Distributions associated with that Channel. Each Channel has a schedule that determines when the Distributions associated with it are to be sent to the Subscribers.

1 In ConsoleOne, locate the container where the Channel objects reside (worksheet item 20).

This container should already exist.

We suggest for ease of management that you use the same OU for all Channels.

2 Right-click the Channel object's container, click *New > Object*, select *Channel*, then click *OK*.

3 Specify a name for the Channel (worksheet item 21) and click *OK*.

You could name your Channels according to the Distributions you intend for them. For example, Channel - Antivirus Update.

4 Right-click the new Channel object and click *Properties*.

5 Select the *Distributions* tab, click *Add*, browse for and select the Distributions for the Channel (worksheet item 22), then click *OK*.

This associates the Distributions with the Channel. The Subscribers that are subscribed to this Channel receive all of the Distributions currently listed there.

6 To set the Channel's Send schedule, select the *Schedule* tab, select *Interval*, specify the interval as every hour, then click *OK*.

7 Repeat Step 1 through Step 5 for each Channel you have planned (worksheet item 21).

8 Continue with Section 1.2.5, "Subscribing to the Distributions," on page 59.

1.2.5 Subscribing to the Distributions

- "Setting Subscribers' Extract Schedules" on page 59
- "Subscribing to the Channels" on page 60

Setting Subscribers' Extract Schedules

Before a Subscriber can use a Distribution that is sent to it via Tiered Electronic Distribution, it must extract the Distribution. Therefore, the Subscriber's extraction schedule must be set before sending the Distributions.

1 In ConsoleOne, right-click the Subscriber object (worksheet item 23) for a server where you want to set the extraction schedule, then click *Properties*.

2 Select the *Schedule* tab, click the arrow for the drop-down box, select *Run Immediately*, then click *OK*.

This causes the selected Subscriber to extract its Distributions as soon as they are received.

3 Repeat Step 1 and Step 2 as necessary until all Subscriber schedules have been set.

4 Continue with "Subscribing to the Channels" on page 60.

Subscribing to the Channels

Subscribers must subscribe to a Channel in order to receive the Distributions associated with that Channel. In the following steps, you will associate all of your Subscribers to the Channels created previously.

1 In ConsoleOne, right-click a Channel object (worksheet item 21) and click *Properties*.

2 Select the *Subscribers* tab, click *Add*, browse for each of the Subscriber or Subscriber Group (worksheet item 24) objects to be subscribed to this Channel, click *Select*, then click *OK*.

3 Select the *General* tab and make sure the *Active* check box is selected.

4 Click *OK* to close the Channel object's properties and save the changes.

5 Select *No* when prompted to copy security certificates.

6 Repeat Step 1 through Step 5 for each Channel (worksheet item 21).

7 Continue with Section 1.2.6, "Sending the Distributions," on page 60.

1.2.6 Sending the Distributions

Now that you have installed, created, and configured your Distributors, Subscribers, Channels, and Distributions, you can begin the Distribution process.

Do the following in order:

1. "Scheduling the Distribution and Refreshing the Distributor" on page 60

2. "Verifying That the Distribution Process Was Successful" on page 61

Scheduling the Distribution and Refreshing the Distributor

1 In ConsoleOne, right-click the Distributor object (worksheet item 2) and click *Properties*.

2 On the Distribution object's *Build Schedule* tab, click *Send Distribution immediately after building*, then click *OK* to close the properties.

The Distribution is sent as soon as it is built, regardless of the Channel's Send schedule.

3 Right-click the Distributor object and click *Refresh Distributor*.

This causes the Distributor to read eDirectory and obtain all of the changes that were made in eDirectory. The manual refresh of the Distributor is the recommended method. For more information, see "Determining the Distributor's Refresh Schedule" on page 49.

4 Continue with "Verifying That the Distribution Process Was Successful" on page 61.

Building the Distribution begins immediately (according to the Build schedule you set previously). The Distribution is sent within five minutes (according to the Send schedule you set previously).

As soon as the Subscribers receive the entire Distribution, they extract the contents to the Subscriber's working directory that you specified in the Subscriber object's properties.

Verifying That the Distribution Process Was Successful

There are a number of ways you can verify that your Distribution process has worked:

- **iManager:** The Tiered Distribution View and Subscriber Distribution View are the easiest methods for determining this information. For help on using those views, access the iManager Help on those pages.

- **Reporting:** Run a report on the Distribution to see its status. For information on Tiered Electronic Distribution reporting, see Chapter 11, "Reporting," on page 371.

- **Log files:** Depending on the logging levels you are using, you can review the log files for distribution statuses. The log files (`.log`) can be found in the Distributors' and Subscribers' working directories.

- **Distribution files:** Compare the Distribution file on the Distributor's file system (under `\zenworks\pds\ted\dist`) with the Subscriber's file system (under `\zenworks\pds\ted\sub\`*`individual_Distribution's_path`*) to see if it was received. The Distribution file uses the same name on both servers.

1.3 Managing Your Distribution System

Your Policy and Distribution Services system is now set up and ready for use. You can revisit Section 1.2, "Configuring Your Distribution System," on page 50 at any time and use the applicable sections to update your distribution system.

You can manage your distribution system using the ConsoleOne and iManager tools. There is some functionality in one tool that is not in the other. Generally, you can use ConsoleOne for installation and setup tasks, and iManager for management tasks. For more information, see Section 2.5, "Comparing the ZENworks Server Management Role in iManager with ConsoleOne Capabilities," on page 82.

For information on using ConsoleOne, see the following:

- Chapter 3, "Tiered Electronic Distribution," on page 85
- Chapter 4, "Server Policies," on page 197
- Chapter 5, "Server Software Packages," on page 243
- Chapter 6, "Desktop Application Distribution," on page 279
- Chapter 11, "Reporting," on page 371

For information on using iManager, see Chapter 2, "Novell iManager," on page 63.

Novell iManager

2

If you have not yet installed Novell® iManager, see"Management-Specific Workstation Requirements" in the *Novell ZENworks 7 Server Management Installation Guide*. ZENworks® 7 Server Management supports iManager 2.0.2 and 2.5/2.6. However, version 2.5 or later is required for Windows Server 2003.

The ZENworks Server Management role in iManager enables you to manage Server Policies and Tiered Electronic Distribution objects, agents, and processes from any location where the Web browser Internet Explorer 6 SP1 or later is available. The Server Management plug-ins to iManager only work in this browser. Other Web browsers are not supported in ZENworks 7.

Using the ZENworks Server Management role, you can:

* Create, modify, and delete Tiered Electronic Distribution objects (Distribution, Subscriber, Distributor, Channel, Subscriber Group, and External Subscriber).

* Create, modify, delete, distribute, and enforce policies and policy packages.

* View a graphical representation of your distribution system, which makes it easy to track a Distribution from Distributor to end-node Subscriber, no matter how many parent Subscribers the Distribution passes through.

* Display a browser-based console, called the Remote Web Console, for each Distributor Agent and Policy/Package Agent in your system. From the Remote Web Console, you can check the configuration of any agent, monitor the activities of any agent, and control many agent functions, such as forcing an action on a Distributor or Subscriber server to happen immediately, and monitoring the status of a Distribution or Subscriber.

The following sections help you make the most of the features available to you in the ZENworks Server Management role:

2.1 Accessing the ZENworks Server Management Role in iManager

Review the following sections to log in to iManager and to become familiar with ZENworks role in iManager:

2.1.1 Logging in to iManager

The steps to log in to iManager are different for versions 2.0.2 and 2.5/2.6, because version 2.0.2 is tree-dependent and version 2.5/2.6 is not.

To access iManager in your Web browser:

-
-

Logging in to iManager 2.0.2

1 If you are not logged in to the Novell eDirectory™ tree where Tiered Electronic Distribution objects are located, log in.

> **TIP:** If you are running iManager on a Windows server where the Novell Client™ is not installed, specify the IP address of a server where a replica of your eDirectory tree resides, instead of providing the tree name itself.

2 Access the following URL:

`http://`*server*`/nps/iManager.html`

where *server* is the IP address or DNS hostname of the server where iManager is installed.

The following dialog is displayed:

3 If the iManager login page does not appear, make sure that you entered the correct server designation and that you entered nps and `iManager.html` exactly as shown in the example, because it is case sensitive.

> **TIP:** You might need to use https instead of http.

4 Provide the username and password for the server that you identified in Step 2, then press Enter or click *Login*.
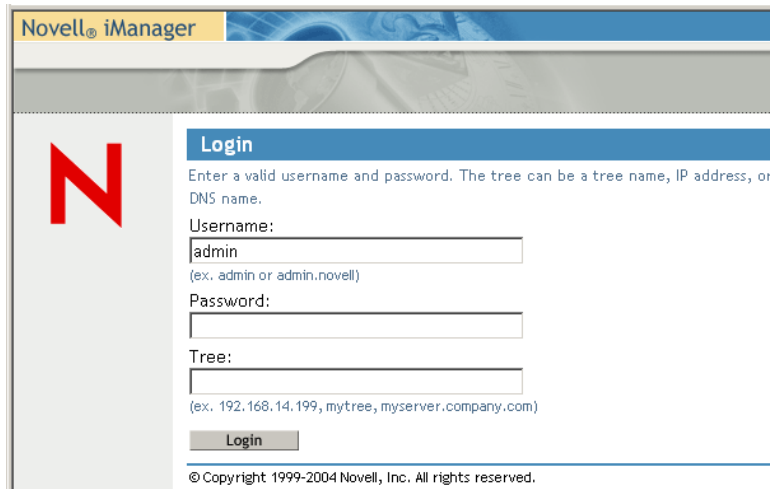
Logging in to iManager 2.5/2.6

1 Access the following URL:

`http://`*`server`*`/nps/iManager.html`

where *server* is the IP address or DNS hostname of the server where iManager is installed.

The following dialog is displayed:

2 If the iManager login page does not appear, make sure you entered the correct server designation and that you entered nps and `iManager.html` exactly as shown in the example, because it is case sensitive.

TIP: You might need to use https instead of http.

3 Provide the username and password for the server that you identified in .

4 Enter the tree designation for that server, then press Enter or click *Login*.

iManager 2.5/2.6 is not tree-dependent. Therefore, you can specify the tree during login, instead of before logging in (as in version 2.0.2), and can identify the tree using either the IP address of a server, a tree name, or the DNS name of a server.

2.1.2 Becoming Familiar with the Interface

1 After you successfully log in, the main iManager page is displayed. The top frame provides icons that represent its features:

2 Move the mouse pointer over the icons to review the purpose their functions.

The mouse-over text appears to the right of the row of icons.

By default, the *Roles and Tasks* icon is active, which is where the ZENworks functions reside.

2.1.3 Viewing the Roles and Tasks

1 Open iManager, then select the *Roles and Tasks* icon.

By default, the ZENworks Server Management role should be displayed in the left pane at the bottom of the tree structure.

2 In the left panel, expand *ZENworks Server Management* to list the available tasks:

```
⊟ ZENworks Server Management
      Create TED Object
      Delete TED Object
      Edit TED Object
      Remote Web Console
      Subscriber Distribution View
      Tiered Distribution View
```

These tasks provide the following functionalities:

Task	Functionality
Create TED Object	Create any Tiered Electronic Distribution object, except a Distributor or Subscriber.
Delete TED Object	Delete any Tiered Electronic Distribution object.
Edit TED Object	Edit the properties of any Tiered Electronic Distribution object.
Remote Web Console	Viewing and managing Tiered Electronic Distribution information or Server Policies information.
Subscriber Distribution View	Viewing and managing selected Subscribers and all of their Distributions.
Tiered Distribution View	Viewing and managing selected Distributions or Distributors and all of their Distributions.

3 If some of the above ZENworks Server Management tasks are not displayed, and you have Role-Based Services (RBS) configured, you might need to upgrade or reinstall the ZENworks Server Management module for the administrators who need access to the missing tasks.

For example, after upgrading the ZENworks Server Management plug-ins for iManager, if a new task was introduced by the upgrade, it will not be displayed for the RBS collections that are configured.

To solve this, follow the steps applicable to the version of iManager you are using:

- "Upgrading Collections in iManager 2.0.2" on page 67
- "Reinstalling Collections in iManager 2.5/2.6" on page 67

4 Continue with the task that you want to perform:

- Section 2.2, "Managing Tiered Electronic Distribution Objects," on page 67
- Section 2.3, "Monitoring the Distribution Process," on page 70
- Section 2.4, "Managing the Agents through Remote Web Console," on page 73

Upgrading Collections in iManager 2.0.2

1 Open iManager, then click the *Configure* icon.

2 Under *RBS Configuration*, click *Configure iManager*.

3 Select *Upgrade Collections*, then click *Next*.

4 Make sure the collections you want to upgrade are selected, then click *Next*.

Only the collections that have out-dated or previously not installed iManager modules are displayed.

5 Make sure that ZFSModule is selected, select the scope, then click *Start*.

6 Click *Close* after the module is shown to be successfully upgraded.

7 Click the *Roles and Tasks* icon.

The missing ZENworks Server Management roles should now be displayed under *ZENworks Server Management*.

8 Continue with .

Reinstalling Collections in iManager 2.5/2.6

1 Open iManager, then click the *Configure* icon.

2 Expand *Role Based Services*, then click *RBS Configuration*.

3 Under the *Name* column, select the desired collection to edit.

4 Under the *Name* column, select the *ZENworks Server Management* role.

5 Under the *Reinstall* column, click the check box for the listed ZFSModule name.

6 Click *Reinstall* (the column heading).

7 Click *OK* in response to the information dialog box to reinstall the module.

8 After the module is shown to be successfully reinstalled, click the *Roles and Tasks* icon.

The missing ZENworks Server Management roles should now be displayed under *ZENworks Server Management*.

9 Continue with .

2.2 Managing Tiered Electronic Distribution Objects

Acting in the ZENworks Server Management role in iManager, you can create, edit, or delete some of the following Tiered Electronic Distribution objects in eDirectory:

Distributor (cannot create)
Channel
Distribution
Subscriber (cannot create)
Subscriber Group
External Subscriber

For these Tiered Electronic Distribution objects, you can perform all of the same management tasks in iManager that you can perform in ConsoleOne®:
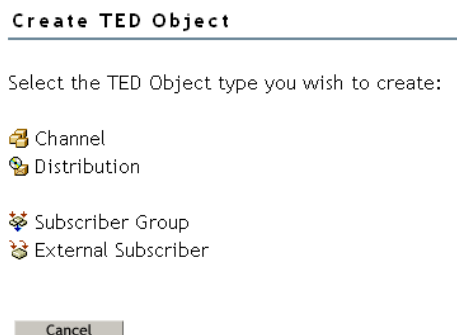
- Section 2.2.1, "Creating Tiered Electronic Distribution Objects in iManager," on page 68
- Section 2.2.2, "Editing Tiered Electronic Distribution Object Properties in iManager," on page 69
- Section 2.2.3, "Deleting Tiered Electronic Distribution Objects in iManager," on page 69

The following Policy and Distribution Services management tasks cannot be performed in iManager and must be performed using ConsoleOne:

- Managing the Server Management database. See Chapter 10, "ZENworks Database," on page 359
- Generating reports from the Server Management database. See Chapter 11, "Reporting," on page 371

2.2.1 Creating Tiered Electronic Distribution Objects in iManager

1 Open iManager, select *Roles and Tasks* in the top frame, expand *ZENworks Server Management* in the left frame, then click *Create TED object*.

Create TED Object

Select the TED Object type you wish to create:

Channel
Distribution

Subscriber Group
External Subscriber

[Cancel]

2 Select the type of object you want to create.

Any Distribution type you can create in ConsoleOne, you can also create in iManager.

3 Provide the information required for that object type, such as a unique name for the object, the context where you want to create the object, and so on.

Click the *Help* icon (question mark) for more information.

4 Click *OK* to finish creating the object.

5 Continue with Section 2.2.2, "Editing Tiered Electronic Distribution Object Properties in iManager," on page 69 to configure the new Tiered Electronic Distribution object.

2.2.2 Editing Tiered Electronic Distribution Object Properties in iManager

1 Open iManager, select *Roles and Tasks* in the top frame, expand *ZENworks Server Management* in the left frame, then click *Edit TED Object*.

2 Browse to and select the Tiered Electronic Distribution object whose properties you want to edit, then click *OK*.

The same property pages and options are available in iManager that are available in ConsoleOne.

You can click *Help* on each property page for information on setting the options.

3 Configure the object as needed, then click *OK* to save the new properties settings.

2.2.3 Deleting Tiered Electronic Distribution Objects in iManager

1 Open iManager, select *Roles and Tasks* in the top frame, expand *ZENworks Server Management* in the left frame, then click *Delete TED Object*.

2 Browse to and select one or more Tiered Electronic Distribution objects to delete, then click *OK* to list the objects on the Delete TED Objects page.

3 Click the *Help* icon for information about the repercussions of deleting specific types of objects from your distribution system.

4 Click *OK* to delete the listed objects, then click *OK* again to confirm.

5 Follow any instructions in the online help to reconfigure remaining objects so that the deletion does not disrupt your distribution system.

2.3 Monitoring the Distribution Process

The Tiered Distribution View enables you to track a Distribution from its Distributor through any parent Subscribers down to the end-node Subscriber. This helps you determine which Subscribers have received the Distribution, where they received it from, and when they received it. This, in turn, helps you troubleshoot and correct any problems that might occur during the distribution process.

The Subscriber Distribution View provides a status view of all Distributions for each Subscriber that you add to a watch list. You can use this view to troubleshoot a Subscriber's Distributions.

These capabilities are not available in ConsoleOne.

The following sections explain how to use these views:

- Section 2.3.1, "Using the Tiered Distribution View," on page 70
- Section 2.3.2, "Using the Subscriber Distribution View," on page 71

2.3.1 Using the Tiered Distribution View

To access the Tiered Distribution View in iManager:

1 Open iManager, select *Roles and Tasks* in the top frame, expand *ZENworks Server Management* in the left frame, then click *Tiered Distribution View*.

2 Browse to and select the Distribution you want to track, then click *Next*.

3 Select the Channel through which you want to track the Distribution, then click *Next*.

   The Distribution System window lists Subscribers that should receive the Distribution.

4 Click *Expand All* to display the routing hierarchy between the Distributor that built and sent the Distribution and the end-node Subscribers that should have received it.

   or

   Select an individual server to expand its part of the hierarchy.

   Icons indicate the status of the Distribution:

Icon	Meaning
	The Distribution has been received and extracted successfully.
	The Distribution has been received but not yet extracted. Check the Subscriber's extract schedule to see whether extraction has been attempted. If extraction was attempted and failed, check the Subscriber's event log to see what error occurred during extraction. See Section 2.4.2, "Managing the Distributor Agent," on page 77.
	The Distribution was not successfully received by the Subscriber. Check the Subscriber's event log for an error message describing the problem. See Section 2.4.2, "Managing the Distributor Agent," on page 77.
	The Distributor has not received any response from the Subscriber concerning the status of the Distribution. Check the status of the Subscriber and any parent Subscribers between it and the Distributor. See Section 2.4.2, "Managing the Distributor Agent," on page 77.

5 Fill in the time space in the *Refresh screen every __ seconds* field, then click *Start* to refresh the display at that frequency.

Only seconds can be entered.

This is useful for troubleshooting the distribution process as it occurs.

6 To display status information, select a Distributor or Subscriber, then click *Remote Web Console*.

For information about the types of status information you can obtain, see Section 2.4, "Managing the Agents through Remote Web Console," on page 73.

7 To check configuration information, select a Distributor or Subscriber, then click *eDirectory Configuration*.

You can edit the Distributor or Subscriber object properties just as if you had clicked *Edit TED Object* under *ZENworks Server Management*. The same property pages and options are available in iManager that are available in ConsoleOne.

2.3.2 Using the Subscriber Distribution View

To access the Subscriber Distribution View in iManager:

1 Open iManager, select *Roles and Tasks* in the top frame, expand *ZENworks Server Management* in the left frame, then click *Subscriber Distribution View*.

2 Click *Add* to select the Subscribers that you want to track.

Each Subscriber added does not initially display its Distributions.

3 Select an individual Subscriber to expand its Distribution list.

or

Click *Expand All* to display the Distributions for each displayed Subscriber.

Icons indicate the status of the Subscribers' Distributions:

Icon	Meaning
❌	The Distribution was not successfully received, or the extraction failed.
	Check the Subscriber's event log for an error message describing the problem. See Section 2.4.2, "Managing the Distributor Agent," on page 77.
⚪	The Distribution's status is not yet known, because the Distributor could not be contacted, or because the Distributor has not yet received the status from the Subscriber.
	Check the status of the Subscriber or the Distributor. See Section 2.4.2, "Managing the Distributor Agent," on page 77.
🟢	The Distribution was successfully received, but for a non-critical reason it has not yet been extracted, such as the Extract schedule has not yet started.
	Check the Subscriber's extract schedule to see whether extraction has been attempted.
🟢	The Distribution has been successfully received and extracted.

The Subscribers and Distributions are sorted by status, then alphabetically within a status. To display the more critical Distribution statuses first, the status order is:

1. Critical
2. Unknown
3. Received
4. Extracted

When one of these icons appear next to:

 * **"Subscriber" (root item in tree structure):** The status applies to one or more of the subordinate Distributions. Therefore, "Subscribers" shows the most critical icon of any status in the list.

 * **Subscriber icon:** The status applies to one or more of the Subscriber's Distributions. Therefore, the status icon shows the most critical status icon for any of the Subscriber's Distributions.

 * **Distribution icon:** The status only applies to this Distribution.

4 Mouse-over a Distributions to display the following information:

DNS Name

NDS Name

TED Version

Receive Status

Time Received

Should Extract (either True or False displays to indicate whether the Distribution is subscribed to by the Subscriber)

Time Extracted

Extraction Status

Distributor

Parent Subscriber

If you mouse-over a status icon, it displays a short sentence of the most critical status for the related object (Subscriber or Distribution).

5 Fill in the time space in the *Refresh screen every __ seconds* field, then click *Start* to refresh the display at that frequency.

Only seconds can be entered.

Click *Stop* to discontinue refreshing.

This is useful for determining whether a correction to a Distribution worked, or to troubleshoot the distribution process as it rolls out to different Subscribers.

6 To display status information, select a Subscriber, then click *Remote Web Console*.

This option does not apply to Distributions.

For information about the types of status information you can obtain, see Section 2.4, "Managing the Agents through Remote Web Console," on page 73.

7 To edit configuration information, select a Subscriber or Distribution, then click *eDirectory Configuration*.

You can edit the Subscriber or Distribution object properties just as if you had clicked *Edit TED Object* under the *ZENworks Server Management* role. The same property pages and options are available in iManager that are available in ConsoleOne.

2.4 Managing the Agents through Remote Web Console

On NetWare® servers, you can monitor the Distributor Agent and the Policy/Package Agent at the server console where they are running. In addition, you can monitor the agents running on any supported platform (NetWare, Windows, Linux, or Solaris) from Internet Explorer using the ZENworks Server Management role in iManager.

- Section 2.4.1, "Setting Up Passwords for Remote Web Console," on page 73
- Section 2.4.2, "Managing the Distributor Agent," on page 77
- Section 2.4.3, "Managing the Policy/Package Agent," on page 80
- Section 2.4.4, "Opening Multiple Remote Web Console Windows," on page 82

However, the following Policy and Distribution Services management tasks cannot be performed in iManager and must be performed using ConsoleOne:

- Creating, editing, and deleting policy packages. See Chapter 4, "Server Policies," on page 197.
- Creating, editing, and deleting software packages. See Chapter 5, "Server Software Packages," on page 243.

2.4.1 Setting Up Passwords for Remote Web Console

To secure the features provided by Remote Web Console, you can add a password in one of the following ways:

- "Adding a Password Using iManager" on page 73
- "Adding a Password by Editing the Zws.properties File" on page 75
- "Adding a Password Using a Distributed Server Package" on page 75
- "Removing Password Protection Using iManager" on page 76

Adding a Password Using iManager

1 Open iManager and click *ZENworks Server Management > Remote Web Console*.
2 In the *Display* field, select *Policy Package Agent*.
3 Click the *Actions* tab, then click *Set Password*.

The following is displayed:

ZENworks Server Management Web Console   [?]

Server: distributor-1nw.provo.novell.com   Display: Policy Package Agent [v]   Detach
View Services

| Configuration \ Policies \ Software Packages \ Schedule \ **Actions** |
Down Server | Refresh | Set Password

Setting a password for this server adds additional access control for Web
Console. Setting the password to a blank value will remove password protection.

Old Password: [   ]
New Password: [   ]
Confirm New Password: [   ]

[OK]

4 If a previous password exists, then enter it in the *Old Password* field; otherwise, leave the field blank.

IMPORTANT: Passwords are case sensitive.

5 Enter the new password twice, once in the *New Password* field and again in the *Confirm New Password* field, then click *OK*.

The following is displayed:

ZENworks Server Management Web Console   [?]

Server: distributor-1nw.provo.novell.com   Display: Policy Package Agent [v]   Detach
View Services

| Configuration \ Policies \ Software Packages \ Schedule \ **Actions** |
Down Server | Refresh | Set Password

Setting a password for this server adds additional access control for Web
Console. Setting the password to a blank value will remove password protection.

The password has be successfully set.

Old Password: [   ]
New Password: [   ]
Confirm New Password: [   ]

[OK]

You do not need to click *OK* again. Clicking OK does not exit the page; it only causes the entries in these two fields to be validated.

The password is requested the next time Remote Web Console is accessed, even without reopening iManager.

Adding a Password by Editing the Zws.properties File

1 Open `zws.properties` in a text editor that is appropriate for the following platforms:

Linux: `/etc/opt/novell/zenworks/zws.properties`

NetWare: *volume*`:\zenworks\zws\zws.properties`

Windows: *drive*`:\zenworks\zws\zws.properties`

2 Locate or add the following line:

`xmlrpcPassword=`

This line is case sensitive.

3 Either replace the old password or append your new password to this line.

The password is case sensitive.

4 Save and exit the `zws.properties` file.

The password is requested the next time Remote Web Console is accessed, even without reopening iManager.

Adding a Password Using a Distributed Server Package

1 In ConsoleOne, create a Distributed Server Package.

For more information, see Section 4.2.2, "Creating a Policy Package Object," on page 209.

Depending on the purpose of the policy package, provide a descriptive package name, such as "Distributed Server Package - New RWC Password" or "Distributed Server Package - Replace RWC Password."

2 Right-click the newly created Distributed Server Package object, then click *Properties*.

3 On the *Policies* tab, select the applicable platform, then click *Add*.

4 Select *Text File Changes*, type a name for the policy in the *Policy Name* field, then click *OK*.

The new policy should be displayed and selected. If not, select the check box in the *Enabled* column for the new policy.

5 Click *Properties*, then click *Add*.

6 In the New Text File Change dialog box, fill in the fields:

Filename: The name of the file to be edited by the policy, including its full path.

Change Description: Provide a short description of the change.

7 Depending on whether you are replacing an existing password or inserting the password for the first time, do one of the following:

- If you are creating a new password, fill in the fields:

  Change Mode: Select *Append to File* from the drop-down list.

  New String: Enter `xmlrpcPassword=`*your_new_password*, which provides this new line and password. Both the key and password are case sensitive.

- If yo are replacing an existing password, fill in the fields:

  Change Mode: Select *Search File* from the drop-down list.

  Search Type: Select *Start of Line* from the drop-down list.

Search String: Type the search string (the beginning of the line).

Case Sensitive: Select this check box to enable it.

Find All Occurrences: Select this check box to enable it.

Result Action: Select *Replace Line* from the drop-down list.

New String: Enter `xmlrpcPassword=`*`your_new_password`*, which fully replaces the existing line and password. Both the key and password are case sensitive.

8 For multiple platforms, repeat Step 3 through Step 7.

9 Click *OK* to save the changes.

10 Create a Distribution for this package and assign it to the Distributor.

For more information, see Section 3.4.4, "Creating a Distribution," on page 124.

The password is requested the next time Remote Web Console is accessed after this Distribution has been applied, even without reopening iManager.

Removing Password Protection Using iManager

1 Open iManager and click *ZENworks Server Management > Remote Web Console*.

2 In the *Display* field, select *Policy Package Agent*.

3 Click the *Actions* tab, then click *Set Password*.

The following is displayed:

4 In the *Old Password* field, enter the current password.

IMPORTANT: Passwords are case sensitive.

5 Make sure that both the *New Password* field and the *Confirm New Password* field are empty, then click *OK*.

The following is displayed:

You do not need to click *OK* again. Clicking OK does not exit the page; it only causes the entries in these two fields to be validated.

The password is no longer requested the next time Remote Web Console is accessed, even without reopening iManager.

2.4.2 Managing the Distributor Agent

To access the Remote Web Console for a Distributor or Subscriber:

1 Open iManager, select *Roles and Tasks* in the top frame, expand *ZENworks Server Management* in the left frame, then click *Remote Web Console* to display the following:

2 Specify the IP address or DNS hostname of a server where the Distributor Agent or Policy/Package Agent is running, then click *OK*.

or

Browse to and select a Distributor or Subscriber object or the Server object representing the server where the Distributor Agent is running, then click *OK*.

If you have passwords in effect, the following is displayed:

ZENworks Server Management Web Console

This server is password protected. Please enter the password to access Remote Web Console.

Password: [   ]

[OK] [Cancel]

3 Enter a valid Remote Web Console password.

The password is case sensitive.

4 Click *Help* on each Remote Web Console page for information on using the features available on that page.

In the *Display* field, *Tiered Electronic Distribution* is the default. The other option is *Policy Package Agent* (see Section 2.4.3, "Managing the Policy/Package Agent," on page 80).

Tabs at the top of the *Remote Web Console* frame provide various types of information related to the Policy and Distribution Services agents. Additional options are available on each tab.

Configuration | Distributions \ Channels \ Events \ Security
Configuration | Subordinate Configuration | Threads | Route to Subscriber

5 Continue with the task that you want to perform:

- "Managing Tiered Electronic Distribution Objects" on page 78
- "Monitoring Policy and Distribution Services Agent Status" on page 79
- "Monitoring Distribution Status" on page 79
- "Forcing Policy and Distribution Services Agent Actions" on page 80
- "Managing Security Certificates" on page 80

Table 2-1 through Table 2-5 summarize these tasks, give details for the Remote Web Console tab and option to use for each task, and indicate whether the task can also be performed using ConsoleOne.

Managing Tiered Electronic Distribution Objects

Table 2-1   *Policy and Distribution Services Agent Tasks*

Policy and Distribution Services Agent Management Task	Remote Web Console Tab and Option	ConsoleOne
List all object properties of Distributor and Subscriber objects in a single list	*Configuration > Configuration*	No
List the object properties of any subordinate Subscriber in the routing hierarchy	*Configuration > Subordinate Configuration*	No
List all object properties of Distribution objects (except type-specific information) in a single list	*Distributions > Distribution Information*	No

Policy and Distribution Services Agent Management Task	Remote Web Console Tab and Option	ConsoleOne
List all object properties of Channel objects in a single list	*Channels > Channel Information*	No
Display information about the Server Management database	*Configuration > Database*	Yes

If the Distributor has not been refreshed since changes were made to object properties in eDirectory, the object properties displayed in the Remote Web Console are different from the object properties displayed in ConsoleOne.The Remote Web Console displays object information from the point of view of the Distributor Agent.

Monitoring Policy and Distribution Services Agent Status

Table 2-2 *Monitoring Agent Status Tasks*

Policy and Distribution Services Agent Management Task	Remote Web Console Tab and Option	ConsoleOne
View and continuously refresh the current Distributor event log, complete with message severity levels	*Events > Distributor Event Log*	No
View and continuously refresh the current Subscriber event log, complete with message severity levels	*Events > Subscriber Event Log*	No
Display the current status of the various distribution threads started by the Policy and Distribution Services agents to perform their various functions	*Configuration > Threads*	No

Monitoring Distribution Status

Table 2-3 *Monitoring Distribution Status Tasks*

Policy and Distribution Services Agent Management Task	Remote Web Console Tab and Option	ConsoleOne
List all Distributions currently being processed by the Distributor or Subscriber, along with detailed status information	*Distributions > Active Distributions*	No
Display status information for a selected Distribution that has been received by a Subscriber	*Distributions > Received Distributions*	No
Display the route that a Distribution must take through the routing hierarchy from a Distributor or parent Subscriber to any subordinate Subscriber	*Configuration > Route to Subscriber*	No

Forcing Policy and Distribution Services Agent Actions

Table 2-4 *Forcing Agent Actions Tasks*

Policy and Distribution Services Agent Management Task	Remote Web Console Tab and Option	ConsoleOne
Immediately refresh a Distributor so that it reads eDirectory to check for modified Distributions	*Configuration > Refresh Distributor*	Yes
Immediately build a Distribution	*Distributions > Build Distribution*	Schedule dependent
Immediately send all Distributions listed in a selected Channel	*Channels > Distribute Channel*	Not with one click

Managing Security Certificates

Table 2-5 *Managing Security Certificates Tasks*

Policy and Distribution Services Agent Management Task	Remote Web Console Tab and Option	ConsoleOne
List the security certificates that are available on a Subscriber	*Security > Show Certificates*	No
Delete security certificates from a Subscriber	*Security > Show Certificates > Remove Certificate*	No
Have the Distributor sign Subscribers' Certificate Signing Request (`.csr`) files so that the Subscribers can receive encrypted Distributions from the Distributor	*Security > Sign CSR*	Yes

2.4.3 Managing the Policy/Package Agent

The Policy/Package Agent is responsible for installing the software and enforcing the policies that it receives and extracts. The Remote Web Console enables you to manage the Policy/Package Agent, which is not possible using ConsoleOne.

To access the Remote Web Console for a Policy/Package Agent:

1 Open iManager, select *Roles and Tasks* in the top frame, expand *ZENworks Server Management* in the left frame, then click *Remote Web Console* to display the following:

ZENworks Server Management Web Console

⦿ IP address or DNS name:

⦿ Distributor, Subscriber, or Server object name:

OK   Cancel

2 Specify the IP address or DNS hostname of a server where the Policy/Package Agent is running, then click *OK*.

or

Browse to and select a Subscriber object or the Server object representing the server where the Policy/Package Agent is running, then click *OK*.

3 If you have passwords in effect, the following is displayed:

Enter a valid Remote Web Console password.

The password is case sensitive.

4 In the *Display* field, select *Policy Package Agent* (*Tiered Electronic Distribution* is the default).

Click *Help* on each Remote Web Console page for information on using the features available on that page.

Tabs at the top of the *Remote Web Console* frame provide various types of information related to the Policy/Package Agent.

5 Continue with the task that you want to perform.

The following table summarizes these tasks and gives the *Remote Web Console* tab for each task.

Policy/Package Agent Management Task	Remote Web Console	ConsoleOne
List the plug-ins that are currently loaded for enforcing server policies	Configuration	No
List all the variables that the Policy/Package Agent has values for	Configuration	No
List all the policies that the Policy/Package Agent enforces on a Subscriber server	Policies	No
Immediately enforce one or more policies on a Subscriber server	Policies	No
Remove individual policies from a Subscriber server	Policies	No
Immediately refresh one or more policies so that the Distributor Agent reads eDirectory to check for modifications	Policies	No

Policy/Package Agent Management Task	Remote Web Console	ConsoleOne
List all the software packages that the Policy/Package Agent installs on the Subscriber server	Software Packages	No
Determine the current status of all software packages installed on the Subscriber server	Software Packages	No
Create and run a program or script on the Subscriber server once or repeatedly	Schedule	No
Down the Subscriber server	Actions	No
Restart the Policy/Package Agent	Actions	No

2.4.4 Opening Multiple Remote Web Console Windows

On any Remote Web Console page, click Detach in the upper right corner to display the current page in a new browser window. This enables you to access multiple Remote Web Console features at the same time. For example, you could detach one window for the Tiered Electronic Distribution agents and another window for the Policy/Package Agent. Or you could detach a window for the Remote Web Console and still be able to perform other ZENworks Server Management tasks in the main Novell iManager window.

2.5 Comparing the ZENworks Server Management Role in iManager with ConsoleOne Capabilities

Table 2-6 *Differences between iManager and ConsoleOne*

Task	iManager	ConsoleOne

Creating, editing, and deleting the following Tiered Electronic Distribution objects:
 	
    Distributor (cannot create)
    Subscriber (cannot create)
    Distribution
    Channel
    Subscriber Group
    External Subscriber
	Yes	Yes

Creating, editing, and deleting the following Policy and Distribution Services components:
 	
    Policy Package
    Server Software Package
    Desktop Application
	No	Yes

Task	iManager	ConsoleOne

Setting up the following Distribution types:
 	
    Desktop Application
    File
    FTP
    HTTP
    MSI
    Policy Package
    RPM
    Software Package
	Yes	Yes

Immediately refreshing a Distributor	Yes	Yes
Immediately building a Distribution	Yes	Not with one click
Immediately sending to Subscribers all Distributions listed in a Channel	Yes	Not with one click
Monitoring Policy and Distribution Services agent event logs and status	Yes	No
Listing and managing the policies on a Subscriber server	Yes	No
Listing and checking the status of software packages installed on a Subscriber server	Yes	No
Running programs and scripts on a Subscriber server	Yes	No
Downing a Subscriber server	Yes	No

Managing security certificates:
 	
    Listing available certificates
    Resolving certificates
    Signing CSRs
	
	Yes
	No
	Yes
		
		No
		Yes
		Yes

Managing the Policy/Package Agent	Yes	No

Tiered Electronic Distribution

<div style="text-align: right">3</div>

Novell® ZENworks® Server Management provides Tiered Electronic Distribution for managing distributions of files, policies, and software across your network.

Tiered Electronic Distribution is integrated with other Novell network management applications that snap in to the ConsoleOne® framework to take advantage of Novell eDirectory™ management and file access control. Tiered Electronic Distribution can also be managed using the ZENworks Server Management role in Novell iManager.

For information on Tiered Electronic Distribution, see the following sections:

3.1 Common Distribution Tasks

Table 3-1 through Table 3-6 provide documentation links to common Tiered Electronic Distribution tasks. All links are to sections in this Policy and Distribution Services portion of the *Administration* guide.

Tiered Electronic Distribution Objects

Table 3-1 *Common Tiered Electronic Distribution Tasks*

Task	Instructions
Create a Distributor or Subscriber	"Installation on NetWare and Windows Servers" in the *Novell ZENworks 7 Server Management Installation Guide*
Configure multiple Tiered Electronic Distribution objects	Section 3.9, "Configuring Multiple Tiered Electronic Distribution Objects," on page 167
Change the DNS name or IP address of a Tiered Electronic Distribution server	"Changing DNS Names or IP Addresses for Tiered Electronic Distribution Servers" on page 188

Distributor

Table 3-2 *Common Distributor Tasks*

Task	Instructions
Configure a Distributor object	"Configuring Distributors" on page 107
Create a routing hierarchy for a Distributor	"Understanding Distribution Routing" on page 97 and "Configuring Distributors" on page 107
Delete a Distributor object	"Deleting a Distributor Object and How Its Distributions Are Affected" on page 111
Refresh a Distributor	"Manually Refreshing the Distributor" on page 110
Create a security certificate on a Distributor and copy it to its associated Subscribers	"Creating Security Certificates for Non-Encrypted Distributions" on page 316

Distribution

Table 3-3 *Common Distribution Tasks*

Task	Instructions
Create a Distribution	Section 3.4, "Distributions," on page 111
Delete a Distribution	"Deleting a Distribution" on page 138
Managing orphaned Distributions (when their Distributor object has been deleted)	"Deleting a Distributor Object and How Its Distributions Are Affected" on page 111
Schedule and send a Distribution	Section 3.10, "Sending Distributions," on page 174
Force a Distribution to be sent	"Forcing a Single Distribution To Be Sent" on page 176
Use a parent Subscriber to send a Distribution	"Sending Distributions Through Parent Subscribers" on page 176
Send a Distribution to another tree	"Sending Distributions between Trees" on page 177
Import or export a Distribution manually	"Manually Importing and Exporting Distributions" on page 142
Create and send a File Distribution using a wizard	"Using the Distribution Wizard" on page 144

Channel

Table 3-4 *Common Channel Tasks*

Task	Instructions
Create a Channel	"Creating and Configuring Channels" on page 147
Force a Channel to fire	"Forcing a Channel To Be Sent" on page 148

Subscriber

Table 3-5 *Common Subscriber Tasks*

Task	Instructions
Configure a Subscriber object	"Configuring Subscribers" on page 151
Create an External Subscriber object	"Creating and Configuring External Subscribers" on page 166
Configure the `tednode.properties` file for a Subscriber server that does not have its own configuration capability	Section 3.13, "Editing the Tednode.properties File," on page 193

Network Traffic Management

Table 3-6 *Common Network Traffic Management Tasks*

Task	Instructions
Control bandwidth usage for Distribution traffic by setting the I/O rates	"Controlling I/O Rates and Concurrent Distributions" on page 185
Minimize network messaging traffic	"Minimizing Messaging Traffic" on page 186

3.2 Understanding Tiered Electronic Distribution

Review the following sections for an understanding of Tiered Electronic Distribution:

3.2.1 Distribution Management through Tiered Electronic Distribution

Tiered Electronic Distribution provides you with a way to manage your servers through the distribution of electronic data between servers. For example, application programs, collections of data files, software patches, and server policies.

When you install Policy and Distribution Services, the installation process creates Tiered Electronic Distribution and server policy objects in the eDirectory tree, copies software to the various servers, and sets up basic configurations for the Tiered Electronic Distribution and Server Policies components according to your installation selections.

The Tiered Electronic Distribution software can be hosted on NetWare®, Windows 2000, Windows 2003 Server, Linux, and Solaris servers.

Tiered Electronic Distribution uses a tiered distribution model that enables one server to indirectly service hundreds or even thousands of other servers. Tiered Electronic Distribution makes it easy to distribute files and policy packages by building them into compressed data files and hosting them in distribution channels for dissemination to the appropriate servers.

Tiered Electronic Distribution lets you schedule the distribution processes to take advantage of off-peak hours. It also sends notification of distribution status by sending e-mail messages, logging events, displaying real-time messages, database reporting, and sending SNMP traps.

Server Management can efficiently process (send/receive/extract) Distributions that are large in size and contain a substantial number of files, such as an entire 4GB volume with greater than 50,000 file entries.

3.2.2 The Basic Distribution Process

The Tiered Electronic Distribution distribution process is based on the creation of Distributions (compressed file collections) that you use to move files and policies to your network servers. For more information, see Section 3.10.1, "Understanding the Distribution Processes," on page 175.

Following is a simplified distribution process. It is governed by schedules that you set for each of the Tiered Electronic Distribution objects involved with the Distribution file.

1. A Distributor creates a security certificate to provide distribution security.

2. A Distribution is built on the Distributor server's file system according to the configuration you create in the Distribution object.

3. You associate the Distribution with a Channel.

4. You subscribe your target Subscriber servers to the Channel. This causes them to receive all of the Distributions contained in that Channel.

5. The certificate (from 1 above) is copied to Subscriber servers for Distribution security verification.

6. The Channel's listed Distributions are sent from the Distributor to the Subscriber servers whose security certificates are valid.

7. The Subscriber extracts the files or policies from the compressed Distribution file and applies them according to the Distribution object's configuration.

The schedules that you need to coordinate for sending Distributions are the Distributor's Refresh schedule, the Distribution's Build schedule, and the Channel's Send schedule. However, we recommend that you leave the Distributor's Refresh schedule set to the default of Never. For more information, see "Determining the Distributor's Refresh Schedule" on page 49.

The schedules that you need to coordinate for receiving and extracting Distributions are the Channel's Send schedule and the Subscriber's Extract schedule.

3.2.3 Tiered Electronic Distribution's eDirectory Objects

Tiered Electronic Distribution uses eDirectory objects and the related software for performing its distribution functions. The Distinguished Name (DN) of all Tiered Electronic Distribution objects includes the server name and component function of the host server.

The eDirectory schema extensions included in Tiered Electronic Distribution define the classes of eDirectory objects that are created in your eDirectory tree, including information that is required or optional at the time the object is created. Every object associated with Tiered Electronic Distribution in an eDirectory tree has a class defined for it in the tree's schema.

You will extend the schema of your tree for the eDirectory objects listed in Table 3-7 when you install ZENworks 7 Server Management:

Table 3-7   *Tiered Electronic Distribution eDirectory Objects*

Object	Basic Function	More Information
Distributor	Build, send Distributions	Section 3.3, "Distributors," on page 95
Distribution	Contain files, policies	Section 3.4, "Distributions," on page 111
Channel	List Distributions	Section 3.5, "Channels," on page 146
Subscriber	Receive, extract Distributions	Section 3.6, "Subscribers," on page 149
Subscriber Group	Channel subscriptions by multiple Subscribers	Section 3.7, "Subscriber Groups," on page 156
External Subscriber	Enable distributing between trees	Section 3.8, "External Subscribers," on page 158

3.2.4 Relationships of the Tiered Electronic Distribution Objects

Figure 3-1 illustrates the relationships of the main Tiered Electronic Distribution objects:

Figure 3-1   *The Distributor, Distribution, Channel, Subscriber, and External Subscriber Objects*
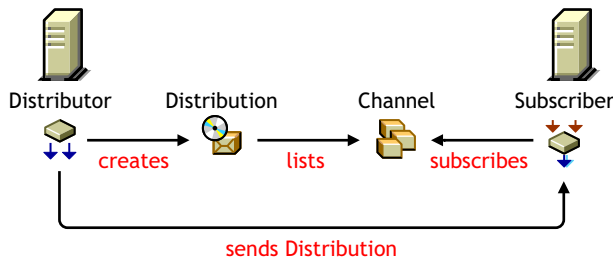
Note the following from this illustration:

- A Distributor creates a Distribution
- The Distribution is listed in a Channel
- A Subscriber subscribes to the Channel
- The Subscriber receives the Distribution from the Distributor (possibly via a parent Subscriber)

3.2.5 Physical Network Connections

Distributor and Subscriber servers can be physically connected to the network in any configuration, including having some servers across WAN links. The following describes the possible physical interactions between Distributor and Subscriber servers:

- A Subscriber server can be in the same geographic location as its Distributor server
- A Subscriber server can be in a different geographic location from its Distributor server, such as across a WAN link
- A Distributor server can service multiple Subscriber servers
- A Subscriber server can be serviced by multiple Distributor servers
- A Subscriber server can receive its Distribution files directly from a Distributor server
- A Subscriber server can receive its Distribution files indirectly via another Subscriber server acting as a parent Subscriber

3.2.6 Distribution Flow Details

Figure 3-2 illustrates the physical flow of Tiered Electronic Distribution Distributions:

Figure 3-2   *Tiered Electronic Distribution Flow*
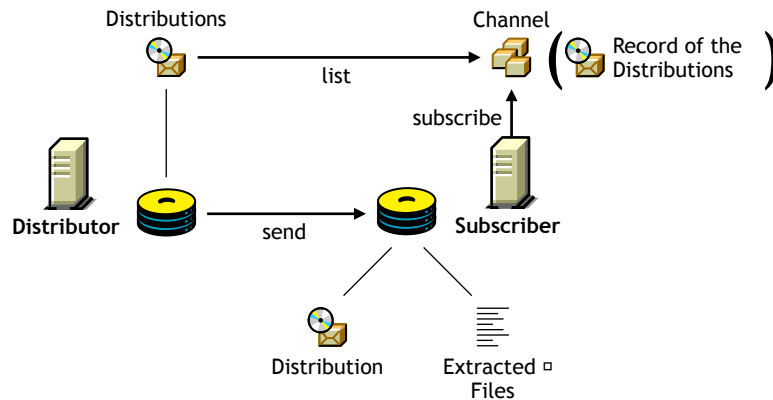
Note the following from the illustration:

- A Distribution file is stored on the Distributor server's hard drive
- The Channel lists a Distribution (it does not hold a copy of the Distribution)
- The Subscriber subscribes to a Channel to obtain all of the Distributions listed there
- The Subscriber extracts the Distribution contents from the file's compressed format and writes the content to the volume and directory specified in the Distribution's configuration

---

IMPORTANT: When there are multiple versions of a File or Desktop Application Distribution, the Subscriber maintains copies of each of the versions, as is specified in the Distribution object's properties. The default is to maintain 10 versions per Distribution type.

3.2.7 Tiered Electronic Distribution Processes

The following processes are used to perform Tiered Electronic Distribution functions:

- "Distributor Agent" on page 92
- "Policy/Package Agent" on page 92
- "Tiered Electronic Distribution Software Running on the Subscriber Server" on page 92
- "Distribution Processes Summary" on page 93

Distributor Agent

The Distributor Agent is installed on each server where you select the Distributor option during installation.

This agent has the following functions:

- Reads eDirectory for all Tiered Electronic Distribution configuration information (Distribution, Channel, and Subscriber) according to the Refresh schedule
- Builds Distributions based on the information contained in the Distribution objects that are associated with the Distributor
- Builds Distributions according to the Build schedule
- Sends Distributions according to the Send schedule
- Handles all notifications and events for the Subscriber
- Sends DS configuration information found in Subscriber objects to each Subscriber as part of each Distribution
- Logs Tiered Electronic Distribution information to the `ted.log` file for reporting purposes

Policy/Package Agent

The Policy/Package Agent is installed on each server where you selected the Policy and Distribution Server option during installation.

This agent has the following Tiered Electronic Distribution functions:

- Reads and enforces policy information that has been extracted from Policy Package Distributions

  For more information on policies, see Chapter 4, "Server Policies," on page 197.
- Installs Server Software Packages that have been extracted from Software Package Distributions

  For more information on software packages, see Chapter 5, "Server Software Packages," on page 243.
- Logs policy and software package information to the `zfs-startup.log` file for reporting purposes

Tiered Electronic Distribution Software Running on the Subscriber Server

Tiered Electronic Distribution software is installed on each server where you selected the Policy and Distribution Server option during installation.

This software has the following functions:

- Subscribes a Subscriber server to Channels for receiving Distributions
- Receives and extracts the following Distribution types to the server's file system according to the Extract schedule:

  Desktop Application [1]
  File
  FTP
  HTTP

MSI

Policy Package

RPM

Software Package

[1] The Desktop Application Distribution is only available when ZENworks Desktop Management is installed.

◆ Installs the following extracted Distributions:

Desktop Application

MSI

RPM

◆ In the parent Subscriber role, receives a Distribution and forwards it on to other Subscriber servers

Distribution Processes Summary

Table 3-8 *The Distribution Processes*

Function	Process	Explanation
Building and Sending Distributions	Distributor Agent	Discovers, builds, and sends all Distributions using the Distributor server's CPU and file system.
Extracting Distributions	Tiered Electronic Distribution software running on the Subscriber server	Extracts the Distribution's data onto the Subscriber server using the Subscriber server's CPU and file system. Also notifies the Policy/Package Agent when there are Server Policies to be enforced, or Server Software Packages to be installed.
Installing Distributed Software	Policy/Package Agent	Installs Server Software Packages onto the Subscriber server using the Subscriber server's CPU and file system.
Enforcing Installed Policies	Policy/Package Agent	Reads and enforces the extracted policies on the Subscriber server using its CPU and file system.

3.2.8 The Tiered Distribution Model

The power of the tiered distribution model is that you can spread the workload for sending Distributions. This is particularly important to the Distributor servers. By sharing distribution duties with parent Subscribers, a Distributor server can have more resources available for reading eDirectory, building each of its Distributions, and logging information to the database.

Tiered distribution levels can be very deep, providing a very large number of Subscribers that any one Distributor can service—without doing so directly.

Figure 3-3 illustrates a distribution routing hierarchy containing a Distributor, several parent Subscribers, and many end-node Subscribers:

Figure 3-3   *Distribution Route Hierarchy Showing Parent Subscribers and End-Node Subscribers*
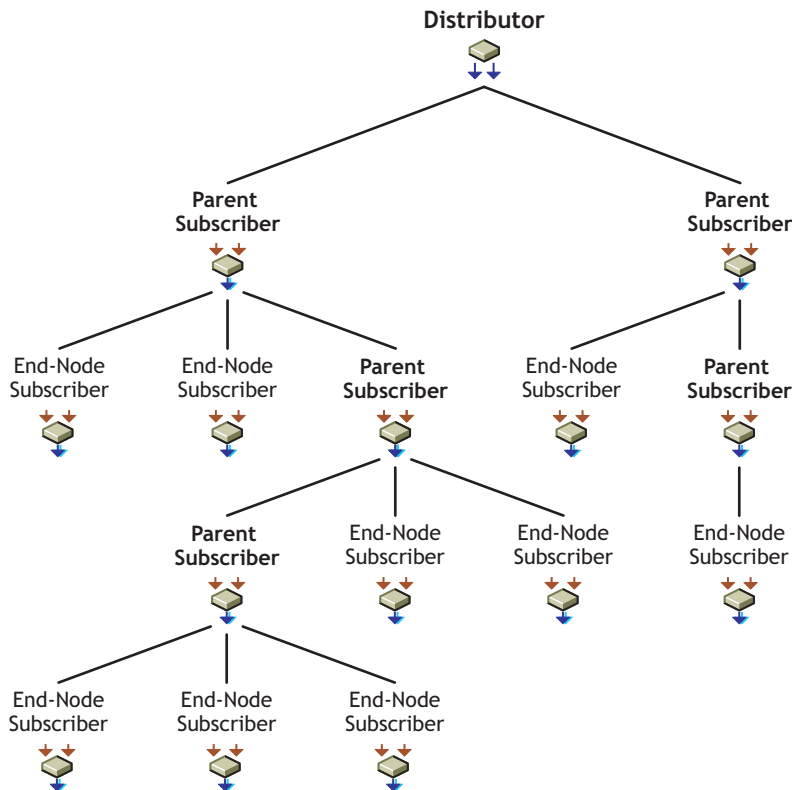
The Distributor can service hundreds of parent Subscribers directly, or service just a few first-tier parent Subscribers and let them do the bulk of the distribution work. In the above illustration, the Distributor only sends its Distribution to two parent Subscribers, yet nine end-node Subscribers receive the Distribution.

The parent Subscribers shown in this illustration can also receive the Distribution for extraction if they were also subscribed to the Distribution's Channel. If all of the parent Subscribers in the above illustration were subscribed to receive the Distribution being sent to the end-node Subscribers, the Distributor services 14 total Subscriber servers while itself sending the Distribution only twice.

Each parent Subscriber can service hundreds of other parent Subscribers or end-node Subscribers (the intended recipients of the Distributions). The workload for passing on a Distribution by a parent Subscriber is minimal in compared to the workload for the Distributor to build the Distribution.

As you can see, the tiered distribution model allows you to minimize the distribution workload for your Distributor servers.

3.2.9 Tiered Electronic Distribution's Key Components

In summary, the key components of Tiered Electronic Distribution include:

- eDirectory schema extensions that include objects for Distributors, Distributions, Channels, Subscribers, and External Subscribers
- ConsoleOne snap-ins and iManager plug-ins that provide creation, configuration, and management of Tiered Electronic Distribution
- A Distributor Java process hosted on a NetWare, Windows 2000, Windows 2003 Server, Linux, or Solaris server for handling distribution of data packages to Subscribers
- A Subscriber Java process hosted on a NetWare, Windows 2000, Windows 2003 Server, Linux, or Solaris server that subscribes to a Channel for its Distributions
- A routing hierarchy for each Distributor that has a hierarchical list of Subscribers who can both receive Distributions for themselves and pass the Distributions on to other Subscribers
- Parent Subscribers that pass Distributions on to other Subscribers
- An External Subscriber object that allows distributing between trees or to servers that do not have eDirectory server objects
- The Distributor Agent that controls the actual processes of building the Distribution files on the Distributor
- Policy/Package Agent that extracts and enforces policy information from Policy Package Distributions, and extracts and installs the contents of software packages
- Certificates that provide distribution security

3.3 Distributors

The following sections provide concepts and instructions for the Distributor object:

3.3.1 Understanding Distributors

The Distributor object (TED Distributor) is an eDirectory object that defines the properties for the Distributor.

- "Database Logging" on page 97

Functional Relationship with Other Tiered Electronic Distribution Objects

Figure 3-4 illustrates that a Distributor sends its Distributions to Subscriber servers:

Figure 3-4 *Distributor Sending to Multiple Subscribers*
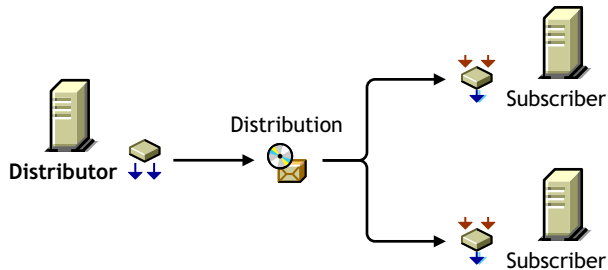
Figure 3-5 illustrates that a Distributor can list any one of its Distributions in several Channels, and several of its Distributions in one Channel:

Figure 3-5 *Distributor Listing Distributions in Multiple Channels*
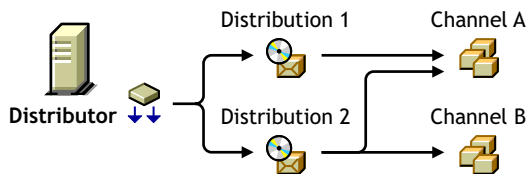
Distributor Description

The Distributor server's main Tiered Electronic Distribution function is to create and send Distributions. It also logs information to a database file, if you have one assigned for the Distributor.

The Distributor Agent builds a Distribution file on the Distributor server from the information you provide when you create and configure a Distribution object. A Distributor can own multiple Distributions.

When a Distributor builds a Distribution, it can optionally create a digest that provides an MD5 checksum for the Subscriber to compare against. Digests are used by Subscribers to verify that the Distributions have not been tampered with while in transit. Creating a digest is optional per Distributor, so the digests might not always be available for a checksum comparison by any Subscriber where this option is enabled.

Digests also detect corruption in a Distribution's package. In the case of corruption, the Subscriber renames the `distfile.ted` Distribution file to `distfile.corrupt` and the Distribution is rebuilt and sent the next time the Channel's schedule fires.

A Distributor lists its Distributions in Channels. Distributors do not own Channels. However, a Distributor is the sole owner of its Distributions.

The Distributor sends its Distributions to Subscribers (usually parent Subscribers for passing on the Distributions). If an end-node Subscriber does not respond to a Distributor (or a parent Subscriber) that is trying to send a Distribution to it, the Distributor retries sending a Distribution every two

minutes for 30 minutes, then stops. It does not attempt to re-send the Distribution until the Channel's Send schedule starts again.

Scheduling

A Distributor's Refresh schedule determines when it reads eDirectory for changes to its Distributions and other Tiered Electronic Distribution objects. A Distributor builds all new Distributions it finds and rebuilds any of its Distributions that have changed. The new or rebuilt Distributions are then available to be sent when a Channel's Send schedule starts.

IMPORTANT: We recommend that the Distributor's Refresh schedule be left at the default of Never, unless you have a reason to schedule the refresh. If you set the Refresh schedule, it is possilbe that the Distribution building and sending processes can be interrupted and restarted. This could possibly cause an infinite loop situation where the Distributions never get built or sent.

A Distributor can build its Distributions any time its Refresh schedule starts.

If you delete a Distribution, you should also refresh the Distributor immediately so that it recognizes the deletion and not try to build a Distribution that no longer exists. For information on deleting Distributions, see Section 3.4.8, "Deleting a Distribution," on page 138.

For information on scheduling, see Chapter 8, "Scheduling," on page 325.

Routing Distributions

The Distributor contains a distribution route, which is a hierarchical list of Subscribers that indicate the routes the Distributor can take to send its Distributions to its Subscriber servers. For information on routing hierarchies, see "Understanding Distribution Routes" on page 42.

Multiple Distributors in the Tree

You can have multiple Distributor objects in the tree; however, you can only have one Distributor installed per server. The need for multiple Distributors is dependent on several factors. For more information, see Section 1.1.4, "Are Additional Distributors Needed?," on page 38.

Database Logging

Individual Distributors can log information to their own database files, or all Distributors can log information to one common database file. For information on databases, see Chapter 10, "ZENworks Database," on page 359.

3.3.2 Understanding Distribution Routing

A distribution route represents the most efficient path to any given segment of your WAN. A distribution route is a list of parent Subscribers that relay Distributions on to other parent or end-node Subscribers. You can use Parent Subscribers to minimize the workload for a Distributor because they can pass on Distributions to other Subscribers.

The following sections explain how a Distributor moves its Distributions to your network's servers:

- ◆ "Understanding Parent Subscribers" on page 98
- ◆ "Understanding Routing Hierarchies" on page 100

Understanding Parent Subscribers

A parent Subscriber is a Subscriber that acts as a proxy for the Distributor to store and pass Distributions so that the Distributor does not need to send its Distributions directly to every Subscriber. Parent Subscriber servers do not need to be recipients themselves of a Distribution to temporarily store it for passing on to other Subscriber servers.

Distributors Send Distributions Using Parent Subscribers

A Distributor server must actually send each of its Distributions, because the Distribution files reside in its own file system.

Sending Distributions can create an enormous workload for a Distributor if it must individually send each of its Distributions to every Subscriber server on the network. Therefore, parent Subscribers are used to help send Distributions.

A detailed understanding of your network's topology is important for properly configuring distribution routes and selecting parent Subscribers. If necessary, create a diagram of your network showing all WAN links to determine how to use parent Subscribers.

Passing on Unsubscribed Distributions

A Subscriber does not need to subscribe to a Channel containing a Distributor's Distributions to be in the Distributor's routing hierarchy. A parent Subscriber itself does not need to be the recipient of the Distribution it is passing on.

Further, a parent Subscriber does not need to subscribe to the same Channels as its subordinate Subscribers to be able to pass on those Channel's Distributions.

Sharing the Distribution Load

In the illustration under "The Routing Hierarchy" on page 100, each Subscriber listed could be a parent to other Subscribers on its LAN. For example, if every Subscriber listed in the illustration was a parent to 20 end-node Subscribers, the Distributor could service 210 total Subscribers while only physically sending its Distributions to three of the Subscribers (the first-tier parent Subscribers, numbers 01, 04, and 09).

To further illustrate, parent Subscriber 04 would be servicing 104 Subscribers while only directly sending to two parent Subscribers (05 and 06) and its own 20 end-node Subscribers.

Balancing Workloads

A Distributor can use parent Subscribers in a routing hierarchy to explicitly determine routes for its Distributions. This eases its workload in distributing to Subscribers.

A parent Subscriber can also help a Distributor with its workload by acting as a proxy for the Distributor to pass on Distributions to other Subscribers. You can have multiple parent Subscribers on a given LAN to share the distribution workload on the LAN.

We estimate that the number of Subscribers and/or parent Subscribers that any one Distributor or parent Subscriber should service to be about 40. This figure is dependent on such factors as network speed, sizes of Distributions, and so on.

You should place parent Subscribers where they can help in load-balancing for Distributors and other parent Subscribers.

Figure 3-6 illustrates a WAN environment with parent Subscribers:

Figure 3-6 *WAN Environment with Parent Subscribers*
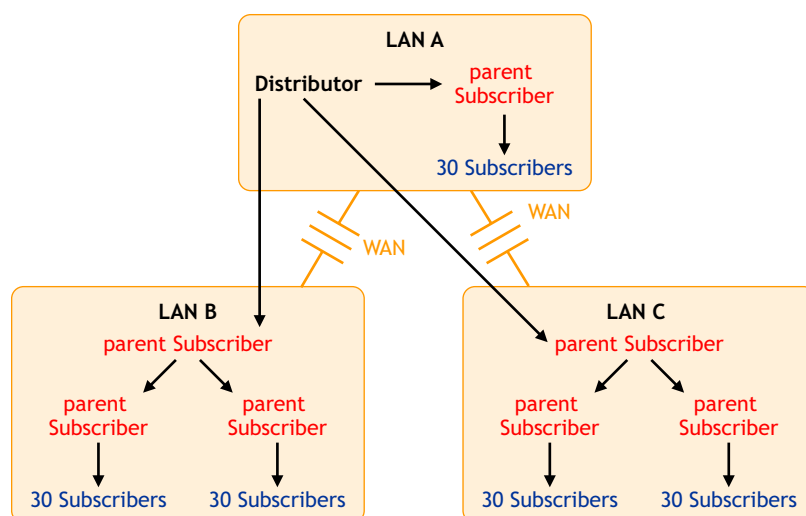
Note the following from this illustration:

- ◆ Assume that the three parent Subscribers that the Distributor's distribution lines point to are the first-tier Subscribers in the Distributor's routing hierarchy.

- ◆ Assume that the other four parent Subscribers (in LAN B and LAN C) are listed in the second tier of the distribution hierarchy.

- ◆ The Distributor does not need to send the Distributions directly to the 30 Subscribers on LAN A because the parent Subscriber in LAN A does that.

- ◆ The Distributor only sends its Distributions directly to the three parent Subscribers, but a total of 157 Subscribers can receive those Distributions.

- ◆ One parent Subscriber in LAN B (and the same for LAN C) was used solely for receiving Distributions directly from the Distributor, then passing them on to other parent Subscribers, which in turn passed them to their 60 Subscribers. For large systems, this scheme can make a parent Subscriber on the other side of a WAN link more available to a Distributor, instead of that parent Subscriber being so busy passing Distributions to its many other end-node

Subscribers that it can make the Distributor wait. Consider this hierarchical design where it might be applicable in your network.

The Distributor has the workload of reading eDirectory for Distribution changes, building the Distributions, sending the Distributions, and writing to the Server Management database. By minimizing the number of Subscribers that a Distributor itself must directly send Distributions to, you can give the Distributor more resources for its various functions.

Understanding Routing Hierarchies

Tiered Electronic Distribution provides a routing hierarchy to automate sending your Distributions from the Distributor servers to your Subscriber servers.

The Routing Hierarchy

To ease a Distributor's workload in sending Distributions, each Distributor has its own routing hierarchy, which is a hierarchical list of Subscribers that indicate the routes Distributions can take to send a Distribution to a Subscriber. The Subscribers in the routing hierarchy are the parent Subscribers. You can nest parent Subscribers many levels deep.

A parent Subscriber can receive a Distribution and extract it, as well as pass that same Distribution on to other Subscribers.

You can modify distribution routes at any time by editing the properties of the Distributor objects.

Figure 3-7 illustrates a Distributor's routing hierarchy:

Figure 3-7   *Distributor's Routing Hierarchy*

```
Distributor
├── Subscriber 01
│   ├── Subscriber 02
│   └── Subscriber 03
├── Subscriber 04
│   ├── Subscriber 05
│   │   └── Subscriber 06
│   │       └── Subscriber 07
│   └── Subscriber 08
└── Subscriber 09
    └── Subscriber 10
```
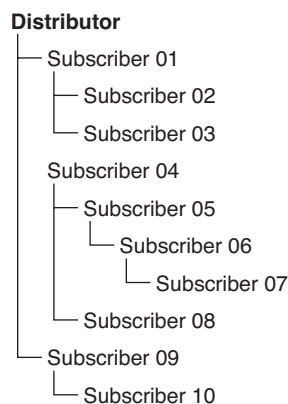
The only Subscribers you need to include in the Distributor's routing hierarchy are those that are used to pass on Distributions to other Subscribers. Subscribers that are not used to pass on Distributions are referred to as end-node Subscribers.

Distributing Using the Hierarchy

Assume that Subscriber 07 is a parent to Subscriber 11 (which is not in the routing hierarchy). The distribution route from the Distributor to Subscriber 11 would be as shown in Figure 3-8:

Figure 3-8 *Distributing Within the Hierarchy*
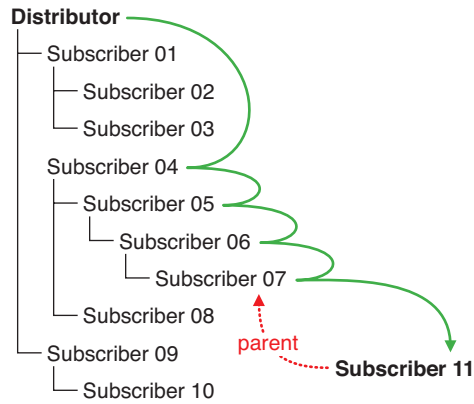
The Distributor used four parent Subscribers (04, 05, 06, and 07) to send the Distribution to Subscriber 11.

Subscribers Orphaned from the Routing Hierarchy

If Subscriber 11 did not have a parent Subscriber (such as Subscriber 07), the Distribution would come directly from the Distributor as shown in Figure 3-9:

Figure 3-9 *Subscribers Orphaned in the Distribution Hierarchy*
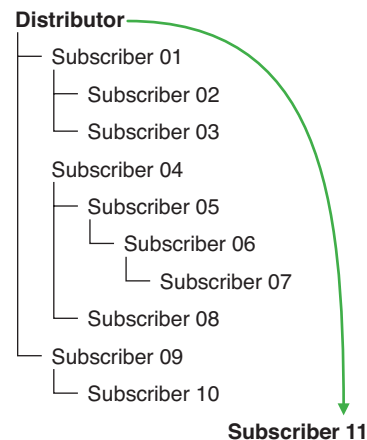
The only Subscribers you need to include in a routing hierarchy are those that are used to pass Distributions on to other Subscribers. The end-node Subscribers (Subscribers that are only receiving and not passing on Distributions) do not need to be listed in the hierarchy. They have links in eDirectory to their parents.

Subscribers that exist in a routing hierarchy are generally parent Subscribers, although this is not required.

IMPORTANT: Subscribers that do not utilize parent Subscribers can increase the workload on the Distributor and increase network traffic across WAN links. All Subscribers should have a parent Subscriber, except for the first tier Subscribers that receive Distributions directly from the Distributor.

Rerouting Because of Changes to the Routing Hierarchy

If a parent Subscriber is changed, or the routing list (on the Routing Hierarchy tab of the Distributor object's properties) is changed, the change is reflected in the routing slip (data file used in the distribution process), because it is calculated each time the Channel schedule starts. A refresh is required for the Distributor to read eDirectory and obtain the new routing hierarchy.

If a Subscriber server is removed from the network, and it was being used in a Distributor's routing hierarchy, you need to edit the Distributor object's properties to adjust the routing hierarchy because of that Subscriber's removal. Then refresh the Distributor so it can recognize the newer routing hierarchy.

Sharing Parent Subscribers with Other Distributors

If you have multiple Distributors, they can share portions of each other's distribution routes, meaning Subscribers can be listed in the distribution routing hierarchies of more than one Distributor. This is because the route to a Subscriber is dependent on the Distributor, and can be different for any given Distributor to Subscriber path.

Figure 3-10 illustrates the use of multiple Distributors and parent Subscribers in sending Distributions:

Figure 3-10 *Using Multiple Distributors and Parent Subscribers*
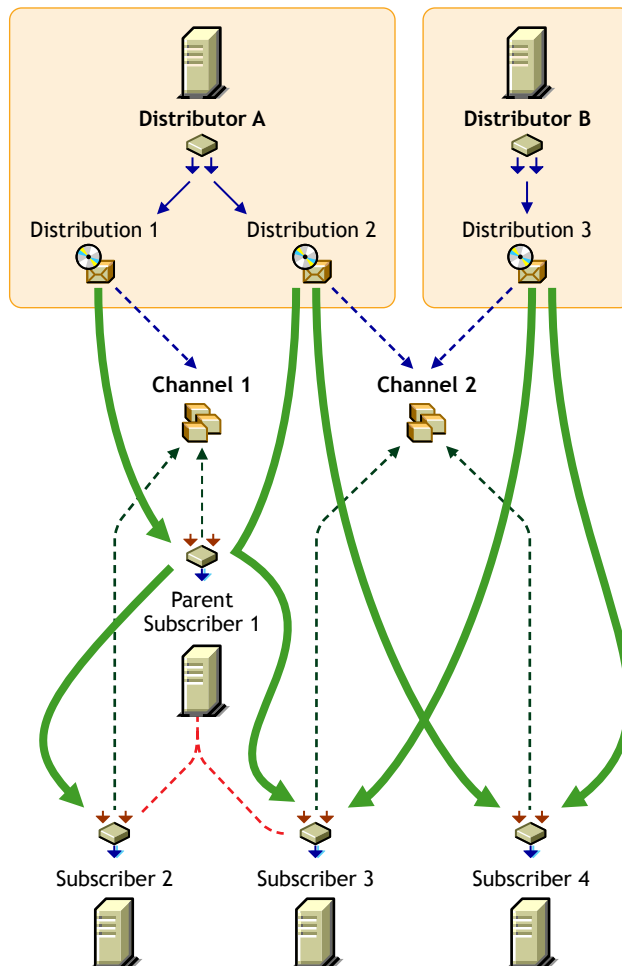
The arrows and lines indicate the subscription and Distribution connections to the Channels (dotted lines) and the distribution paths from the Distributors to the Subscribers (solid lines).

Figure 3-10 does not show distribution route hierarchies. For the purpose of this illustration, assume the following:

- Subscriber 1 is in Distributor A's hierarchy
- Subscriber 1 is a parent to Subscribers 2 and 3
- Subscribers 3 and 4 are in Distributor B's hierarchy
- Subscriber 4 is not in Distributor A's hierarchy

Note the following from the illustration concerning the use of multiple Distributors and parent Subscribers in sending Distributions:

- **Distribution ownership:** Distributors have ownership of their own Distributions and build and send each of their Distributions.

- **Multiple Distributors:** Multiple Distributors can list their Distributions in the same Channel. This means a Subscriber can receive Distributions from multiple Distributors.

- **Channel usage by Distributors:** Distributors can list their Distributions in any Channel, and they can list one Distribution in multiple Channels.

- **Multiple Distributions per Channel:** A Channel can have multiple Distributions from one or more Distributors.

- **Channel subscriptions:** Each Subscriber subscribes to any of the Channels that have the Distributions it needs. A Subscriber can subscribe to multiple Channels, and a Channel can have multiple Subscribers subscribed to it.

- **Parent Subscribers:** A parent Subscriber is used as a proxy for the Distributor to pass on Distributions to other Subscribers.

- **Orphaned Subscribers:** If a Subscriber is not in a Distributor's distribution route, or the child of a parent Subscriber in that hierarchy, the Distributor sends the Distribution directly to the Subscriber. This can be an issue for WAN links and other topology issues.

Distributing Across WAN Links

When you include parent Subscribers in the routing hierarchy, this can minimize network traffic by limiting the number of times a Distributor needs to pass a Distribution across a WAN link.

Because Distributors can send Distributions to parent Subscribers, which in turn can send them on to other Subscribers, a way is provided to send Distributions over a WAN link just once, instead of many times to reach every Subscriber on the other side of the WAN link.

Generally, you should have at least one parent Subscriber on every LAN to minimize the number of times a Distribution needs to cross a WAN link. Even if there are only two Subscribers on a LAN, you can reduce network traffic by using one of them as the parent Subscriber to the other.

Parent Subscribers are especially helpful with slow WAN links.

Consider the following when you determine how to distribute across your WAN links:

- **Parent Subscribers on the Distributor's LAN segment:** You should assign at least one Subscriber to be a parent Subscriber for all of the other Subscribers on a Distributor's LAN segment. That way the Distributor can have more resources for sending Distributions across WAN links.

- **Parent Subscribers for bridging WAN links:** You can minimize the number of Subscribers that a Distributor must directly service across WAN links by assigning at least one parent Subscriber on all other LAN segments and including those parent Subscribers in the Distributor's routing hierarchy.

  For example, your WAN has four LANs. With the Distributor in one LAN segment, it must send Distributions across three WAN links to get to Subscribers on the other three LAN segments. Let's assume each of the other LANs has 160 Subscribers that all need a Distribution from the Distributor. Without using parent Subscribers in the Distributor's routing hierarchy, the Distributor would need to send the Distribution 480 times across WAN links. In using

parent Subscribers (four per LAN segment to share the Distribution workload on the LAN), the Distributor would only need to send the Distribution nine times.

◆ **Primary parent Subscribers on a LAN:** You can further minimize WAN traffic by tiering parent Subscribers on the other side of a WAN link from the Distributor. In other words, you can have just one parent Subscriber in the routing hierarchy that would also be a parent to several other parent Subscribers on its LAN segment.

Using Figure 3-10 as an example, Subscriber 1 on each LAN segment could be the parent Subscriber for Subscribers 2, 3 and 4. In turn, parent Subscribers 1, 2, 3, and 4 would each service their own Subscribers. That would allow the Distributor to just pass a Distribution across a WAN link once to Subscriber 1, which would take care of passing that Distribution on to the other three parent Subscribers, saving the Distributor three extra WAN link transmissions. Therefore, in contrast to Figure 3-10, the 9 transmissions would be paired down to only three.

Out-of-Tree Distributions

To use Policy and Distribution Services in multiple trees, you must install the software separately in each tree. However, you only need to install the Server Management objects to one of the trees.

For example, if your network topology is as shown in Figure 3-11:

Figure 3-11 *Network Topology with One Distributor*
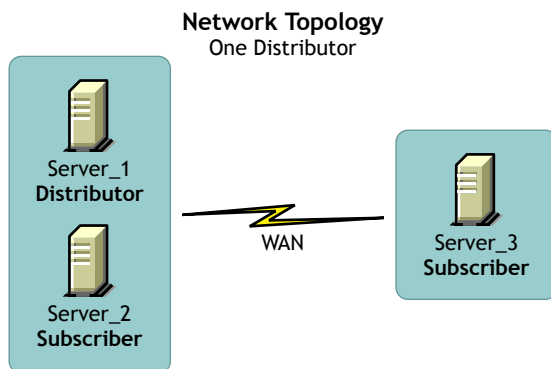
You could have the Tiered Electronic Distribution configuration shown in Figure 3-12 for the Distributor's routing hierarchy:

Figure 3-12   Distribution Flow in One Tree

Distribution Flow
One Tree-One Distributor

Tree_A

Server_1
Distributor_A

Server_2
Subscriber_1
(Parent to Subscriber)

Server_3
Subscriber_2
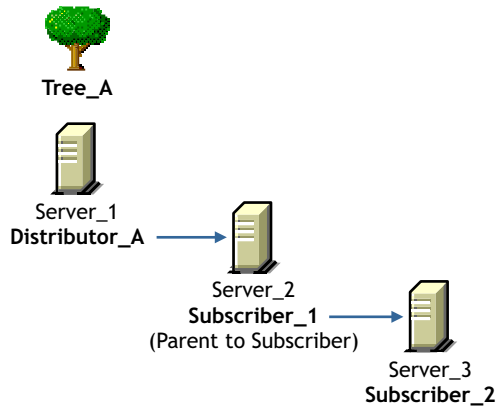
In this example, the Subscriber and server objects all exist in Tree_A. This allows you to have centralized management of the Tiered Electronic Distribution objects, regardless of your network topology.

Although you can create the Distributor and Subscriber objects in only one tree, you can install the Policy and Distribution Services software to any server in your network, whether the server's eDirectory object resides in the same tree where the Tiered Electronic Distribution objects are created, or the server does not have an eDirectory server object in any tree (such as a Windows server in a domain). This allows you to have centralized management of Tiered Electronic Distribution in environments where you have multiple trees and mixed server operating systems (such as NetWare and Windows servers).

For information on how External Subscribers are used for sending Distributions between trees, see Section 3.10.4, "Sending Distributions between Trees," on page 177.

Routing Hierarchy Configuration Guidelines

You should place parent Subscribers in the routing hierarchy using the following guidelines:

- Include at least one parent Subscriber on each LAN segment to minimize WAN traffic
- Include multiple parent Subscribers on each LAN that has 40 or more Subscribers to minimize a parent Subscriber's workload
- Make sure that every Subscriber that is not included in a Distributor's distribution route is assigned to a parent Subscriber on its LAN

Parent Subscribers are not always required for a WAN link. For example, if you have only two Subscribers on a LAN connected by a fast WAN link, the traffic difference between sending the Distribution once versus twice could be negligible. However, for a slow WAN link this might not be the case.

The factors in determining whether a Subscriber can receive Distributions directly from the Distributor instead of through a parent Subscriber are:

- Network connections

  For example, are you distributing:

  - only within a LAN?
  - across a slow or fast WAN?
  - across firewalls?
  - in a NAT?

- Frequency of sending Distributions
- Size of the Distributions

3.3.3 Creating Distributors

By understanding your network's topology, your Distributions (how many, their sizes, and how often you might expect them to be rebuilt), and how many Subscribers receive the various Distributions, you can determine how many Distributors you need.

Distributors must be created by installing their software and eDirectory objects using the *ZENworks 7 Server Management with Support Pack 1 Program* CD. For more information, see "Policy-Enabled Server Management Installation" in the *Novell ZENworks 7 Server Management Installation Guide*.

To determine whether you need multiple Distributors, see Section 1.1.4, "Are Additional Distributors Needed?," on page 38.

3.3.4 Configuring Distributors

Distributor objects are automatically created when the Distributor's software is installed to a server. You can edit your Distributor object's properties at any time.

Not all properties associated with the Distributor object are required. Required properties are noted in the following steps; all others are optional.

1 In ConsoleOne, right-click the Distributor object, then click *Properties*.

2 Click *General > Settings* and fill in the following fields:

Use policy: Select to use the effective policy if you want to use the values set in the Tiered Electronic Distribution policy. This field is displayed if a Tiered Electronic Distribution policy has been created, distributed to the Distributor server, extracted by the Policy/Package Agent, and enforced on the server. If you select this option, the rest of the fields are dimmed and the policy settings are used instead.

Input rate: The rate Distributions received (for its Subscriber). The default is the maximum that the connection can handle. This rate is used to control a Distributor server's use of narrow bandwidth links.

Output rates based upon Distribution's priority: Sets the default output rate to minimize network traffic for Tiered Electronic Distribution objects. This determines the send rate for Distributors. The default value is the maximum that the connection can handle. Blank means that bandwidth is taken from third-party software.

There are three output priorities where you can specify a rate:

- **High priority:** These Distributions are sent before any Medium or Low priority Distributions.
- **Medium priority:** These Distributions are sent after all High priority and before any Low priority Distributions.
- **Low priority:** These Distributions are sent after all High and Medium priority Distributions.

For more information, see Section 3.4.5, "Prioritizing Distributions," on page 127.

Maximum concurrent Distributions to build: Specifies the maximum number of distribution threads that can be running concurrently for building Distributions. The default value is 5. Valid values are from 1 to 10.

This number can help in load-balancing a Distributor's building activity.

Maximum concurrent Distributions to send: Specifies the maximum number of distribution threads that can be running concurrently for sending Distributions. The default value is unlimited (a blank field).

This number can help in load-balancing a Distributor's sending activity and spread network traffic over an entire scheduling window.

Connection time-out: Specifies the allotment of time before the Distributor server times out when connecting to another node. The default value is 300 seconds (five minutes), after which it ends the connection and does not retry until the send schedule starts again. The available range in seconds is 1 to 60,000.

You can increase or decrease this setting to allow messages to pass back and forth between the agents during the distribution process. If one node is expecting to receive a message from another, there should be a reasonable time to wait before assuming that the sender is no longer available.

IMPORTANT: This interval must be increased on slow or busy links where longer delays are frequent.

Working directory: Specifies the directory to be used by the Distribution. It contains Distributions, persistent status, and temporary working files. The default path is:

- **NetWare:** `sys:\zenworks\pds\ted\dist`

  IMPORTANT: The default volume is sys: on NetWare servers. We recommend that you do not use the sys: volume because the directory's content can become quite large.

- **Windows:** `c:\zenworks\pds\ted\dist`
- **Linux and Solaris:** `/var/opt/zenworks/zfs/pds/ted/dist`

The Distributor's working directory is also used whenever a Distribution is created. A directory is created under the working directory using the DN of the Distribution object. For more information, see Section 3.12, "Working Directories," on page 189.

3 Click *General > Messaging* and fill in the following fields:

Use policy: Select to use the effective policy if you want to use the values set in the Tiered Electronic Distribution policy. This field is displayed if a Tiered Electronic Distribution policy has been created, distributed to the Distributor server, extracted by the Policy/Package Agent, and enforced on the server. If you select this option, the rest of the fields are dimmed and the policy settings for messaging are used instead.

Server console: Specifies the level of output messages to send to the Distributor console on the server console.

For more information on the message notification levels, see Section 3.11.5, "Minimizing Messaging Traffic," on page 186.

SNMP trap: Specifies the level of messages to send via SNMP.

Log file: Specifies the level of messages to send to the log file.

Path and filename: You can specify a custom log file's name and location for this Distributor object. The default is:

- ◆ **NetWare:** `sys:\zenworks\pds\ted\dist\ted.log`

  IMPORTANT: The default volume is sys: on NetWare servers. We recommend that you do not use the sys: volume because the log file can become quite large.

- ◆ **Windows:** `c:\zenworks\pds\ted\dist\ted.log`
- ◆ **Linux and Solaris:** `/var/opt/zenworks/zfs/pds/ted/dist/ted.log`

For information on creating custom log files for all Distributor objects by using the Tiered Electronic Distribution policy, see Section 4.3.5, "Creating Custom Log Files Using Policies," on page 237.

Delete log entries older than __ days: Log file entries for a Distributor are deleted after they are older than the number of days specified. The default is six days.

E-mail: Specifies which level of messages are sent via e-mail.

Users: Specifies e-mail users for notification.

Address attribute: Specifies e-mail addresses for notification.

You can add users or groups stored in eDirectory or provide the e-mail addresses for users who are not contained in eDirectory. The e-mail Address Attribute associated with an eDirectory user is the default attribute.

IMPORTANT: If you select e-mail as a method for receiving notification, be aware that additional network traffic can be created.

4 Select the *Schedules* tab.

The schedule for a Distributor determines how often it reads the information contained in the Tiered Electronic Distribution objects in eDirectory. It reads the Channel, Distribution, and Distributor objects based on this schedule. You can set this up to reflect how often you expect information in these objects to change, or how often new objects might be created. However, we recommend that you leave its schedule set to the default of Never to prevent the possibility of the refresh interrupting the building or sending process, which could cause an infinite loop where the Distribution never finishes being built or sent.

We recommned that you after you create a Distribution, you force the Distributor to read eDirectory by right-clicking the Distributor object and selecting the *Refresh* menu option.

5 Select a schedule and fill in the fields:

Use policy: Select to use the effective policy if you want to use the values set in the Tiered Electronic Distribution policy. This field is displayed if a Tiered Electronic Distribution policy has been created, distributed to the Distributor server, extracted by the Policy/Package Agent, and enforced on the server. If you select this option, the rest of the fields are dimmed and the policy settings are used instead.

Schedule type: The Refresh schedule you selects determines when the Distributor reads eDirectory again.

> **IMPORTANT:** We recommend the Distributor's Refresh schedule be daily, unless changes to Distributions warrant a more frequent refresh. However, do not refresh the Distributor more often than every five minutes. The following can need up to five minutes to complete their processes: Distribution building, eDirectory replication, and tree walking (when no Search policy is defined).

For information on available schedules, see Chapter 8, "Scheduling," on page 325.

6 Select the *Routing* tab and create the Distributor's routing hierarchy.

Subscriber routing hierarchy: Configure the routes the Distributor uses when sending Distributions to the Subscribers. You should plan this hierarchy in advance.

Use the following method to create the hierarchy:

6a Select the Distributor.

6b Click *Add*, select one or more Subscribers, click *Select*, then click *OK*.

You can have multiple Subscribers directly under the Distributor.

6c Select one Subscriber.

6d Click *Add*, select one or more Subscribers, click *Select*, then click *OK*.

You can have multiple Subscribers directly under each Subscriber.

6e Repeat Step 6c and Step 6d for each Subscriber until you have created the desired hierarchy.

7 Select the *Distributions* tab to view the Distributions being serviced by this Distributor.

8 To edit a Distribution, select the Distribution, click *Details*, edit the properties, then click *OK* to exit the Distribution object's properties.

9 When you have finished configuring the Distributor and its Distributions, click *OK* to exit the Distributor object's properties.

> **IMPORTANT:** Changes made to Tiered Electronic Distribution objects (other than Distribution) are not in effect until the Distributor reads eDirectory.

3.3.5 Manually Refreshing the Distributor

Any time you make a change in eDirectory that affects the Distributor, you must manually refresh the Distributor so that it knows of that change.

For example, when you create a new Distribution, the Build schedule does not make the Distributor aware of the new Distribution. You must manually refresh the Distributor so that it can detect the change in eDirectory.

To refresh the Distributor:

1 In ConsoleOne, right-click the Distributor object.

2 Click *Refresh Distributor*.

This causes the Distributor to read eDirectory and obtain all of the changes that were made in eDirectory. The Distributor Agent can then act on any changes applicable to the Distributor.

To perform this task in iManager, see .

Distribution building begins according to the current Build schedule. The Distribution is sent according to the Send schedule.

As soon as Subscribers receive an entire Distribution, they extract the contents to their working directories that are specified in the Subscriber objects' properties.

3.3.6 Deleting a Distributor Object and How Its Distributions Are Affected

You can delete Distributor objects from eDirectory. However, you can lose the following important information that you might want to reuse for the Distributor's replacement:

 * The Distributor's distribution hierarchy

   This is part of the Distribution object's properties, and it shows which Subscriber servers are used for passing on the Distributions.
 * The list of its Distributions

   The Distributor's Distributions become orphaned and unusable.

For information on how to handle orphaned Distributions, see .

3.4 Distributions

The following sections provide concepts and instructions for the Distribution object:

3.4.1 Understanding Distributions

The Distribution (TED Distribution) object contains a list of data packages or data grouping information.

Functional Relationship with Other Tiered Electronic Distribution Objects

Figure 3-13 illustrates a Distribution's relationship with its Distributor and the Channels:

Figure 3-13   Distributor with a Distribution Listed in Multiple Channels
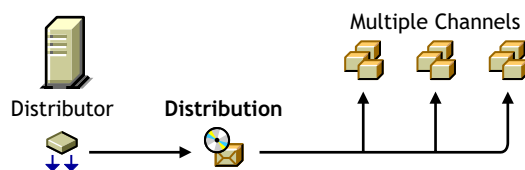
The Distributor associates its Distributions with the Channels.

Figure 3-14 illustrates that a Distributor sends Distributions to Subscriber servers:

Figure 3-14   A Distributor Sends Its Distribution to Two Subscribers.
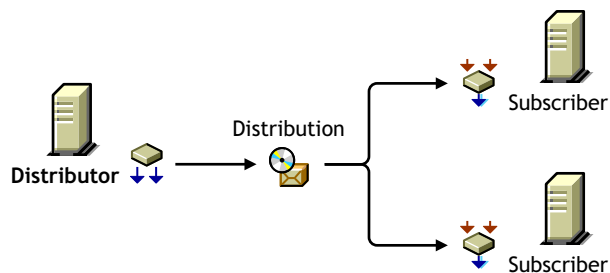
Distribution Description

A Distribution is a compilation of software and/or files, or a policy package, that the various servers in your network might need.

A Distribution is owned by only one Distributor. A Distribution keeps a list of its Channel associations, and you can place the Distribution in multiple Channels.

When a Distribution is built, it is built according to its type. There are seven types of Distributions:

Desktop Application [1]
File
FTP
HTTP
MSI
Policy Package
RPM
Software Package

[1] The Desktop Application Distribution is only available when Desktop Management is installed.

For information on the different Distribution types, see "The Distribution Types" on page 118.

Scheduling

A Distribution has a Build schedule that notifies its Distributor how often the Distribution needs to be built. If a Distribution has changed since the last time it was built, a new one is created.

Distributions can also be made active or inactive to control whether they should be built.

For information on scheduling, see Chapter 8, "Scheduling," on page 325.

How New Versions of Existing Distributions are Created and Distributed

After you have configured a Distribution object and set the various distribution schedules, newer versions of existing Distributions are automatically created and distributed according to the following parameters:

- **Refresh schedule:** This schedule determines when a Distributor reads eDirectory for changes to any of its Distributions. If changes are detected for a particular Distribution, it is rebuilt according to that Distribution's Build schedule.

  For more information on the Refresh schedule, see "Distributor Object's Refresh Schedule" on page 329.

- **Build schedule:** This schedule is set independently for each Distribution. When the schedule starts for a Distribution that has been determined to have had changes to it, the Distributor proceeds to rebuild that Distribution.

  For more information on the Build schedule, see "Distribution Object's Build Schedule" on page 330.

- **Maximum revisions:** This field (in the Distribution object's properties, click General > Settings), determines how many versions of a Distribution are kept on the Distributor and Subscriber servers' file systems. For some Distribution types, this field determines whether a partial Distribution (delta) or complete Distribution is rebuilt. Otherwise, this field is used mainly to control disk space usage.

  When the maximum number of revisions is being approached, an SMTP e-mail notification is sent, if SMTP notifications have been configured.

  For more information on the Maximum Revisions field schedule, see "Maximum Revisions" on page 116.

These parameters determine when a Distribution needs to be rebuilt. The other schedules (Send and Extract) determine when the rebuilt Distribution file is sent and extracted.

Distribution Security

Policy and Distribution Services provides several means for securing Distributions:

- "Certificates" on page 114
- "Encryption" on page 114
- "Inter-Server Communications" on page 114

Certificates

A certificate is a security mechanism used by Policy and Distribution Services to ensure that the Distribution received by a Subscriber was actually sent by the Distributor owning that Distribution. Without a matching certificate, a Subscriber cannot receive Distributions from the Distributor.

For more information, see Section 7.1, "Distribution Security Using Signed Certificates and Digests," on page 307.

Encryption

You can encrypt Distributions for when you send them outside your secure network.

For more information, see Section 7.2, "Distribution Security Using Encryption," on page 317.

Inter-Server Communications

You can secure communications between Tiered Electronic Distribution components residing inside and outside your secure network by installing inter-server communications security where needed.

For more information, see Section 7.3, "Security for Inter-Server Communication Across Non-Secured Connections," on page 321.

Distribution Deletions

When a Distribution is built, any deletions in the Distribution object or on the Distributor server's file system, such as deleting files or directories, causes those files or directories to also be deleted from the Distribution when it is rebuilt. However, synchronization must be enabled in order for the files and folders to also be removed from the Subscriber server's file system.

For more information, see Section 3.11.1, "Directory Sync Granularity for File Distributions," on page 178.

Clean Up of Temporary Distribution Files

To reduce the amount of disk space taken up by temporary Distribution files, those files are now automatically cleaned up. Previously, after a Subscriber server extracts a Distribution, the `distfile.ted` file is left in the Subscriber server's working directory.

With the clean-up feature, you can have the `distfile.ted` file and other temporary Distribution files automatically deleted after the Distribution has been successfully extracted. However, so that the Subscriber is not sent the Distribution again, the status file is left in the Subscriber's working directory to indicate that the Distribution has been extracted and cleaned up.

IMPORTANT: The Distribution clean-up feature works only for Distributions that have been both sent and received by ZENworks 7 Server Management Distributors and Subscribers. Distributions sent or received by Distributors or Subscribers running prior versions of Server Management software cannot be cleaned up.

Parent Subscribers

For parent Subscribers who might need to forward the Distribution, the files are not cleaned up on the parent Subscriber's server, so that it can still forward the Distribution.

A parent Subscriber that has had a Distribution cleaned up which it is not forwarding, and then receives the same Distribution for forwarding, will receive the Distribution again, even though its status file indicates that it does not need it.

Distribution Types and Clean-Up

The FTP, HTTP, RPM, MSI, Software Package, and Policy Package types of Distributions are always cleaned up. The Maximum Revisions field is not available for those types of Distributions.

The File and Desktop Application types of Distributions can have their temporary files cleaned up after the Distribution has been extracted when the Maximum Revisions field is set to 1. However, for File Distributions, if the Verify Distributions check box is selected, the Distribution is not cleaned up, even if the Maximum Revisions field is set to 1.

Resetting Clean-Up Statuses in iManager

In iManager, you can reset the status file, which forces a Subscriber to receive a Distribution that has been cleaned it up.

Clean-Up Is Not Rollback

Cleaning up the temporary files does not cause any roll back of extracted Distributions. Clean-up is simply removing the temporary files.

Clean Up of Reassigned Distributions

The working directories for a Distribution that is reassigned from an old Distributor to a new Distributor are not automatically cleaned up on the old Distributor's server. You need to manually clean up that Distribution's temporary files on the old Distributor server.

3.4.2 Distribution Issues

Consider the following in determining your Distributions:

- File sizes and their potential for compression (`.jpg` files won't benefit as much from compression as text files)
- The bandwidth of WAN links
- The frequency of file changes
- Network resource constraints, such as low disk space or extra bandwidth availability

The better you can determine this type of information, the better you can balance resource usage and minimize the use of resources.

You can configure Distributions to copy only files that are different than the target, or copy all files in their original state.

The following sections provide information about other issues with Distributions:

- "Maximum Number of Concurrent Distributions" on page 116
- "Maximum Revisions" on page 116
- "I/O Rate (Bytes per Second)" on page 117
- "Updating the Distributor's eDirectory Information" on page 117
- "Checking the Distribution Package Changes" on page 117
- "MSI Distribution Extraction Errors" on page 117

Maximum Number of Concurrent Distributions

This is an attribute found in the Distributor and Subscriber objects. It is used to control the number of Subscribers that can be serviced concurrently when sending Distributions. This is helpful if the Distributor or parent Subscriber is servicing a large number of Subscribers. It prevents the Distributor from spreading itself very thin and sending the Distribution to all of the Subscribers at once.

For example, if a Distributor or parent Subscriber sends to 100 Subscribers and the number of concurrent Distributions is set to 10, then the sender starts with 10 connections. As one connected Subscriber finishes receiving the Distribution, another Subscriber is added in its place in the list of 10. This continues until all 100 have been serviced.

Maximum Revisions

Each Distribution allows you to determine how many versions of the Distribution are kept by the Distributor and Subscribers in their working directories. The default is infinite for all Distribution types, whether the Distribution is created in ConsoleOne or iManager; however, the File and Desktop Application types of Distributions have a default of 10 if they are created in ConsoleOne. Make sure that you fill in the Maximum Revisions field attribute when creating Distributions. Consider disk space availability when calculating the maximum number of revisions.

If you select to limit the revisions, the e-mail fields are available, where you can specify a trigger to notify e-mail recipients when your maximum number is approached, as well as define who the e-mail recipients are. If you select to allow unlimited revisions, the e-mail fields are not available.

The File Distribution only builds a complete Distribution the first time it creates the Distribution. All subsequent versions are just the differences (deltas) between a current version and its previous version. However, when the File Distribution reaches its maximum number of revisions, it deletes all previous versions and build an entirely new Distribution (called a baseline), and starts from 1 in counting the number of revisions.

When the maximum number of revisions is met for FTP, HTTP, and Server Software Package Distribution types, the agent deletes the oldest version of the Distribution and adds the current version to the revisions. Therefore, it never exceeds the maximum number entered in the Distribution object.

When the maximum number of revisions is being approached, an SMTP e-mail notification is sent if SMTP notification has been configured.

I/O Rate (Bytes per Second)

This is an attribute found in the Distributor and Subscriber objects. It is used to control the amount of bandwidth used by the Distributor or parent Subscriber when sending Distributions. The default is unlimited, meaning the sender uses all the bandwidth available in sending Distributions.

Updating the Distributor's eDirectory Information

The Distributor must be updated with the configuration information contained in the Tiered Electronic Distribution objects in eDirectory.

Configuration changes include any changes made to the attributes of the Distributor object, Distribution objects belonging to that Distributor object, or Channel objects to which the Distributor object is associated.

The Distributor has a schedule that determines how often it reads eDirectory for configuration information. Set this schedule to coincide with the frequency at which Tiered Electronic Distribution objects are modified in eDirectory.

You can also force an eDirectory refresh by right-clicking a Distributor object and selecting the Refresh menu option, or by using the ZENworks Server Management role in iManager (see).

Checking the Distribution Package Changes

The Distribution's Build schedule tells the Distributor the frequency at which the Distribution should be checked for changes.

For example, the Distribution schedule might specify a weekly build. The Distributor rebuilds that package and compares it to the previous version to see if there have been any changes.

MSI Distribution Extraction Errors

Some MSI Distributions can fail to extract on Windows 2000 servers (usually displaying error 1603), but not on Windows Server 2003 servers. The difference is in how the two operating systems differently handle the rights needed to install the MSI packages.

This can be solved for Windows 2000 servers by editing the properties in the MSI Distribution:

1 In ConsoleOne, access the MSI Distribution's properties.

2 On the *Type* tab, select the MSI package listed under *Selected Packages*.

3 Click *Edit Parameter List* to open the Edit Parameters dialog box.

4 In the *Custom Parameters* field, enter:

```
ALLUSERS=1
```

5 Click *OK* to save the change.

6 Repeat Step 2 through Step 5 for each MSI package listed under *Selected Packages*.

7 Click *OK* to save the updated MSI Distribution properties.

3.4.3 Determining the Distributions

You can distribute whatever you can represent on the file system. This includes server applications and files. For example, the applications or files could fulfill one of the following purposes:

- Installing server software (such as virus protection software)
- Updating server software (such as a NetWare support pack)
- Updating files (such as virus patterns) on servers
- Enforcing standardization of server files or configurations (such as replacing the `autoexec.ncf` file on a NetWare server with an updated version)

Use a descriptive method for naming the Distributions. You can also use these names for naming the Channels associated with the Distributions. For example:

```
VirusProtect
VProtectPatterns
NW51patch4
NW6patch1
AUTOEXECNCF000326
```

The following sections explain the different Distribution types and issues related to determining your Distributions:

- "The Distribution Types" on page 118
- "Determining the Sizes and Frequencies for Distribution Packages" on page 123

The Distribution Types

There are several Distribution types. Each type has unique features that tailor it for specific needs.

- "Desktop Application" on page 119
- "File" on page 119
- "FTP" on page 121
- "HTTP" on page 121
- "MSI" on page 121
- "Policy Package" on page 122
- "RPM" on page 123
- "Software Package" on page 123

For information on how to configure each Distribution type, see Section 3.4.4, "Creating a Distribution," on page 124 (specifically, Step 6 on page 126).

For the File and FTP types of Distributions, a Distribution Wizard is available for automating the process of creating them. For more information, see Section 3.4.12, "Using the Distribution Wizard," on page 144.

Desktop Application

Distributes the Application objects (that are created in Desktop Management) and the application's associated files to specified locations on the eDirectory tree and target Subscriber servers. This Distribution type allows you to solve geographic, workload, and redundancy issues for applications distributed by Novell Application Launcher that otherwise might require much of your time in manual configuration work in Desktop Management. For more information, see Chapter 6, "Desktop Application Distribution," on page 279.

The Desktop Application Distribution type is not supported for Linux and Solaris servers.

This Distribution type automatically distributes a modified copy of the original Application object to a context in the eDirectory tree (a Subscriber's working context), and automatically copies the application's files to the Subscriber server that can locally service its users and workstations. It performs all of the appropriate hookups to the modified Application object to render it fully functional.

For the Desktop Application Distribution, you can set the maximum number of revisions in the Distribution object. When the version number reaches the number that you set, the Distributor rebuilds the entire Distribution. By default, this number is 10.

You can send Desktop Application Distributions to Subscriber servers on a tree that is different from the Distributor server's. However, the recipient server's Subscriber object must reside in the same tree as the modified Application objects that are created by the Distribution. The External Subscriber object is used on the Distributor's tree to send a Desktop Application Distribution to a server on another tree.

File

With this type you can select files and/or directories from the Distributor server's file system for distribution, and select a destination location for extraction on the Subscriber.

The File type is sequential, meaning it controls the order for the building and extraction of Distributions. This prevents the building and extracting processes from being performed out of sync.

IMPORTANT: Linux and Solaris file systems are case sensitive to allow paths and filenames that are identical except for case differences. However, if you select two such files, only the first file selected during extraction is distributed, because the File type is not case sensitive. Therefore, do not place two files into a File Distribution where their paths and filenames are identical except for case differences.

Also, if a NetWare server is the target for a File Distribution, you might encounter an error due to code page differences where extended characters are used (such as ê, ë, ì, or í). The information in "Extended Characters in Directory Paths" on page 294 in the Desktop Application Distribution section is also applicable to File Distributions.

By default, Cache and Forward is used. This process allows a parent Subscriber to begin sending a Distribution to subordinate Subscribers before it has finished receiving the Distribution. This allows entire Distributions to be sent more quickly through a chain of parent Subscribers in the Distributor's routing hierarchy than if they each had to wait until each Subscriber had completed receiving the Distribution before it started sending.

The File Distribution is useful for distributing large Distributions that change often, thus requiring updates that need to be distributed frequently.

For the first version of a Distribution, the Distributor builds the entire Distribution (creating a baseline). A unique feature of the File type is that for all subsequent versions it calculates the differences at build time and only builds a delta of the Distribution.

The File type does this by keeping a list of the files and directories contained in a Distribution on the source machine (the Distributor or a parent Subscriber). If a source file changes, a new Distribution is built the next time its Build schedule starts. However, this new Distribution only contains the files that are different between the previous version and the current version. This is known as a delta of the original Distribution.

This delta of the Distribution file is what is distributed to the Subscribers-not the entire Distribution.

The File type is also effective when changes are frequent because it can build much smaller deltas.

There is no option to send the entire File Distribution. However, after the maximum number of revisions has been met, the Distribution is completely rebuilt and all deltas and previous revisions are deleted. Therefore, if you set the maximum number of revisions to 1, deltas are not used and the entire Distribution is built and sent every time.

For example, the first build is the baseline Distribution (version 1), the first update (Delta 1) is version number 2, the second update (Delta 2) is version number 3, and so on until the number of revisions you set is reached, which triggers a new baseline rebuild. By default, this number is 10.

Pre and Post actions can be set for File Distributions. For more information, see Section 3.4.6, "Pre and Post Processing for Distributions," on page 127.

You can set the maximum number of revisions in the Distribution object.

If synchronization is enabled, you can use the File type for removing files and directories from the Subscriber server's file system upon extraction of the Distribution in one of two ways:

 • **Edit the Distribution object:** Remove files from the list of files and directories in the Distribution object. When the Distribution is built again, those files and directories are not included.

 • **Remove files from the Distributor's file system:** Remove files from the Distributor's file system that were part of the Distribution. When the Distributor is refreshed, it rebuilds the Distribution without those files and directories.

In both cases, upon extraction of the Distribution, and with synchronization enabled, those files and directories are removed from the Subscriber server's file system. For more information on synchronization, see Section 3.11.1, "Directory Sync Granularity for File Distributions," on page 178.

To manually force a Distribution to be built, you can use iManager (see "Forcing Policy and Distribution Services Agent Actions" on page 80).

FTP

With this type you can create a Distribution consisting of files from one or more FTP sources. Each source can contain one or more directories and/or files.

When an FTP site directory entry is a directory, all of its files and subdirectories are built for the Distribution.

Server Management now supports retrieval of symbolic link files. This allows the Linux or Solaris environments to receive FTP files that would be considered invalid on other platforms.

Whenever a Distribution's Build schedule starts:

- The FTP type creates a new Distribution only if the new version would be different than the previous version.
- The Distributor builds the entire new Distribution.
- The Distributor sends each new version of the Distribution to the appropriate Subscribers.

You can set a maximum number of revisions in the Distribution object to conserve disk space. By default, the number is unlimited.

HTTP

With this type you can create a Distribution consisting of one or more HTTP sources. Each source can contain one or more target entries.

Whenever a Distribution's Build schedule starts:

- The HTTP type creates a new Distribution only if the new version would be different than the previous version.
- The Distributor builds the entire new Distribution.
- The Distributor sends each new version of the Distribution to the appropriate Subscribers.

You can set a maximum number of revisions in the Distribution object to conserve disk space. By default, the number is unlimited.

MSI

Distributes Microsoft Software Installer (MSI) packages to Windows servers, where the MSI engine is used to install the Windows-specific software included in an MSI Distribution. Any vendor can create MSI packages for their software for installing in a Windows environment.

The Installshield* AdminStudio* ZENworks Edition software for creating `.mst` files is available on its own CD that is provided with the ZENworks 7 product.

MSI 3 is supported as a version that can be distributed. However, ZENworks 7 does not individually support any of MSI 3's new features.

The components of an MSI Distribution consist of `.msi`, `.msp`, and `.mst` files:

- **.msi file:** An MSI package containing Microsoft software to be installed by the MSI engine.

  An MSI package can include just the `.msi` file, or the `.msi` file with the other files in its folder and all of the files contained in any subfolders.

- **.mst file:** A file that adds, deletes, or changes the properties in an MSI package to enable customizing of the installation for different groups of users.
- **.msp file:** An MSP package that provides a patch to an MSI package.

  MSI-based patch files might have filename extensions other than `.msp`.

An MSI Distribution might contain:

- One or more MSI packages
- One or more MSI packages with one or more `.mst` files applied to each MSI package
- One or more MSI packages with one or more MSP packages
- One or more MSI packages with one or more MSP packages and one or more `.mst` files applied to each MSI package
- One or more MSP packages only, because `.msp` files contain the information necessary for identifying the MSI packages' applications that they are to patch

  Patching can include modifying the settings of a machine, as well as updating files.

You can determine the application order of the `.mst` files for each MSI package, and you can determine the execution order of the MSI and MSP packages listed in the Distribution.

When an MSI Distribution includes both of the MSI and MSP components, post-installation actions are added by the Distributor to the Distribution to ensure the correct order of completion.

Because an MSP is designed to modify a specific MSI package, you need to make sure that you have the correct order of execution.

Some MSI Distributions can fail to extract on Windows 2000 servers. To solve this problem, see .

IMPORTANT: Because an MSI Distribution recursively gathers all of the files from the MSI file's location, if you have multiple `.msi` files in a given location, all other files and subdirectories contained therein are gathered once for each `.msi` file. The distribution gathering process cannot determine which other files or subdirectories belong to each `.msi` file, so you can end up with a much larger MSI Distribution file than is necessary. Therefore, instead of storing your `.msi` files in one directory, place each into its own subdirectory with its own supporting files and subdirectories.

Policy Package

This type provides the mechanism for applying policies to servers. In previous versions of Policy and Distribution Services, all policies were enforced through eDirectory object and container associations. With ZENworks 7 Server Management, policies are now distributed Subscriber servers for enforcement using the Distributed Policy Package. However, policies for Distributors continue to be enforced through context associations using the Container Package or Service Location Package.

With the Policy Package Distribution, you send policies directly to servers as Distributions, which are extracted on the receiving Subscriber server. The contained policies are then enforced on that server.

You can set a maximum number of revisions in the Distribution object to conserve disk space. By default, the number is unlimited.

To send a Policy Package Distribution to a Subscriber using an External Subscriber object, you must edit the `agentinfo.properties` file to prevent trusted tree errors. For more information, see "Preventing Trusted Tree Errors for Policy Package Distributions" on page 163.

For more information on each policy, see Section 4.1.6, "Server Policy Descriptions," on page 204.

RPM

You can distribute any Red Hat Package Manager (RPM) packages that you have previously created to your Linux and Solaris servers using the RPM Distribution.

For Solaris, RPM must first be installed on the server, because it is not installed with Solaris software by default. Solaris' equivalent is PKG.

Whenever a Distribution's Build schedule starts:

- The Distributor builds the entire new Distribution.
- The Distributor sends each new version of the Distribution to the appropriate Subscribers.

You can set a maximum number of revisions in the Distribution object to conserve disk space. By default, the number is unlimited.

Software Package

A Server Software Package is created in ConsoleOne in the Server Software Package namespace. For more information, see Chapter 5, "Server Software Packages," on page 243.

Software Package is the most robust type of Distribution. It includes installation prerequisites, pre-installation instructions, post-installation instructions, and the ability to modify text fields, SET parameters, registry settings, and the `products.dat` file.

With the Software Package Distribution you can select `.cpk` files for distribution. This allows you to place a software product into a Distribution for automatic installation on the receiving server. This can include software updates to existing server software on the server.

You can select multiple `.cpk` files for one Distribution. Then, individual `.cpk` files are applied on the Subscriber, depending on whether the `.cpk` file's prerequisites are met.

IMPORTANT: The order that the `.cpk` files are applied on a server is not guaranteed, and `.cpk` files contained in one Distribution that might start in a certain order might not all finish in that same order. Therefore, place each `.cpk` file in its own Distribution if you want them to be installed in a particular order and use Distribution scheduling to determine the order. For more information, see "Forcing the Software Package Distribution Order Using Multiple Distributions" on page 245.

Determining the Sizes and Frequencies for Distribution Packages

A Distribution's size and frequency of being built and sent depends on the following:

- The size and number of files being distributed. Knowing this helps in determining the amount of disk space to be used on Distributor, Subscriber, and parent Subscriber servers.
- A Software Package Distribution (`.cpk`) always builds an entirely new version of the Distribution each time the source changes.
- HTTP and FTP Distributions always build an entirely new version of the Distribution whether the source has changed or not.

- How often the packages change and need updating. Knowing this helps determine how frequently new versions of the package are created. Servers required to rebuild large Distribution packages on a regular basis should have the processing power to perform this work. The creation of many versions of a package also affects the amount of disk space used in the Distributor's working directory.

- The number of versions of a Distribution package that are retained. This also affects disk space usage on the Distributor's and Subscribers' servers.

- The File Distribution creates a delta file for each new version of the Distribution until it reaches the number you have specified in the Maximum Number of Revisions field (10 is the default). Then it begins a new baseline Distribution. The delta file contains only the differences between the last and current versions of the Distribution.

3.4.4 Creating a Distribution

1 In ConsoleOne, select the container where you want the Distribution to be created, click *File > New > Object*, select the *Distribution* type, then click *OK*.

2 Specify a Distribution name.

> **IMPORTANT:** Periods (.) are not allowed in Distribution names. Instead, use dashes (-) or underscores (_) as word separators. If you use a period in the Distribution name, the Distribution is not sent, and the Distributor is not reloaded after it has been exited.

3 To give the Distributor ownership of the Distribution, browse to select the Distributor object, select *Define Additional Properties*, then click *OK*.

The Distribution object's properties are displayed.

Each Distribution belongs to a single Distributor that builds and sends the Distribution.

4 Click *General > Settings* and fill in the fields:

Active: Required. In order to make a Distribution available to Subscribers, it needs to be active.

Use digests: Digests are used by Distributors and Subscribers to verify that Distributions have not been tampered with while in transit. The digest provides an MD5 checksum for the Subscriber to compare.

Creating a digest takes more time on larger Distributions. The number of minutes per megabyte is dependent on the hardware configuration of the server where the digest is being created.

Digests also detect corruption in a Distribution's package. In the case of corruption, the Subscriber renames the `distfile.ted` Distribution file to `distfile.corrupt` and the Distribution is rebuilt and sent the next time the Channel's schedule fires.

Encrypt: You can have the Distribution encrypted if you are sending it across non-secured connections. Encryption provides security for the Distribution during transit between the Distributor and Subscriber when they are not within the same firewall. Select either Strong or Weak encryption. You also must have the same version of NICI 2.6.4 installed to each of these servers for encryption to work (see "Installing NICI 2.6.4" on page 54). However, if you already have NICI 2.4.6 installed, it is optional whether you upgrade to NICI 2.6.4, because these versions are compatible with each other.

Maximum revisions: This number helps you to control disk space usage by determining how many versions of a particular Distribution are kept in the Distributors' and Subscribers' working directories. The default is 10. Select *Limited* and enter a number.

Increase the number if data is changing often and the changes are minimal (smaller delta files). Decrease the number if data is not changing very often, or if a significant amount of data is changing (larger delta files).

The following e-mail options are available if you set a maximum number. If you select *Unlimited*, these options are dimmed.

- ◆ **Approaching maximum revision email notification list:** Contains the e-mail addresses of anyone who is to be notified when a Distribution is approaching the maximum revisions set in the *Maximum revisions* field. Here, you can either remove a single or all displayed addresses.

- ◆ **Email address (maximum revision notification):** You can add e-mail addresses to the list in *Approaching maximum revision email notification list*. Just enter an e-mail address and click *Add* and it is displayed in the listing.

- ◆ **Send notifications when Distribution revision is ___ or less of reaching maximum revisions:** Enter a number to indicate how close "approaching" is. When the current revision number of Distribution plus this number equal the maximum revisions, an SMTP notification is sent to the listed addressees.

  SMTP must be configured and its e-mail server address listed in the next field.

- ◆ **Email server address:** The SMTP server used to send the e-mail notifications. For example, mail.novell.com.

  For information on configuring SMTP e-mail notifications, see "SMTP Host" on page 212.

Priority: You can give the Distribution a priority that determines how it is sent in relation to other Distributions. A High priority means it is sent before Medium or Low priority Distributions. For information on prioritizing Distributions, see Section 3.4.5, "Prioritizing Distributions," on page 127.

Distributor: The DN of the Distributor object that builds and sends this Distribution.This attribute cannot be modified. You selected the Distributor when you created the Distribution object.

Description: Provide useful details about the Distribution, such as the name of the desktop application, the files and directories it contains, intended user groups, and so on.

5 Click *General > Restrictions* and select a platform restriction:

Platform restrictions: If you want to select specific operating system versions as a prerequisite to receiving this Distribution, deselect No Restrictions and select the desired operating system version. You can select from the following:

No Restrictions
NetWare All
NetWare 4.*x* (earlier versions of ZfS supported these platforms)
NetWare 5.0 (earlier versions of ZfS supported this platform)
NetWare 5.1
NetWare 5.*x*
NetWare 6.*x*
Windows Server
Solaris
Linux

Selecting the No Restrictions check box means that the Distribution can be sent to any platform.

If you select NetWare All, you do not need to select any of the individual NetWare platforms.

6 Select the *Type* tab and use the drop-down box to choose a Distribution type in the *Select Type* field:

Section A.1, "Desktop Application," on page 393

Section A.2, "File," on page 393

Section A.3, "FTP," on page 397

Section A.4, "HTTP," on page 400

Section A.5, "MSI," on page 402

Section A.6, "Policy Package," on page 405

Section A.7, "RPM," on page 406

Section A.8, "Software Package," on page 407

For some Distribution types, when entering information into a field, such as a directory name, be sure to press Enter or the change is not saved.

IMPORTANT: For the FTP, HTTP, RPM, Software Package, and Desktop Application types of Distributions, if a target file is found to be locked during extraction, the Subscriber throws an exception stating that the file could not be copied. The Distributor receives this information from the Subscriber and logs the failure in the reporting database.

---

7 Select the *Schedule* tab and select a schedule:

The Build schedule determines how often the Distributor builds a new version of the Distribution.

Two options allow you to override the Channel's Send and Subscriber's Extract schedules:

- **Send Distribution immediately after building:** Overrides the Channel's Send schedule, allowing you to immediately send the Distribution, rather than wait for the Send schedule to start. However, the Subscriber's Extract schedule determines when it is extracted for use.

- **Extract Distribution immediately after receiving:** Overrides the Subscriber's Extract schedule, allowing the Distribution to be immediately extracted, rather than wait for the Extract schedule to start. This is useful for Distributions that need to be extracted immediately, such as a Distribution that provides virus patterns.

Build schedule for File Distributions: This type builds a new Distribution and compares it with the previous version for changes. If there are changes, the File type builds a file consisting of the differences between the current version and the previous version. When the maximum number of versions is reached, the type builds a complete Distribution (not just a file containing the differences) and deletes all previous versions.

Build schedule for HTTP, FTP, and Software Package Distributions: These types build new versions of the Distribution each time the Build schedule starts, regardless of whether the Distribution has changed. It sends this new version to all Subscribers.

When sending a Distribution, the sender retries every 2 minutes for 30 minutes, then stops. It does not begin sending again until the Channel schedule starts again.

8 Select the *Channels* tab and fill in the field:

Channels: Each Distribution must be associated with at least one Channel in order for it to be sent to a Subscriber. A Distribution is be sent to all Subscribers of the selected Channel or Channels.

9 If you want to set pre or post actions for the Distribution, see Section 3.4.6, "Pre and Post Processing for Distributions," on page 127 for the steps.

10 Click *OK*, then select *Yes* to resolve the certificates.

For NetWare and Windows servers, this copies the security certificates from the Distributor to Subscriber subscribed to the Channel. For Linux and Solaris servers (if you do not have drives mapped to them), you may need to resolve the certificates manually.

For information, see Section 7.1.6, "Resolving Certificates," on page 311.

3.4.5 Prioritizing Distributions

Distributions can be prioritized in two ways:

   ◆ **Send queue:** You can prioritize the order in which Distributions are sent: High, Medium, or Low. For example, in a given Channel, all High priority Distributions are sent first, then the Medium priority Distributions are sent, and then the Low priority Distributions are sent.

   Because Distributions with mixed priorities cannot be sent concurrently, you can control the order in which Distributions are sent by the priorities that you assign them.

   ◆ **Output rate:** You can configure different output rate settings for a Distribution, based on a priority: High, Medium, or Low. This allows you to control the bandwidth a Distribution uses. For example, if you want your High priority Distributions to utilize the most bandwidth, you should configure their output rates with the High priority. Blank means that bandwidth is taken from third-party applications.

The Maximum Number of Concurrent Distributions value is affected by prioritizing. This value is subordinate to the priorities set for the Distributions. For example:

   ◆ You have the concurrent Distribution number set to 10.

   ◆ There are 3 High priority Distributions.

   ◆ There are 6 Medium priority Distributions.

   ◆ There are 20 Low priority Distributions.

   ◆ Initially, only the 3 High priority Distributions are sent concurrently.

   ◆ After all 3 of the High priority Distributions are sent, the 6 Medium priority Distributions are sent concurrently.

   ◆ After all 6 of the Medium priority Distributions are sent, 10 of the 20 Low priority Distributions are sent concurrently, and so on.

3.4.6 Pre and Post Processing for Distributions

Pre and post processing actions are new features for Distributions in ZENworks 7 Server Management:

   ◆ "Pre and Post Processing Actions Now Available in Distributions" on page 128

   ◆ "Pre and Post Actions in Software Packages versus Distributions" on page 128

   ◆ "The Pre and Post Feature Enhances Software Package Distribution Processing" on page 128

   ◆ "Error Messages Given When Valid Distribution Types Are Not Selected" on page 129

   ◆ "Pre and Post Distribution Processing Actions" on page 129

Pre and Post Processing Actions Now Available in Distributions

To apply execution logic to a Distribution, pre and post actions are now available for the following Distribution types:

File
FTP
HTTP
MSI
RPM
Software Package 1

1 Previously, only a Server Software Package had this functionality. Now both the software package and its Software Package Distribution can have pre and post actions defined.

The benefit of having pre and post actions in these Distribution types is that you are no longer restricted to using only Server Software Packages to perform those actions.

The pre and post processing actions are not available for the following Distribution types:

Desktop Application
Policy Package

Pre and Post Actions in Software Packages versus Distributions

In Server Software Packages, the pre and post features are contained in two different tabs: Pre-Installation and Post-Installation, with Script and Load/Unload tabs for accessing the various options.

For the Distribution types that now have this feature, a Pre/Post Actions tab has been added to their Distribution object's properties (with Pre-Distribution Actions and Post-Distribution Actions tabs).

The following options are available from the Pre/Post Actions tab:

Load Java Class
Script
Start Process
Stop Process
Start Windows Service
Stop Windows Service

For more information on these options, see "Pre and Post Distribution Processing Actions" on page 129.

The Pre and Post Feature Enhances Software Package Distribution Processing

When either a pre or post action is defined for a Software Package Distribution, the following is done:

1. A list of `.cpk` files contained in the Distribution is created in the Type tab of the Distribution object.

2. All pre actions are processed according to the order you defined for them.

3. The `.cpk` files are processed serially.

4. All post actions are processed according to the order you defined for them.

You can use Pre and Post Actions in a Distribution object containing multiple software packages to ensure pre and post actions are performed before and after the software packages listed in the Distribution are processed. However, pre and post processing only guarantees the order on ZENworks 7 Server Management Subscribers, because this functionality is not backwards compatible with ZENworks for Servers 3.*x* Subscribers.

Error Messages Given When Valid Distribution Types Are Not Selected

There are some instances when the Pre/Post Actions tab display only a message:

- In a Distribution object's properties, if you have not yet selected a Distribution type, the following message is displayed on the Pre/Post Actions tab:

  ```
  You must select a Distribution type before you can configure pre or
  post actions.
  ```

  However, you must not only select a Distribution type, you must also "save" it by clicking Apply. Then the Pre or Post page recognizes the Distribution and the Pre or Post actions can be applied.

- If the Distribution type you have selected is either Policy Package or Desktop Application, the following message is displayed on the Pre/Post Actions tab, because pre/post actions are not supported for those types of Distributions:

  ```
  This Distribution type does not support pre or post distribution
  actions.
  ```

Pre and Post Distribution Processing Actions

In each of the following sections, the information provided applies to both the Pre-Distribution Actions and Post-Distribution Actions subtabs of the Pre/Post Actions tab. The difference is that Pre-Distribution Actions occur before the main Distribution is extracted and Post-Distribution Actions occur after the Distribution has completed extracting.

- "Start Process Action" on page 130
- "Stop Process Action" on page 131
- "Start/Stop Windows Service Action" on page 132
- "Script Action" on page 133
- "Load Java Class Action" on page 134

Start Process Action

This action works for Windows services, Java processes, and NLM processes.

The Start Process action is similar to the Load NLM/Process action in Server Software Packages, as illustrated in Figure 3-15:

Figure 3-15 *Properties of Distribution to See MSI Type Dialog Box*
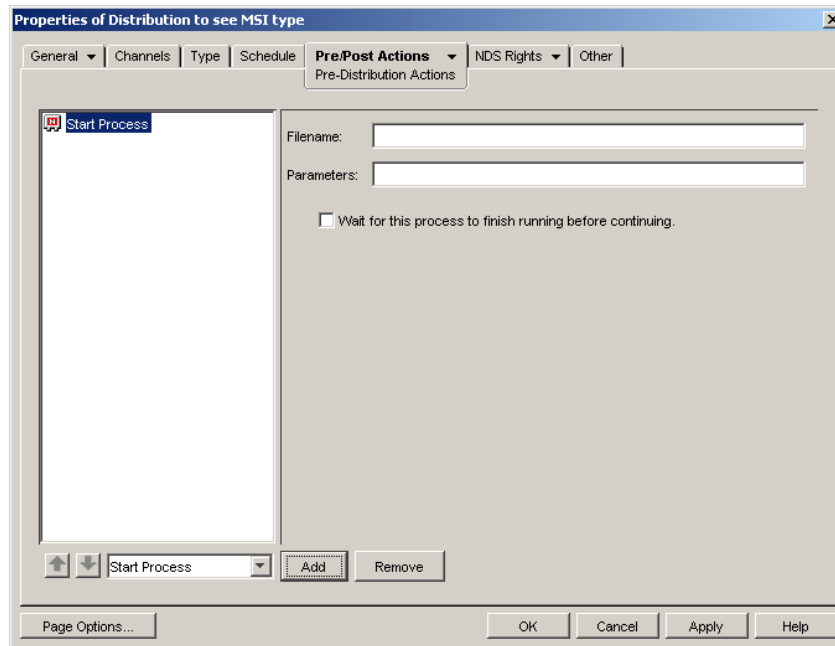
To add a Start Process action, select the option in the drop-down box and click the Add button. Then fill in the fields:

- **Filename:** This must be the exact name. For NetWare, include the `.nlm` extension.

  For Linux and Solaris, you must include the full path.
- **Parameters:** Include any command line parameters for the NLM™ or process being run.
- **Wait for this process to finish running before continuing:** You can select this option for an NLM or process that terminates itself. It must terminate within 10 minutes, or the whole loading process fails. By default, this option is deselected.

If you select an NLM to be loaded by the Distribution, and the NLM is already running on the target server, the package installation fails and is rolled back (if rollback is enabled).

You can make sure that an NLM is not already loaded when you are including it in the Distribution by adding a Stop Process option for that NLM before adding the Start Process option-but only if this NLM does not require user input from the keyboard to unload it.

Stop Process Action

This action works for Windows services, Java processes, and NLM processes.

The Stop Process action is similar to the Unload NLM action in Server Software Packages, as illustrated in Figure 3-16:

Figure 3-16   *Properties of Distribution to See MSI Type Dialog Box*
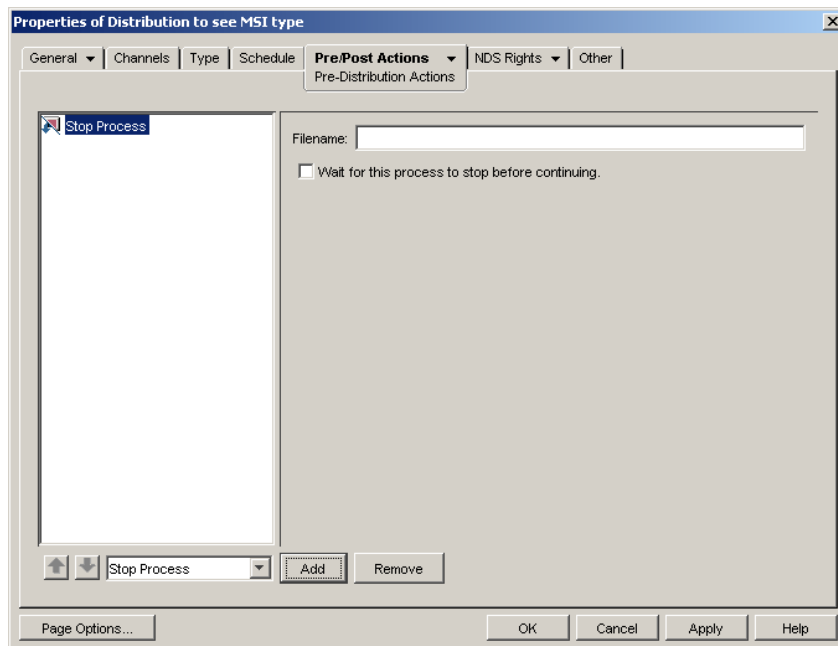
To add a Stop Process action, select the option in the drop-down box and click the Add button. Then fill in the fields:

- **Filename:** This must be the exact name, including the extension if it is an NLM. Because many NLM files require user input to unload, their unloading cannot be automated.

  For Linux and Solaris, only enter the name of the process; you should not enter any path information. All processes running on the machine by that name will be stopped.

- **Wait for this process to finish running before continuing:** You can select this option for a process that unloads itself. By default, this option is deselected.

If an NLM requires intervention to unload, you must remember to unload it manually before trying to install the Distribution.

Start/Stop Windows Service Action

This action works for Windows services only, as illustrated in Figure 3-17 and Figure 3-18::

Figure 3-17 *Properties of Distribution to See MSI Type Dialog Box*
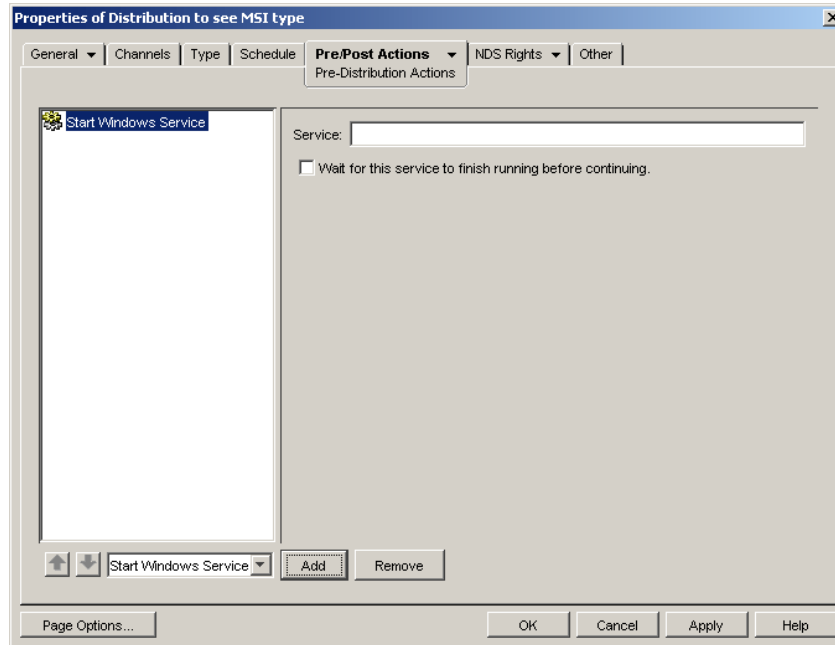
Figure 3-18 *Properties of Distribution to See MSI Type Dialog Box*
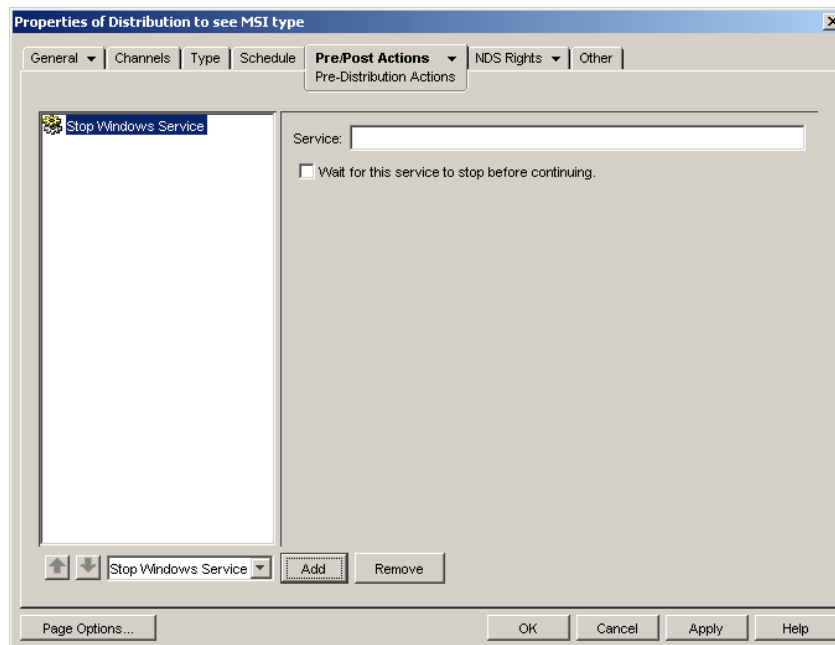
To add a Start/Stop Service action, select the option in the drop-down box and click the Add button. Then fill in the fields:

- **Service:** This must be the exact service name.

◆ **Wait for this service to finish running before continuing:** You can select this option for a service that starts or stops itself. By default, this option is deselected.

Script Action

This action works for Windows services, Java processes, and NLM processes.

You can run server scripts before installing the main Distribution files. Use the arrows to arrange the scripts' execution order.

Figure 3-19 *Properties of Distribution to See MSI Type Dialog Box*
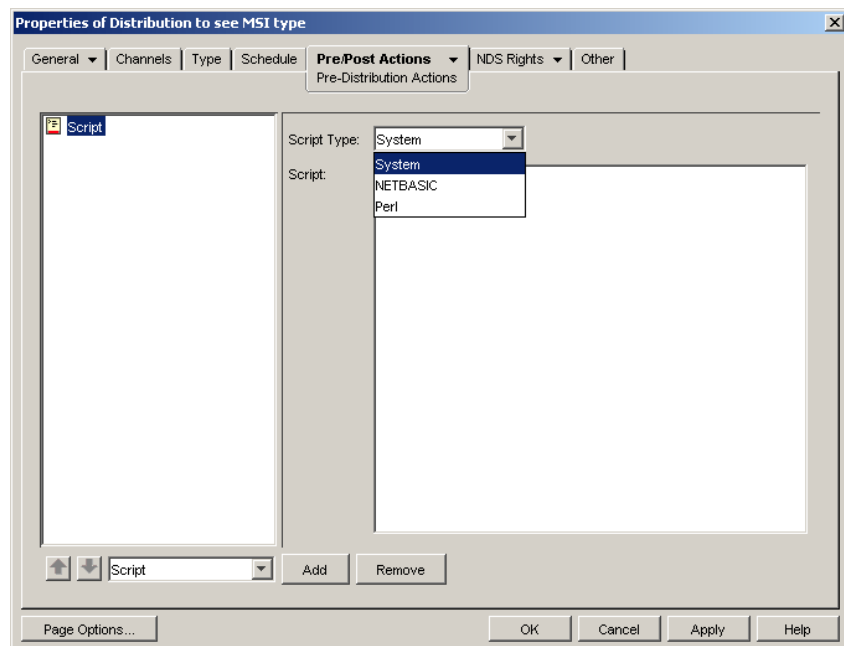
To add a Script action, select the option in the drop-down box and click the Add button. The word "Script" defaults, which you must change to the script filename, including its full path. (Without the path, the script cannot be found to run it.)

Then fill in the fields:

◆ **Script type:** There are three script types: System, NetBasic, and PERL. The text you enter in the Script box must match the type you select.

IMPORTANT: NetBasic is not supported on NetWare 6.5 servers.

◆ **Script:** Enter the text of the script.

WARNING: If a Distribution executes the script, processing done by the script cannot be undone by rollback.

Load Java Class Action

Figure 3-20 *Properties of Distribution to See MSI Type Dialog Box*
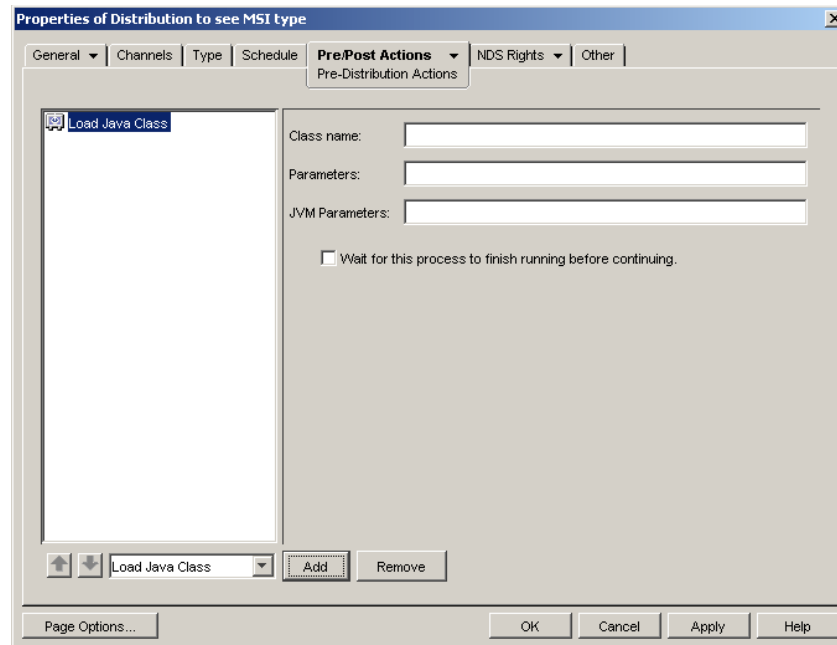
This action works for NetWare only.

To add a Load Java Class action, select the option in the drop-down box and click the Add button. Then fill in the fields:

 * **Class name:** This must be the exact name. The `.class` extension is not necessary.

   > **IMPORTANT:** In order to load a Java class, `java.exe` or `jre.exe` must already be in the path on the server receiving the Distribution. Or, in this field, you can include the full path to the file.

 * **Parameters:** Include any command line parameters for the Java application being run.
 * **JVM parameters:** Include any parameters for the Java machine.
 * **Wait for this process to finish running before continuing:** You can select this option for a Java application that terminates itself. It must terminate within 10 minutes, or the whole loading process fails. By default, this option is deselected.

3.4.7 Reassigning a Distribution to Another Distributor

A single Distributor can service many Distributions, which could cause performance degradation on that Distributor's server. In version 7, there is a way to reassign a Distribution from one Distributor to another to balance the work load without needing to re-create the Distribution.

You can select one or more Distributions and reassign them to another Distributor.

If you delete a Distributor object in ConsoleOne, you are asked if you want to reassign the Distributions that it services.

To reassign a Distribution to another Distributor:

1 Determine which Distributions you want to reassign to another Distributor.

2 In ConsoleOne, do one of the following:

  ◆ Select one or more Distribution objects, click *Tools*, click *ZENworks Server Management*, then click *Assign New Distributor*.
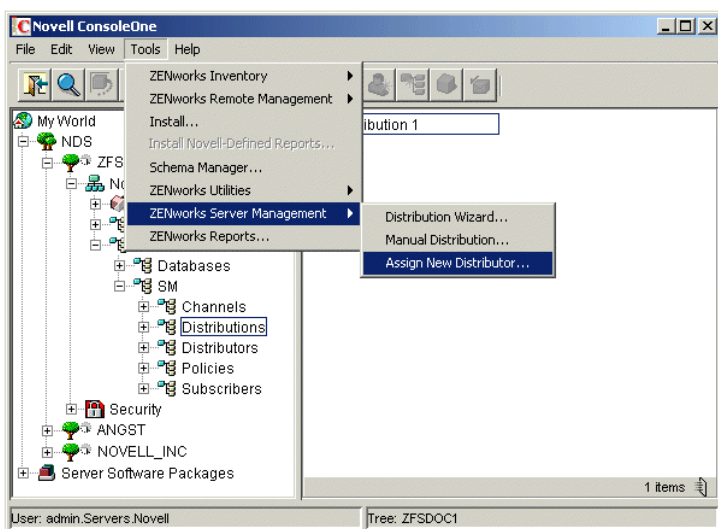
  ◆ Select one or more Distribution objects, right-click the selected objects, then click *Assign New Distributor*.
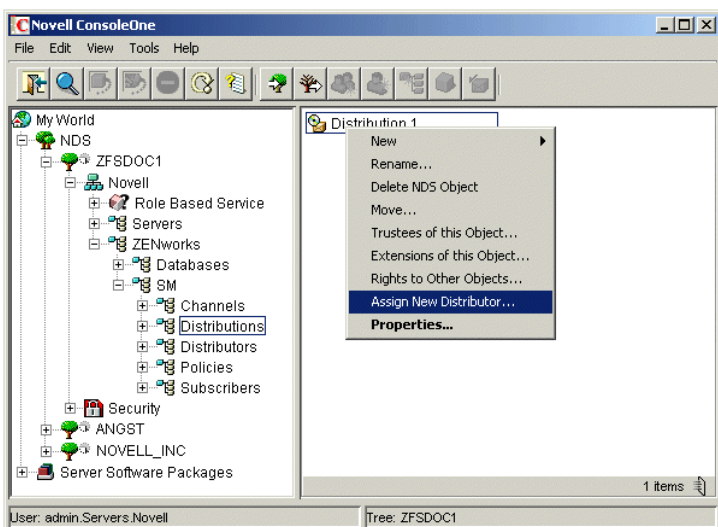
The following dialog box is opened when using either of the above options:
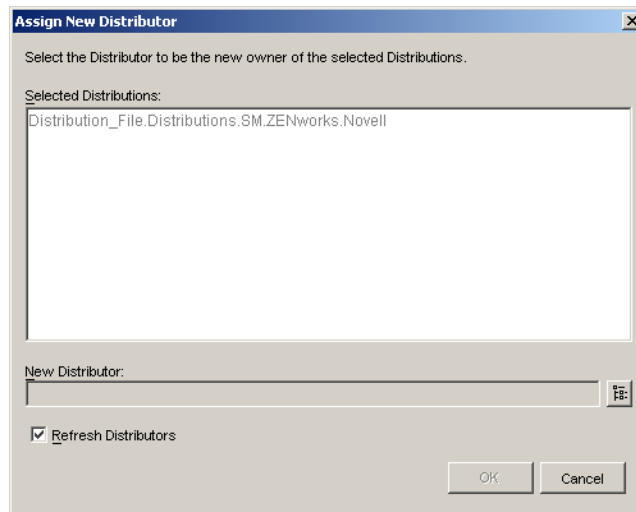
The Distributions you selected are listed in the *Selected Distributions* list.

If you want to change the list, you must click *Cancel* and reselect the Distribution objects.

3 In the *New Distributor* field, browse for the Distributor object that you want to be the new owner of these Distributions.

> **IMPORTANT:** Any files on the current Distributor server's file system that belong in the Distribution must be copied or moved to the new Distributor server's file system, using the identical full path. This is covered in Step 7.

4 If you want the Distributions to be built by the new Distributor owner as soon as you've finished reassigning them, select the *Refresh Distributors* check box.

The new Distributor is refreshed upon exiting this process (see Step 5), so that it immediately recognizes its new Distributions.

> **IMPORTANT:** If you have files to copy, such as for the File, MSI, and Desktop Application types of Distributions, you should wait to refresh the new Distributor until after you have copied or moved the files for those Distributions to the new Distributor server's file system, this task is accomplished in Step 7 through Step 12.

5 Click *OK* to transition the Distribution objects to the new owner.

6 To make the old Distributor aware that it no longer has the Distributions that were reassigned, right-click the old Distributor's object, then click *Refresh Distributor*.

> **IMPORTANT:** The reassignment tool in ConsoleOne only reassigns the eDirectory objects. Therefore, for File or MSI Distributions, the files contained in those Distributions reside on the old Distributor's file system. These files need to be moved to the new Distributor's file system so that the new Distributor has access to them for building these File or MSI types of Distributions. This is covered in Step 7.
>
> For Desktop Application Distributions, you need to review the Application objects to determine which files contained on the old Distributor's file system need to be moved to the new Distributor's file system. This is covered in Step 10.

7 If a Distribution is a File or MSI type, do the following:

   7a In ConsoleOne, right-click the Distribution object for the Distribution that you want to reassign, then click *Properties*.

   7b Select the *Type* tab.

   7c In the *Files to be distributed* list, note all of the files or directories to be distributed, including their full paths.

   7d Exit the Distribution object.

8 Using your file location notes and file management software (such as Windows Explorer), copy or move all of the Distribution's files from the current Distributor server's file system to the file system of the Distributor server that is the new owner of the Distribution.

The full paths and filenames must exactly match between both of the Distributor servers' file systems. If you do not make the paths identical between the old and new Distributor servers, you need to edit the Distribution's properties to match the newer paths.

9 Repeat Step 7 and Step 8 for each Distribution to be reassigned.

10 If a Distribution is a Desktop Application type, do the following:

   10a In ConsoleOne, right-click the Distribution object for the Distribution that you want to reassign, then click *Properties*.

   10b Select the *Type* tab.

   10c Note which Application objects are in the Distribution, then note the `.fil` files for each Application object, including their full paths.

   10d Exit the Distribution object.

11 Using your file location notes and file management software (such as Windows Explorer), copy or move all of the Distribution's Application object files from the current Distributor server's file system to the file system of the Distributor server that is the new owner of the Distribution.

The full paths and filenames must exactly match between both of the Distributor servers' file systems. If you do not make the paths identical between the old and new Distributor servers, you need to edit the Application object's properties to match the newer paths.

IMPORTANT: Although you normally have automatic temporary file clean-up for this Distribution, the temporary files for the Distribution being reassigned must be cleaned up manually from the old Distributor's server.

12 Repeat Step 10 and Step 11 for each Distribution to be reassigned.

13 If you did not elect to refresh the Distributors immediately, and you want the new Distributor to now recognize its new Distributions, right-click the new Distributor's object, click *Refresh Distributor*.

The previous Distributor no longer attempts to build the transitioned Distributions. The Distributor that now owns the Distributions is the one to build and send them, according to the Build and Send schedules.

3.4.8 Deleting a Distribution

If you delete a Distribution object, you must immediately refresh the Distributor that owned the Distribution; otherwise, the following can happen:

- When the Build schedule fires, the Distributor tries to build a Distribution that it thinks still exists, causing an error.
- In iManager, if you select the Distribution Information option for the deleted Distribution, the Distributor receives a 601 null-pointer error.

By immediately refreshing the Distributor, you prevent both of these errors from occurring, because:

- The Distributor reads eDirectory when it is refreshed and no longer knows of the deleted Distribution.
- The Distribution Information option for the deleted Distribution is no longer available in iManager.

3.4.9 Removing a Distribution Object - Auto Removal of Temporary Files

Previously, when you deleted a Distribution or Channel object, removed a Distribution or Subscriber from a Channel, or in some way caused one or more Distributions to no longer be associated with one or more Subscribers, the Distributions' temporary files remained on the Subscriber servers, and you had to find them and delete them manually to recover disk space.

In version 7, when a Distributor refreshes, the temporary files of the Distributions that have been removed (either deleted or removed from a Channel) are automatically deleted from Subscribers to free up disk space.

What Causes Temporary Distribution Files To Be Cleaned Up

A Distribution's temporary files are removed from a Subscriber server's file system when:

- The Distribution object is deleted
- The Channel object hosting the Distribution is deleted
- The Distribution is removed from the Channel
- The Subscriber is unsubscribed from the Channel

What the Distributor Does

When a Distributor refreshes, it determines whether any servers (including parent Subscribers in its routing hierarchy) still need to receive any of its Distributions. Where it is found that a Distribution is no longer needed, the Distributor notifies the Subscribers (including parent Subscribers) to clean up that Distribution's temporary files.

If a Distributor cannot contact a Subscriber or has not received a successful deletion reply, it sends another notification to that Subscriber the next time the Distributor refreshes. Therefore, the Refresh schedule determines how often a Subscriber is notified to clean up a deleted Distribution's temporary files.

A Distributor tries five times to notify a Subscriber to clean up a Distribution. If unsuccessful, the Distributor ceases notifying the Subscriber. Then, the temporary files on the Subscriber server can only be cleaned up manually.

What the Subscriber Does

When a Subscriber receives a notification to remove a Distribution's temporary files, it first determines whether the Distribution to be cleaned up is in the process of being received, sent, or extracted by the Subscriber server. If it is not, the Subscriber removes any forwarding or extraction events that are pending and deletes the Distribution's directory containing the temporary files. Then, the Subscriber notifies the Distributor of the removal so that the Distributor can keep track of which Subscribers have successfully complied.

Parent Subscribers are treated the same as end-node Subscribers for cleaning up Distribution files.

Clean Up of Temporary Distribution Files on the Distributor Server

When the Distributor determines that a Distribution object is deleted from eDirectory, the Distribution's version directories (not the Distribution's directory) are automatically deleted from the Distributor's working directory.

After the Distributor has determined that all notified Subscribers have successfully deleted the Distribution directories from their file systems, the Distributor then deletes the Distribution's directory from its file system.

When Subscribers Must Wait to Clean Up Temporary Distribution Files

Temporary Distribution files cannot be deleted from a Subscriber's file system until the association between the Distribution and the Subscriber is broken. For example:

- When a Distribution is listed in the Channel where the Subscriber is subscribed, the Distribution's files can be received and extracted on the Subscriber server:

Figure 3-21   *Temporary Distribution File Cleanup: A*
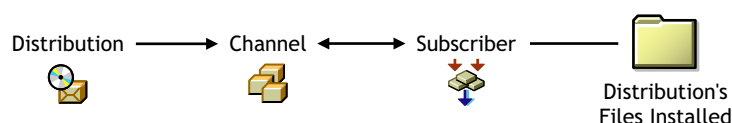
- If the Subscriber is no longer subscribed to the Channel, or the Distribution is no longer listed in the Channel, the Distribution's temporary files can be deleted from the Subscriber server:

Figure 3-22   *Temporary Distribution File Cleanup: B*
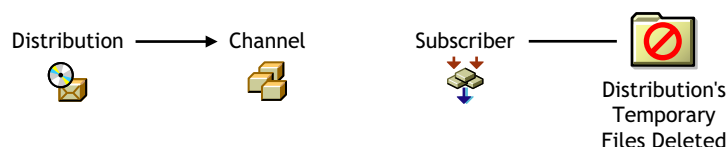
Figure 3-23 *Temporary Distribution File Cleanup: C*
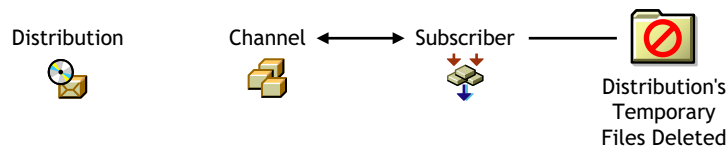
However, if a Distribution and a Subscriber are associated through multiple Channels, the Distribution's temporary files are not deleted from the Subscriber's file system until both the Distribution and Subscriber objects are no longer associated through any Channel. For example:

 ◆ When a Distribution is listed in two different Channels and the Subscriber is subscribed to both Channels, the Distribution's files can be received and extracted on the Subscriber server:

Figure 3-24 *Temporary Distribution File Cleanup: D*
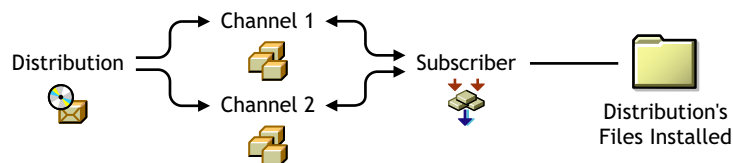
 ◆ When the Distribution is removed from one of the Channels, the Distribution's files can still be received and extracted on the Subscriber server:

Figure 3-25 *Temporary Distribution File Cleanup: E*
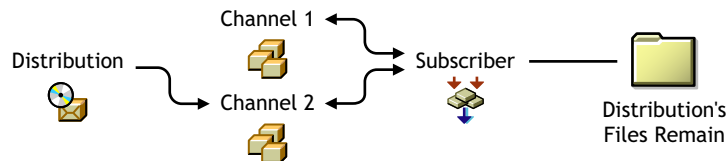
 ◆ When the Distribution is removed from both Channels, the Distribution's temporary files can be deleted from the Subscriber server:

Figure 3-26 *Temporary Distribution File Cleanup: F*
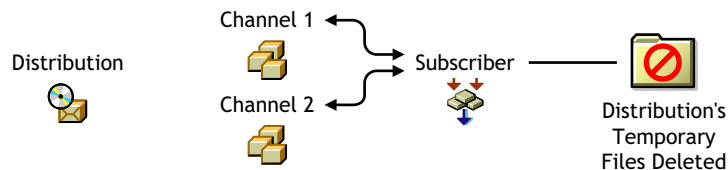
When a Parent Subscriber Can Remove the Temporary Distribution Files

There are two possibilities for when a parent Subscriber can have Distribution files on its server:

 ◆ When both the parent Subscriber and the end-node Subscriber are subscribed to the same Channel to receive its Distributions. The parent Subscriber passes on the Distributions and also extracts them for itself.

 ◆ When the parent Subscriber is not subscribed to the Channel that the end-node Subscriber is. The parent Subscriber only passes on the Channel's Distributions.

In both cases, the following rules apply to when Distribution files can be cleaned up from a parent Subscriber's server:

- If the parent Subscriber unsubscribes to the Channel, the Distributions' files are not deleted from the parent Subscriber's server, so that it can continue to forward those Distributions to the end-node Subscriber server.
- Only after the end-node Subscriber unsubscribes from the Channel is the parent Subscriber able to clean up the Distributions' files from its server.

3.4.10 Handling Orphaned Distributions

The following sections explain how to handle the Distributions of a deleted Distributor object:

- "Orphaned Distributions" on page 141
- "Cleaning Up Orphaned Distributions" on page 141
- "Re-creating Deleted Distributions" on page 142

Orphaned Distributions

Because Distributions belong exclusively to their Distributors, you cannot build and send those Distributions if you delete a Distributor object from eDirectory. The Distributions associated with the deleted Distributor become orphaned and are no longer usable.

Any orphaned Distributions that have already been sent and extracted before you delete the Distributor object are usable by the Subscriber servers where they were extracted. However, these servers no longer receive updated versions of the orphaned Distributions.

You can still see the orphaned Distribution objects in eDirectory, but no current or future Distributor object can be associated with these orphaned Distribution objects.

Cleaning Up Orphaned Distributions

For all Distribution types, you can delete the Distribution's directories on the Subscriber servers' file systems for all orphaned Distributions. We recommend that you delete the Distribution's directories for any Distributions that you intend to re-create.

For most Distribution types, deleting the orphaned Distributions' directories is all you need to do in order to clean up for management and disk space conservation purposes. These Distribution types are:

Desktop Application
File
FTP
HTTP
RPM

However, for the Policy Package and Software Package Distribution types, you might need to undo the processes that the Distributions initiated when they were extracted and installed.

For example, a Policy Package Distribution might require that you use iManager to remove the policies that the Distribution set for the server. For more information, see Step 5 under Section 2.4.3, "Managing the Policy/Package Agent," on page 80.

Re-creating Deleted Distributions

You need to re-create each orphaned Distribution that you want to continue to use. You can do this using an existing Distributor object, or after you install a new Distributor.

After you have re-created a Distribution, all Channels previously associated with the orphaned Distribution need to be associated with the newly created Distribution.

In re-creating the Distributions, you can use the configuration information from the orphaned Distribution objects. When you no longer need the orphaned Distribution objects, you can delete them and they no longer display on the Distributions tab of the Channel object.

3.4.11 Manually Importing and Exporting Distributions

Exporting and importing are useful for:

- Sending a large Distribution to Subscriber servers that are across a slow WAN link from the Distributor server.
- Sending a large Distribution to a parent Subscriber server that is across a slow WAN link, then having that parent pass the Distribution on to its subordinate Subscribers on its side of the WAN.
- Archiving Distributions, and later importing them when and where they are needed again.

The following sections provide information on exporting and importing Distributions:

- "Understanding the Exporting and Importing Processes" on page 142
- "Setting Up Specialized Schedules" on page 143
- "Exporting a Distribution" on page 143
- "Importing a Distribution" on page 143

Understanding the Exporting and Importing Processes

You can manually export a Distribution from a Distributor server by writing to a media source, such as a floppy disk, ZIP disk, CD, or DVD, then you can import it from that media to a Subscriber server.

The export process copies Distribution information to a UNC path or drive mapping, such as a hard drive, floppy disk, or ZIP disk. From the copy on the hard drive, you can then burn the information onto a CD or DVD.

The Distribution information includes the Channel and Distribution data from their eDirectory objects, and the content of the Distribution's file (including all deltas). The Distribution information is copied to a `filename.ted` file that you name when running the Manual Distribution Wizard. You should use the `.ted` extension with the filename. You should also use a descriptive filename so that you can recognize the Distribution when reviewing the media content.

When the exported `.ted` file is imported, the eDirectory object information and the Distribution's content are used to create the Distribution on the Subscriber server's file system. Thereafter, deltas of the Distribution can be sent over the wire, because they are usually much smaller than the original Distribution that was exported and imported.

Distributions can only be exported and imported within the same tree where the associated Channels are known to all Distributors and Subscribers involved.

Setting Up Specialized Schedules

Depending on when you want imported Distributions to be extracted, you might want a different set of schedules set up before exporting the Distribution.

For example, if you want the exported Distribution to be extracted at different times by different Subscribers where it is imported, then:

1 Set the build schedule for the Distribution to be exported to *Immediate*.

2 Add the Distribution to a Channel with a Send schedule set to *Never*.

This prevents Subscribers that have not yet had the Distribution manually imported to them from receiving a Channel's notice to trigger extraction of the yet-to-be-received Distribution.

3 Add all of the Subscribers where the Distribution is to be imported to the Channel you used in Step 2.

4 Refresh the Distributor that owns the Distribution to be exported.

5 After the Distribution has been built, continue with "Exporting a Distribution" on page 143.

If you do not need a specialized schedule, you can just follow the instructions in the next two sections, which assume that existing schedules are acceptable.

Exporting a Distribution

1 In ConsoleOne, click *Tools*, then click *Manual Distribution* to start the Manual Distribution Wizard.

2 Click *Export*, then click *Next*.

3 Select a Channel, select one Distribution from that Channel, then click *Next*.

This Channel's ID is retained in the .ted file for use when importing the Distribution.

4 For the Distribution, provide a path (UNC or drive mapping) and filename (descriptive for identifying which Distribution is on the media), then click *Next*.

The filename should have .ted as its extension.

5 If you are satisfied with the summary, click *Finish*.

The full Distribution is saved as a .ted file to the path that you specified.

6 If your path was to a hard drive, you can now burn the .ted file to a CD or DVD.

Importing a Distribution

1 In ConsoleOne, click *Tools*, then click *Manual Distribution* to start the Manual Distribution Wizard.

2 Click *Import*, then click *Next*.

3 Provide the path and filename to the .ted file, then click *Next*.

This is the .ted file that you exported to a media source.

4 Select parent Subscribers in the top box and individual Subscribers in the bottom box, then click *Next*.

If you select a parent Subscriber that is in the routing hierarchy, all of the Subscribers below it in the hierarchy have the imported Distribution passed on to them, but only if they are already subscribed to the Distribution's Channel.

The Subscribers displayed in the bottom box are those who are currently subscribed to the Distribution's Channel. The heading displays the Channel that is associated with the Distribution being imported. This information is contained in the `.ted` file being imported.

External Subscribers are not listed in the bottom box because they cannot receive manual Distributions.

5 If you are satisfied with the summary, click *Finish*.

The Distribution is copied from the media source you specified and placed in the working directories of the selected Subscribers. The Channel and Distribution objects' information is written to eDirectory.

At this point, imported Distributions are viewable in Remote Web Console in iManager, but not in Tiered Distribution View or Subscriber Distribution View. The next two steps take care of this.

6 If you set up specialized schedules for the imported Distribution (see "Setting Up Specialized Schedules" on page 143), restart the Server Management process on each Subscriber server where it was imported; otherwise, skip to Step 7.

The Distribution is extracted on the Subscriber servers according to their individual Extract schedules. After extraction, you can view the Distribution's information in iManager.

7 To make Distributors recognize that their Subscribers have manually received a new Distribution:

   7a Under the ZENworks Server Management role in iManager, click *Remote Web Console*.

   7b Identify the Distributor owning the imported Distribution in either of the following fields:

      IP Address or DNS Name

      Distributor, Subscriber, or Server Object Name

   7c Click *OK*.

   7d In the *Display* field, select *Tiered Electronic Distribution*.

   7e Click the *Channels* tab, then select *Distribute Channel*.

   7f Click the Channel associated with the imported Distribution, then click *OK*.

      The Distributor begins to send the Distribution listed in the Channel to the Subscribers, but the Subscribers reply that they already have the Distribution, then begin to extract the imported Distribution.

      If a Subscriber is a parent Subscriber that needs to pass the imported Distribution on to subordinate Subscribers, it does so when the Distribution's Channel starts.

3.4.12 Using the Distribution Wizard

Server Management provides the Distribution Wizard to help you learn the process involved in creating and sending a Distribution. You can use this wizard to create and send either a File or FTP Distribution.

To use the Distribution Wizard:

1 In ConsoleOne, select the container where you want the Distribution object created, click *Tools*, then select *Distribution Wizard*.

2 Review the information on the Introduction page, then click *Next*.

3 On the Distributor Selection page, browse for and select the Distributor that owns this File or FTP Distribution, then click *Next*.

4 On the Subscriber Selection page, click *Add*, browse for the Subscribers to receive this Distribution, click *Select*, click *OK*, then click *Next*.

5 On the File Source page, select the file source (the Distributor's file system, or a remote FTP site), then click *Next*.

6 On the Destination Volume or Drive page, select an option and fill in its field, then click *Next*.

Use the same volume or drive for all Subscribers: If each target Subscriber is to have the exact same volume or drive available, select this option and provide the volume label or drive letter.

Use a variable for the volume or drive: If your target Subscribers are using different paths (for example you have NetWare, Windows, Linux, and Solaris Subscriber servers), you can provide a variable value. This value must be defined on each Subscriber in order to receive the Distribution.

7 On the Additional Destination Directories page, provide any additional path information for the target Subscriber servers, then click *Next*.

Your path information is displayed under the *Data Will Be Placed In Path* heading as you enter it. Use this information to verify that the path is valid before continuing.

8 On the File Selection From Distributor Server page, click *Add*, browse for the files or directories to be included, click *Select*, click *OK*, then click *Next*.

You are browsing the Distributor's file system, not the local machine's.

Repeat clicking *Add* until you have all of the files and directories you want in this Distribution.

9 On the Distribution Name and Context page, fill in the fields, then click *Next*.

Distribution name: Provide a unique name for the Distribution.

Context: Browse for and select the container where you want the Distribution object to be created.

10 On the Additional Options page, select or deselect the options as applicable, then click *Next*.

The following options are all selected by default:

Copy the Distributor's security certificate to all Subscribers: This is necessary for the Subscriber to be able to receive and extract this Distribution. This might not be necessary if you run the wizard again with the same Distributor and Subscribers.

Verify that all Subscribers are up and running: If you want to make sure your target Subscribers can receive this Distribution, select this option.

Notify the Distributor to read eDirectory for new information: This causes the Distribution to be built immediately.

11 On the Summary page, review the steps that are take by the Distribution Wizard, then click *Finish* to create the Distribution.

Information is displayed as the Distribution is created and sent.

12 To review the log file, select *Yes* when prompted.

If you select *Yes*, you can review the log file. Click *Close* to exit the log window and the Distribution Wizard.

If you select *No*, the Distribution Wizard is exited.

3.5 Channels

The following sections provide concepts and instructions for the Channel object:

3.5.1 Understanding Channels

The Channel object (TED Channel) contains a list of Distributions associated with it and Subscribers subscribed to it.

Functional Relationship with Other Tiered Electronic Distribution Objects

Figure 3-27 illustrates a Channel's relationship with Distributions and Subscribers:

Figure 3-27   *Channel Relationship with Distributions and Subscribers*

Multiple Distributions ⟶ **Channel** ⟵ Multiple Subscribers
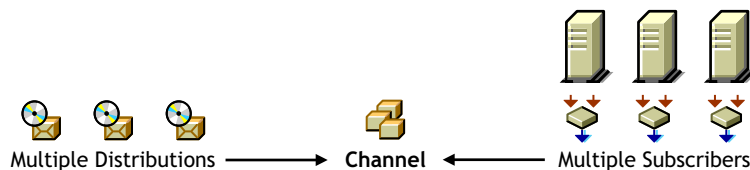
The Distributions are listed in the Channel, and the Subscribers subscribe to the Channel.

Channel Description

Distributors can list one Distribution in multiple Channels, and multiple Distributors can list their Distributions in the same Channel.

You can have as many Channels as you want. Channels do not hold the actual Distributions, only a reference to them. There is no limit to the number of Distribution references a Channel can send. The practical limit is how many Distributions you want to track per Channel.

Scheduling

A Channel's Send schedule determines when a Distribution are sent from the Distributor to its Subscribers.

A Channel can be active or inactive to control when its Distributions are sent.

For information on how time zones can affect scheduling between a Channel and its associated Distributors and Subscribers, see "Scheduling Tiered Electronic Distribution Objects in Different Time Zones" on page 342.

Subscriptions to Channels

Channels can be subscribed to by multiple Subscribers.

To receive a Distribution, a Subscriber must subscribe to the Channel where that Distribution is listed. However, a Subscriber receives all of the Distributions listed in that Channel, which means they are applied to the Subscriber server when they are extracted.

3.5.2 Creating and Configuring Channels

The following sections provide you with the steps to create and configure the Tiered Electronic Distribution objects with ConsoleOne.

Do the following in order for each Distributor:

- "Determining the Channel Names" on page 147
- "Creating the Channel Objects" on page 147
- "Configuring the Channels" on page 148

Determining the Channel Names

In naming Channels, use a descriptive method. For example:

```
VirusProtect
VProtectPatterns
VirusProtection
NW51patch4
NW6patch1
AUTOEXECNCF000326
```

You can manage your Channels more easily by:

- Using names that are purpose oriented
- Using a similar name for the Channel and its Distributions

Continue with "Creating the Channel Objects" on page 147.

Creating the Channel Objects

Channels are used to group Distributions and establish a schedule for passing a Distributor's Distributions to Subscribers that are subscribed to the Channel. A Channel can have Distributions from many Distributors. A Channel can be subscribed to by many Subscribers.

To create a Channel object:

1 In ConsoleOne, select a container object to hold the Channel object, click *File > New > Object*, then select *TED Channel*.

2 Provide a name for the Channel object and click *OK*.

3 Create as many Channel objects as needed to group Distributions by type and/or send schedule.

4 Continue with "Configuring the Channels" on page 148.

Configuring the Channels

You need to configure a Channel object before you can begin using it.

Not all properties associated with the Channel object are required. Required objects are noted; all others are optional.

To configure the Channel object:

1 In ConsoleOne, right-click the Channel object, then click *Properties*.

2 Select the *General* tab and fill in the fields:

Active: Select the check box to enable the Channel to pass on its Distributions.

Description: Provide a useful description, such as what Distributions the Channel is associated with.

3 Select the Distributions tab, then click *Add* to add Distributions.

Distributions: A list of Distributions that are associated with this Channel. For information on creating Distribution packages, see Section 3.4, "Distributions," on page 111.

4 Select the *Subscribers* tab, then click *Add* to add Subscribers to the Channel.

Subscribers subscribed to this Channel: A list of Subscribers and External Subscribers that are subscribed to this Channel.

5 Select the *Schedule* tab, then select a schedule for when to distribute the Channel's Distributions.

For information on available schedules, see Chapter 8, "Scheduling," on page 325.

3.5.3 Forcing a Channel To Be Sent

If you want to send all of the Distributions in a Channel outside of Channel's the normal Send schedule, you can manually force the distribution process.

Assuming that a new Distribution has been built and the Channel's Send schedule is not ready to fire, do one of the following to force a Channel to be sent:

◆ Using the ZENworks Server Management role in iManager, click *Edit TED Object*, browse for and select the Channel, click *OK*, then click *Distribute Channel*.

◆ In ConsoleOne, you have a two-step process:

a. Select the Channel object, click *Properties*, select the *Schedule* tab, select *Run Immediately*, click *OK*, right-click the Distributor object, then click *Refresh Distributor*.

b. After the Distribution has been sent, to reverse the changes made in Step a, select the Channel object, click *Properties*, select the *Schedule* tab, select the schedule that the Channel previously had, then click *OK*.

As soon as a Subscriber receives an entire Distribution, it extracts according to the Subscriber's Extract schedule.

3.6 Subscribers

The following sections provide concepts and instructions for the Subscriber object:

3.6.1 Understanding Subscribers

The Subscriber object (TED Subscriber) is an eDirectory object that defines the properties for the Subscriber.

Functional Relationship with Other Tiered Electronic Distribution Objects

Figure 3-28 illustrates a Subscriber's relationship with the Channels:

Figure 3-28 *A Subscriber that Subscribes to Multiple Channels*
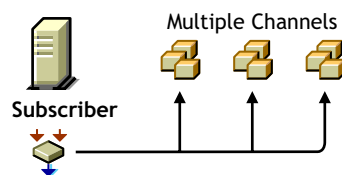
The Subscriber subscribes to the Channels.

Figure 3-29 illustrates the Subscriber's relationship with Distributors and Distributions:

Figure 3-29   *Multiple Subscribers Can Receive the Same Distribution from a Distributor.*
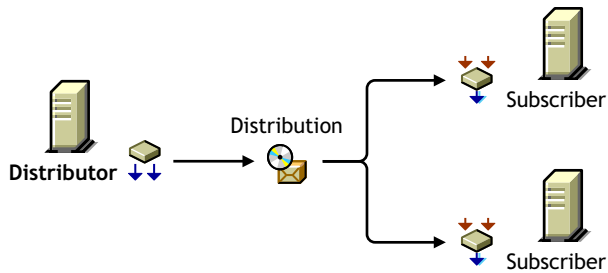
Subscriber Description

The Subscriber is a service that receives and extracts Distributions to obtain the software, files, or policies it needs.

Any server where you want to distribute applications, files, or policy packages must have the Subscriber software installed and a Subscriber object in the eDirectory tree. The Subscriber object can be in a different tree than the server's NCP™ server object, because IP addresses or DNS names are used for moving Distribution files to the Subscriber servers.

Distributions are copied to the Subscriber server's hard drive. The Subscriber receives the Distributions and extracts them to install the software, files, or policies.

Scheduling

A Subscriber's Extract schedule determines when it can extract its Distributions.

For information on scheduling, see Chapter 8, "Scheduling," on page 325.

Subscribing to Channels

Subscribers can subscribe to a Channel to receive all of the Distributions listed in that Channel. A Subscriber object's properties lists the Channels it is subscribed to.

Subscribers can receive Distributions from multiple Distributors because:

- Multiple Distributors can list their Distributions in the same Channel
- Subscribers can subscribe to multiple Channels

Parent Subscribers

Subscribers can be parent Subscribers, which are proxies for the Distributor to pass Distributions to other Subscribers. This helps the Distributor by providing load-balancing for sending Distributions to many Subscribers.

The Subscriber object's properties lists the parent Subscriber through which it receives all of its Distributions. A Subscriber can receive its Distributions directly from the Distributor if it does not have a parent Subscriber and is not listed in the Distributor's routing hierarchy.

Parent Subscribers can also be used to bridge WAN links to ensure that Distribution packages are sent across WAN links a minimum number of times.

Special Character Handling

Syntax differences (such as characters that are invalid to a platform) are now handled for each supported platform. For invalid characters, the agent properly gathers all files, regardless of platform of the Distributor server. The Subscriber server detects whether files in the Distribution package include invalid characters and ignores or skips files during extraction. Skipped files are logged. Previously, the whole Distribution would fail to extract and be installed.

Linux and Solaris support characters in file and directory names that NetWare and Windows do not recognize.

3.6.2 Creating Subscribers

Subscribers must be created by installing their software and eDirectory objects using the *ZENworks 7 Server Management with Support Pack 1 Program* CD. For more information, see "Installation on NetWare and Windows Servers" in the *Novell ZENworks 7 Server Management Installation Guide*.

If a Subscriber object is inadvertently deleted, you can re-create it in ConsoleOne. However, the Revision Number of the new Subscriber object will be less than its Revision Number in the `ted.cfg` file. Therefore, the Subscriber cannot accept any updates to its configuration, because the lower Revision Number causes it to assume that the configuration data is older than what it has. To resolve this problem, delete the `ted.cfg` file on the Subscriber server, and the next time a Distribution is sent to the Subscriber, a new configuration is accepted, and a new `ted.cfg` file created.

3.6.3 Configuring Subscribers

Subscriber objects are automatically created when you install the Subscriber software to a server.

Not all properties associated with the Subscriber object are required. Required objects are noted; all others are optional.

To configure the Subscriber object's properties:

1 In ConsoleOne, right-click the Subscriber object, then click *Properties*.

2 Click *General > Settings* and fill in the following fields:

Use policy: Select to use the effective policy if you want to use the values set in the Tiered Electronic Distribution policy. This field is displayed if a Tiered Electronic Distribution policy has been created, distributed to the Subscriber server, extracted by the Policy/Package Agent, and enforced on the server.

If you select this option, the rest of the fields are dimmed and the policy settings are used instead. The current policy is displayed in parentheses.

Input rate: The rate Distributions are received. The default is the maximum that the connection can handle. This rate is used to control a Subscriber server's use of narrow bandwidth links.

Output rates based upon Distribution's priority: Sets the default output rate to minimize network traffic for Tiered Electronic Distribution objects. This determines the send rate for parent Subscribers to its subordinate Subscribers. The default value is the maximum that the connection can handle. Blank means that bandwidth is taken from third-party applications.

There are three output priorities where you can specify a rate:

- ◆ **High priority:** These Distributions are sent before any Medium or Low priority Distributions.

- ◆ **Medium priority:** These Distributions are sent after all High priority and before any Low priority Distributions.

- ◆ **Low priority:** These Distributions are sent after all High and Medium priority Distributions.

For more information, see .

Maximum concurrent Distributions to send: Specifies the maximum number of distribution threads that can be running concurrently for sending on Distributions. The default value is unlimited (a blank field).

This applies only to parent Subscribers that pass on Distributions to subordinate Subscribers.

Connection time-out: Specifies the number of seconds a Subscriber waits for a response from a Distributor (receiving) or a Subscriber (sending) before ending the connection. If a connection is ended during sending or receiving, the send does not start again until the next time the Channel schedule starts. It then picks up where it left off.

The default value is 300 seconds (five minutes). The available range in seconds is 1 to 60,000. You should make this setting a reasonable time to wait for a response from one node to another.

This interval should be increased on slow or busy links where longer delays are frequent.

Working directory: Specifies the directory to be used by the Distribution. It contains Distributions, persistent status, and temporary working files. The default path is:

- ◆ **NetWare:** `sys:\zenworks\pds\ted\sub`

  > **IMPORTANT:** The default volume is sys: on NetWare servers. We recommend that you do not use the sys: volume because the content of this directory can become quite large.

- ◆ **Windows:** `c:\zenworks\pds\ted\sub`

- ◆ **Linux and Solaris:** `/var/opt/zenworks/zfs/pds/ted/sub`

For more information on the working directory, see .

Parent Subscriber (optional): Specifies a parent Subscriber from which Distributions are received.

This field is where you can enable efficient distribution from a Distributor to its Subscribers. The routing information in a Distributor object's properties accounts only for parent Subscribers (the tiered distribution model). End-node Subscribers (most of the Subscribers in your tree) should not be listed there.

This field allows you to specify for each end-node Subscriber that it receives its Distributions via a specific parent Subscriber, instead of directly from the Distributor. This reduces the workload on the Distributor server, and provides the tiered distribution model for efficient sending of Distributions.

This field is also useful for allowing a parent Subscriber to send a Distribution to an External Subscriber's server in another tree.

Disk space desired to be left free: Use this value to ensure there is enough free disk space for receiving Distributions. A Subscriber does not attempt to receive a Distribution if the disk space value set here is insufficient.

3 Click *General > Messaging* and fill in the following fields:

IMPORTANT: If this Subscriber is on the same server as a Distributor, entries in these fields are ignored. Only the Distributor's messaging settings are used.

Use policy: Select to use the effective policy if you want to use the values set in the Tiered Electronic Distribution policy. This field is displayed if a Tiered Electronic Distribution policy has been created, distributed to the Subscriber server, extracted by the Policy/Package Agent, and enforced on the server.

If you select this option, the rest of the fields are dimmed and the policy settings for messaging are used instead. The current policy is displayed in parentheses.

Server console: Specifies the level of output messages to send to the Subscriber console on the server console.

For more information on the message notification levels, see Section 3.11.5, "Minimizing Messaging Traffic," on page 186.

SNMP trap: Specifies the level of messages to send via SNMP.

Log File: Specifies the level of messages to send to the log file.

Path and filename: You can specify a custom log file's name and location for this Subscriber object. The default is:

* **NetWare:** `sys:\zenworks\pds\ted\dist\ted.log`

  IMPORTANT: The default volume is sys: on NetWare servers. We recommend that you do not use the sys: volume because the log file can become quite large.

  ---

* **Windows:** `c:\zenworks\pds\ted\dist\ted.log`

* **Linux and Solaris:** `/var/opt/zenworks/zfs/pds/ted/dist/ted.log`

This is the same log file that the Distributor uses.

Delete log entries older than __ days: Log file entries for a Subscriber are deleted after they are older than the number of days specified. The default is six days.

E-mail: Specifies which level of messages to send via e-mail.

Users: Specifies e-mail users for notification.

Address attribute: Specifies e-mail addresses for notification.

You can add users or groups stored in eDirectory or provide the e-mail addresses for users who are not contained in eDirectory. The e-mail Address Attribute associated with an eDirectory user is the default attribute.

IMPORTANT: If you select e-mail as a method for receiving notification, be aware that additional network traffic can be created.

---

4 Click *General > Working Context* and browse for a working context.

This is the eDirectory context where the Subscriber creates the objects related to the Desktop Application Distributions it receives.

5 Select the *Schedules* tab, select a schedule, then fill in the fields:

Use policy: Select to use the effective policy if you want to use the values set in the Tiered Electronic Distribution policy. This field is displayed if a Tiered Electronic Distribution policy has been created, distributed to the Subscriber server, extracted by the Policy/Package Agent,

and enforced on the server. If you select this option, the rest of the fields are dimmed and the policy settings for scheduling are used instead.

Schedule type: This schedule determines when the Subscriber extracts the Distributions.

For information on available schedules, see Chapter 8, "Scheduling," on page 325.

6 Select the *Channels* tab and fill in the fields:

* **Channels this Subscriber is subscribed to**

  Lists the Channels the Subscriber is subscribed to.

  Active: To activate a Channel for this Subscriber server so it can receive the Channel's Distributions, click a Channel, then select the check box to enable it. To deactivate a Channel so that the Subscriber does not receive the Channel's Distributions, deselect the check box to disable it.

  Channel: Click Add to create a Channel. Click Details to edit a Channel.

* **Channels subscribed to through Subscriber Group memberships**

  Lists the Subscriber Groups that the Subscriber is a member of, paired with which Channels the Subscriber is subscribed to by virtue of membership in a Subscriber Group.

  These columns are for display only. The Details, Add, and Delete buttons do not apply.

  Active: Indicates whether the Channel subscribed to is active.

  Channel: Displays the Channel subscribed to through membership in a group.

  Subscriber Groups: Displays the groups the Subscriber is a member of. You can sort the listing by clicking the column heading.

7 Select the *Variables* tab and fill in the fields:

Include policy: Select to use the effective policy if you want to use the values set in the Tiered Electronic Distribution policy. This field is displayed if a Tiered Electronic Distribution policy has been created, distributed to the Subscriber server, extracted by the Policy/Package Agent, and enforced on the server.

If you select this option, the variables specified in the Tiered Electronic Distribution policy are added to the list of variables. However, if there are duplicate variables, the variables in the Subscriber prevail.

Variable: Name of the variable. It should indicate how the variable is used. For example, WORKINGVOL.

Value: The value that the Subscriber uses when this variable is specified. For example, data:.

To ensure that extraction takes place, provide an absolute path to the Subscriber. For example, if the path is only the data: volume, make sure the colon (:) is included, because it is a necessary part of the full path.

Description: Describes how the variable is used. For example:
```
Volume for the working directory.
```

For information on variables, see Section 9.6, "Using Variables to Control File Extraction," on page 357.

8 To include this Subscriber in a group, click *Group Membership*, click *Add*, browse for a Subscriber Group object, click *Select*, then click *OK*.

9 When you are finished configuring the Subscriber object, click *OK* to exit the Subscriber object's properties.

3.6.4 Updating Subscriber Configurations

The Subscriber software cannot run on a server if the Subscriber does not know its Tiered Electronic Distribution configuration, such as where it's working directory is. Therefore, during the installation process, you determine a basic Tiered Electronic Distribution configuration for each of the Subscribers that you are installing.

Using this input, the installation program creates a `tednode.properties` file on each Subscriber server that contains the Subscriber's initial Tiered Electronic Distribution configuration. Until a server receives its first Distribution, this `tednode.properties` file provides the server with its Tiered Electronic Distribution configuration information, so that it can function as a Subscriber.

A Subscriber server can only receive configuration information from a Distributor server whose Distributor object is in the same tree as the server's Subscriber object. This is known as the trusted tree, which is established during the installation process. For information on knowing when the trusted tree is necessary, see "Subscriber Software Configuration and Trusted Trees" on page 161.

When a Distributor server sends a Distribution to a Subscriber server, the Distributor first checks to see if that Subscriber server has a current Tiered Electronic Distribution configuration in the form of a `tcpip.nlm` file. If this is the first time the Subscriber has received a Distribution, it does not have that file. The Distributor then sends the tcpip.nlm file to the Subscriber, and the `tednode.properties` file is no longer used by the Subscriber. Then the Distributor checks again to see if the Subscriber server has a current `tcpip.nlm` file. Upon confirmation from the Subscriber, the Distribution is sent. In other words, the Distributor never sends a Distribution to a Subscriber server whose configuration information is not current.

You can update the `tcpip.nlm` file any time you make configuration changes to the Subscriber object's properties. However, Subscribers do not read eDirectory, so when a change is made to the Subscriber, it must rely on the Distributor server to discover those changes and send the new configuration information to the Subscriber server, updating its `tcpip.nlm` file.

If you should install the Subscriber software to a server that does not have a Subscriber object in any eDirectory tree, such as a Microsoft domain server, the `tednode.properties` file is used by such servers, in lieu of having its Tiered Electronic Distribution configuration updated by a Distributor server. In this case, for configuration changes, you need to edit the server's `tednode.properties` file. For more information, see "The Tednode.properties File Requirement" on page 163 and Section 3.13, "Editing the Tednode.properties File," on page 193.

3.6.5 Associating Subscribers with Channels

Before a Subscriber can receive a Distribution, you need to associate the Subscriber to the Channel holding the Distribution. You can do this either from the Subscriber or Channel object's properties:

- ◆ "Associating a Channel with Multiple Subscribers" on page 156
- ◆ "Associating a Subscriber with Multiple Channels" on page 156

Associating a Channel with Multiple Subscribers

To send a particular Distribution to many Subscriber servers:

1 In ConsoleOne, right-click the Channel object where the Distribution is listed, then click *Properties*.

2 Select the *Subscribers* tab, click *Add*, then add the needed Subscribers.

3 Select the *Schedule* tab and select a schedule.

The schedule determines when Distributions that have been received are extracted or installed.

For information on the available schedules, see Chapter 8, "Scheduling," on page 325.

4 Click *OK* to save the changes.

Associating a Subscriber with Multiple Channels

To subscribe a Subscriber server to multiple Channels for receiving different Distributions:

1 In ConsoleOne, right-click the Subscriber object, then click *Properties*.

2 Select the *Channels* tab, click *Add*, then add the needed Channels.

3 Select the *Schedule* tab and select a schedule.

The schedule determines when Distributions that have been received are extracted or installed.

For information on the available schedules, see Chapter 8, "Scheduling," on page 325.

4 Select the *Variables* tab, fill in the following fields, then click *OK*:

Variable name: Can be used to determine the location of the destination directory where the files are extracted. Enter the name of the variable exactly as you are using it within the %...% symbols.

Value: This is the value of the variable, which can be another variable's name.

Description: Text field to provide details about the variable.

For information on variables, see Section 9.6, "Using Variables to Control File Extraction," on page 357.

5 Click *OK* to save the changes.

3.6.6 Deleting Subscriber Objects That Are Part of a Distributor's Routing Hierarchy

If a Subscriber object is removed from eDirectory, or a Subscriber server is removed from the network (whether its Subscriber object is also removed or left in eDirectory), and that Subscriber was part of a Distributor's routing hierarchy, you need to edit the Distributor object's properties to adjust the routing hierarchy accordingly. Otherwise, Distributions that are sent through that parent Subscriber do not reach the designated Subscriber servers.

3.7 Subscriber Groups

The following sections provide concepts and instructions for the Subscriber Group object:

- Section 3.7.1, "Understanding Subscriber Groups," on page 157
- Section 3.7.2, "Creating and Configuring Subscriber Groups," on page 158

3.7.1 Understanding Subscriber Groups

A Subscriber Group is an eDirectory object (TED Subscriber Group) used for grouping Subscribers objects.

- "Functional Relationship with Other Tiered Electronic Distribution Objects" on page 157
- "Subscriber Group Description" on page 157
- "Scheduling" on page 157

Functional Relationship with Other Tiered Electronic Distribution Objects

Figure 3-30 illustrates a Subscriber Group's relationship with Subscribers and Channels.

Figure 3-30   *Using a Subscriber Group*
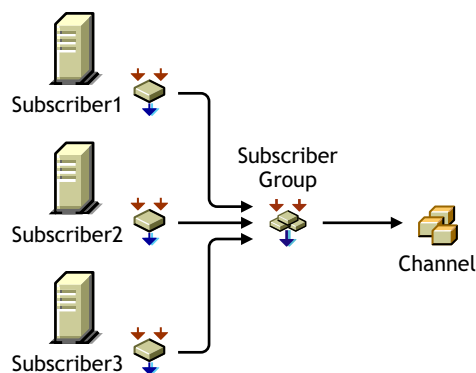
Subscriber Group Description

A Subscriber Group is used for grouping Subscribers that have the same Distribution needs.

Subscriber Groups are useful when you are sending several different Distributions to the same set of Subscribers. There is no need to create a Subscriber Group if it is only associated with one Channel.

For example, Distribution A is in Channel A, Distribution B is in Channel B, and so on. Then, without using a Subscriber Group, you need to subscribe each of your Subscribers to Channel A, then each to Channel B, and so on, which could be a very long process. However, by using a Subscriber Group, you only need to create the group, add the Subscribers to it, then subscribe that one group to each Channel.

Another use of a Subscriber Group is that when the group is associated with two or more Channels, you can edit the group's membership more easily than making the same changes in multiple Channels. For example, to remove a Subscriber from one Subscriber Group, you just edit that one group's properties. To remove that same Subscriber from several Channels, you need to edit each Channel's properties.

Scheduling

Subscriber Groups are not scheduled.

3.7.2 Creating and Configuring Subscriber Groups

1 In ConsoleOne, select the container to hold the Subscriber Group object, click *File > New > Object*, then select *TED Subscriber Group*.

2 In the New TED Subscriber Group dialog box, provide a name for the Subscribe Group (worksheet item 17), select *Define additional properties*, then click *OK*.

3 Click *General > Settings* and provide a description.

4 To populate the group with Subscribers, select the *Members* tab and do the following:

  4a Click *Add*, browse for and select the Subscriber objects (worksheet item 18), then click *OK*.

  4b To remove any Subscribers from the list, select the Subscribers, then click *Delete*.

  4c To view the properties of any Subscriber, select the Subscriber, then click *Details*.

5 Click *OK* when you have finished configuring the Subscriber Group object.

3.8 External Subscribers

The following sections provide concepts and instructions for the External Subscriber object:

- Section 3.8.1, "Understanding External Subscribers," on page 158
- Section 3.8.2, "Using External Subscribers for Out-of-Tree Distributions," on page 163
- Section 3.8.3, "Creating and Configuring External Subscribers," on page 166

3.8.1 Understanding External Subscribers

An External Subscriber is an eDirectory object (TED External Subscriber) that represents a Subscriber object in another tree.

- "Functional Relationship with Other Tiered Electronic Distribution Objects" on page 158
- "External Subscriber Description" on page 159
- "Subscriber Software Configuration and Trusted Trees" on page 161
- "Scheduling" on page 163

Functional Relationship with Other Tiered Electronic Distribution Objects

Figure 3-31 illustrates an External Subscriber's relationship with the Channel:

Figure 3-31   *External Subscriber's Relationship with a Channel*
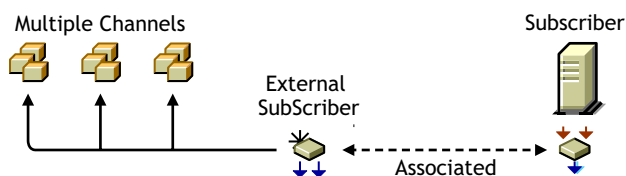
The External Subscriber subscribes to the Channels.

External Subscriber Description

A Distributor cannot send its Distributions to a Subscriber server whose Subscriber object is in a different tree than the Distributor's object, or to a server that does not have a Subscriber object. An External Subscriber object is needed for out-of-tree distributions.

For information on the External Subscriber object, see the following:

The External Subscriber's Purpose

If you installed all of your Tiered Electronic Distribution objects in one tree, an External Subscriber object is not necessary, because you can send your Distributions using the Distributor and Subscriber objects that are in the same tree.

However, the External Subscriber object is useful for sending out-of-tree Distributions when one of the following conditions exists:

- **The target server has no Subscriber object in any tree:** The target server, such as a Windows server in a Microsoft domain, has only the Subscriber software installed on it.
- **The target server has a Subscriber object in a different tree:** The target server has the Subscriber software installed on it, but its Subscriber object is in a different tree than the Distributor object that is sending the Distribution.

Because the External Subscriber is only an object in an eDirectory tree, it does not actually handle the Distribution files; it simply identifies which server is to receive them.

Distribution Information Not Maintained

When sending any Distribution through External Subscribers, trusted tree rights cannot be maintained.

When sending Desktop Application Distributions through External Subscribers, application object associations cannot be maintained. However, it can send the group association, because it creates that.

Duplicate Distribution Management

You can use an External Subscriber object to circumvent the need to duplicate Distribution work in another tree.

For example, a few Subscribers on a tree at a remote site could receive all of their Distributions via the External Subscriber in the Distributor's tree. That would prevent the need to have a Distributor server at the remote site, including duplicating the Distribution configuration and management effort there.

External Subscriber Characteristics

An External Subscriber is associated with a server running the Subscriber software that has no Subscriber object in any tree, or no Subscriber object in the same eDirectory tree as the Distributor from which it receives the Distribution.

External Subscriber objects are associated with a Subscriber server through an IP address or DNS name of that server.

You can send Distributions outside of eDirectory, such as to a Windows server in a Microsoft domain. For more information on this type of Distribution, see "Subscriber Software Configuration and Trusted Trees" on page 161 and "The Tednode.properties File Requirement" on page 163.

External Subscriber objects cannot be parent Subscribers. If an External Subscriber has a parent Subscriber, both the External Subscriber's and parent Subscriber's objects must reside in the same tree.

External Subscriber Requirements

If a target server's Subscriber object is in a different tree from the Distributor object of the server that sends it a Distribution, that target server must be represented by an External Subscriber object in the Distributor's tree.

Because Tiered Electronic Distribution uses IP addresses or DNS names to locate servers, Subscriber objects can be in a different tree than those servers' NCP objects.

An External Subscriber must be subscribed to the Channel that lists the Distributions needed by its associated Subscriber.

The server receiving a Distribution via an External Subscriber must have the Subscriber software installed on it so it can receive and extract the Distribution. It is not required to have a Subscriber object in any tree, such as if it is a Windows server in a domain (see "Subscriber Software Configuration and Trusted Trees" on page 161 and "The Tednode.properties File Requirement" on page 163).

The External Subscriber Object's Properties

The External Subscriber object properties contain only the following:

◆ IP address or DNS name of the Subscriber server in a different tree or a domain (required)

This is the ID of the Subscriber server in one tree that is to receive a Distribution from a Distributor in another tree (the tree where the External Subscriber object resides).

◆ The Channels it is subscribed to (required)

This is for identifying which Distributions need to be sent to the Subscriber server in the other tree.

◆ Membership in a Subscriber Group (optional)

You can use this for subscribing the External Subscriber to the Channels subscribed to by the group.

- Context of a parent Subscriber in the External Subscriber's own tree (optional)

  A parent Subscriber is usually in the Distributor's distribution hierarchy.

  If used, the parent Subscriber does the physical work in sending the Distribution file to the server in the other tree. Otherwise, the Distributor server sends the Distribution directly to the Subscriber server in the other tree.

Subscriber Software Configuration and Trusted Trees

Subscribers can be configured by a Distributor, but External Subscribers cannot. External Subscribers are just objects identifying a server. However, a Subscriber server identified by an External Subscriber object must have a Tiered Electronic Distribution configuration in order to receive the Distributions via the External Subscriber object.

Using the External Subscriber object brings up the need to understand trusted trees:

- "The Reason for Trusted Trees" on page 161
- "Determining the Trusted Tree" on page 162
- "The Tednode.properties File Requirement" on page 163
- "Preventing Trusted Tree Errors for Policy Package Distributions" on page 163

The Reason for Trusted Trees

The following applies to any NetWare or Windows server, whether it has an NCP server object in an eDirectory tree or a server object in a Microsoft domain:

- During installation, the server can have both a Subscriber object created for it and the Subscriber software installed to it
- During installation, the server can have only the Subscriber software installed to it (no Subscriber object is created)
- During installation, you should identify the trusted tree of any server that does not have a Subscriber object created for it

Identifying a trusted tree has two purposes:

- To locate a Distributor that can update the Subscriber's Tiered Electronic Distribution configuration information
- To indicate which tree to accept policies from

A Subscriber server's Tiered Electronic Distribution configuration information is stored in eDirectory in its Subscriber object (which the Distributor reads), and in a `tcpip.nlm` file in the Subscriber server's file system (which the Subscriber reads). A Distributor server sending the configuration information must have its Distributor object in the same tree as the Subscriber object that it is configuring.

A Subscriber server can receive its Subscriber software configuration only from a Distributor in its trusted tree. The trusted tree is where the server's Subscriber object and that Distributor object both reside. This is not the tree where an associated External Subscriber object resides, and it doesn't matter whether it's the same tree where the server's NCP object resides.

A Subscriber server that does not have a Subscriber object in any tree (such as a Windows server in a Microsoft domain), must use its `tednode.properties` file for its Tiered Electronic Distribution configuration information. This file is created on the server when you installed the

Subscriber software. Then it can receive and extract Distributions from a Distributor in another tree (via an External Subscriber object). The extraction process is the time when the trusted tree requirement must be met. For more information, see "The Tednode.properties File Requirement" on page 163.

Determining the Trusted Tree

There are two situations that deal with whether to install Subscriber objects for Subscriber servers:

- **eDirectory server:** When you install the Subscriber software to a server whose NCP server object is in another tree, you have one of the following options:
  - You can create the Subscriber object in the Distributor's tree, which might not be the tree where the Subscriber's NCP server object resides (the server's Subscriber and NCP objects do not need to be in the same tree). In this case, you do not need an External Subscriber object for sending Distributions to that Subscriber, because its object is not out-of-tree.

    The Subscriber server's trusted tree is the same tree where the Distributor object resides. Therefore, it receives its Tiered Electronic Distribution configuration updates from the Distributor in its trusted tree.
  - You can elect to not create a Subscriber object for the server. In this case, you need to use the `tednode.properties` file to configure that Subscriber server. You also need to use an External Subscriber object to send Distributions to that server.

    In order for this Subscriber to have policies enforced on it, you need to identify its trusted tree, which would be the tree it receives Policy Package Distributions from.

- **Non-eDirectory server:** When you install the Subscriber software to a server that is in a Microsoft domain, and therefore does not have an NCP server object in any eDirectory tree, you can create a Subscriber object for this server, but it is not required. If you do not have a Subscriber object, you need to use the `tednode.properties` file to configure that Subscriber server. You also need to use an External Subscriber object to send Distributions to this server.

    In order for this Subscriber to have policies enforced on it, you need to identify its trusted tree, which would be the tree it receives Policy Package Distributions from.

The File Installation Paths and Options page in the installation program contains the Trusted Tree field. However, this field is only be displayed if you deselect the Create eDirectory Objects check box on the Installation Options page. This causes the installation program to install only software for the selected servers.

You must select a trusted tree for each server where you have selected to install the Subscriber software, or your Policy Package Distributions might not extract on that Subscriber server, because policies point to objects in a tree.

For installation instructions concerning the Trusted Tree field, see the steps in the applicable sections under "Installation on NetWare and Windows Servers" in the *Novell ZENworks 7 Server Management Installation Guide*.

The Tednode.properties File Requirement

A `tednode.properties` file must be used to provide configuration information for the following Subscriber servers:

- A Subscriber server that has a Subscriber object and has not yet received its first Distribution. After it does, it then uses the `tcpip.nlm` file given to it by the Distributor in its trusted tree that is sending that first Distribution, and it no longer uses the `tednode.properties` file.

  A Subscriber can only be configured by a Distributor server whose object is in the same tree as the Subscriber's object.

- A Subscriber server that does not have a Subscriber object in any tree.

  This could be a Windows server in a Microsoft domain where you only installed the Subscriber software without creating the object.

  If you installed the Subscriber software (using the ZENworks 7 Server Management installation program) without creating the Subscriber object, the `tednode.properties` file was automatically created and configured.

  For more information, see Section 3.13, "Editing the Tednode.properties File," on page 193.

Preventing Trusted Tree Errors for Policy Package Distributions

In order to prevent trusted tree errors when sending a Policy Package Distribution to a Subscriber using an External Subscriber object, you must edit the `agentinfo.properties` file:

1 On the server using the External Subscriber object to receive a Policy Package Distribution, locate the `agentinfo.properties` file, which is in the `\zenworks\pds\ted` directory.

2 Open the `agentinfo.properties` file in a text editor.

3 Add the following lines in the file:

   TRUSTED_TREE=*source_tree_name*
   TRUSTED_TDN=*External_Subscriber_DN*

   where source_tree_name is the tree where the Distributor object resides that is sending the Policy Package Distribution, and *External_Subscriber_DN* is the fully-distinguished name of the External Subscriber object receiving the Distribution.

4 Save the `agentinfo.properties` file and exit the text editor.

5 When ready, send the Policy Package Distribution to the External Subscriber.

Scheduling

The External Subscriber object is not scheduled.

3.8.2 Using External Subscribers for Out-of-Tree Distributions

Review the following sections to understand how to use External Subscribers for out-of-tree distributions:

- "External Subscriber, One Distributor, and One Tree" on page 164
- "External Subscriber, Multiple Distributors, and Multiple Trees" on page 165

External Subscriber, One Distributor, and One Tree

After you install Policy and Distribution Services software to your servers, you can send Distributions to a server that does not have a Subscriber object in any tree using the External Subscriber object.

The Tiered Electronic Distribution configuration illustrated in Figure 3-32 might exist for the Distributor's routing of its Distributions through External Subscribers:

Figure 3-32 *Distribution Flow in One Tree*
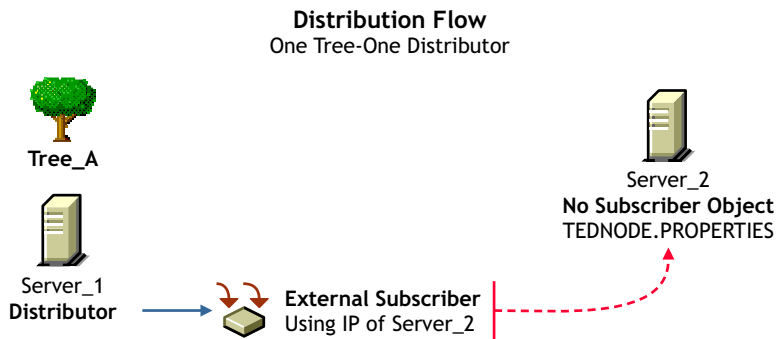
In this example, Server_2 does not have a Subscriber object in any tree. It has only the Subscriber software installed on it so that it can receive and extract Distributions. It can be a NetWare server with an NCP server object in any tree, or a Windows server in a Microsoft domain.

To send a Distribution from Distributor_A to Server_2, create an External Subscriber object in Tree_A and list Server_2's IP address or DNS name in the External Subscriber object's properties.

The eDirectory Distribution View

From an eDirectory perspective, the Distribution is sent from the Distributor object to the External Subscriber object, which in turn sends it to Server_2. You can use a parent Subscriber in Tree_A (not shown) where you do not want the Distributor to be directly sending its Distributions to Server_2.

The Actual Distribution Process

From a topology perspective, the Distribution file is sent from Server_1 to Server_2, using the IP address or DNS name of Server_2 that is located in the External Subscriber object's properties.

Configuring the Subscriber Server

Server_2 receives its Tiered Electronic Distribution configuration information from the `tednode.properties` file installed on its server when the Subscriber software was installed there. Because there is no Subscriber object to configure, you need to edit Server_2's `tednode.properties` file in order to make configuration changes. For information on editing

the `tednode.properties` file, see Section 3.13, "Editing the Tednode.properties File," on page 193.

The Subscriber Server's Trusted Tree

In order for Server_2 to have policies enforced on it, Tree_A needs to be established as its trusted tree during installation of the Subscriber software to the server. For the installation steps, see "Installation on NetWare and Windows Servers" in the *Novell ZENworks 7 Server Management Installation Guide*.

External Subscriber, Multiple Distributors, and Multiple Trees

After you install Policy and Distribution Services software to your servers in multiple trees, you can send Distributions between trees using the External Subscriber object.

The Tiered Electronic Distribution configuration illustrated in Figure 3-33 might exist for the Distributor's routing of its Distributions through External Subscribers:

Figure 3-33   *Distribution Flow in Two Trees*
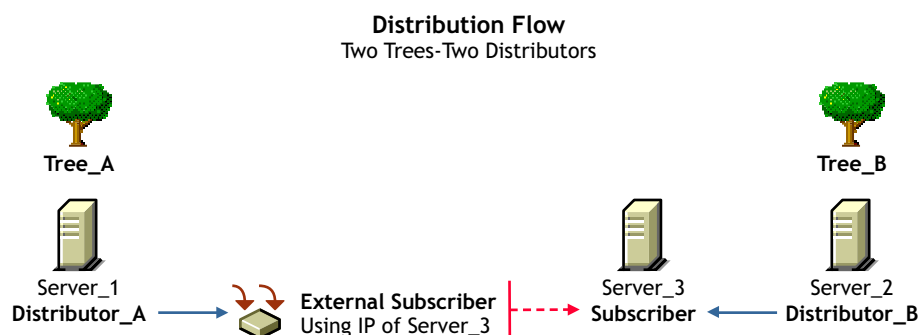
In this example, Server_3 has a Subscriber object in Tree_B.

To send a Distribution from Distributor_A to Server_3, you create an External Subscriber object in Tree_A and list Server_3's IP address or DNS name in the External Subscriber object's properties.

- "The eDirectory Distribution View" on page 165
- "The Actual Distribution Process" on page 165
- "Subscriber Server_3's Trusted Tree and Its Tiered Electronic Distribution Configuration" on page 166

The eDirectory Distribution View

From an eDirectory perspective, the Distribution is sent from Distributor_A to the External Subscriber object, which in turn sends it to Server_3. You can use a parent Subscriber in Tree_A (not shown) where you do not want Distributor_A to be directly sending its Distributions to Server_3.

The Actual Distribution Process

From a topology perspective, the Distribution file is sent from Server_1 to Server_3, using the IP address or DNS name of Server_3 that is located in the External Subscriber object's properties.

Subscriber Server_3's Trusted Tree and Its Tiered Electronic Distribution Configuration

Each tree has a Distributor that provides configuration information for the Subscriber servers in its own tree.

Server_3 receives its Tiered Electronic Distribution configuration information from Distributor_B, because Tree_B was set as Server_3's trusted tree when it was made a Subscriber using the installation program. However, Server_3 cannot extract a Distribution from Distributor_A until it has been configured by Distributor_B, which is done the first time the Subscriber receives a Distribution from Distributor_B.

3.8.3 Creating and Configuring External Subscribers

You can create External Subscriber objects for sending Distributions to Subscriber servers with Subscriber objects residing on other trees or to Subscriber servers that do not have a Subscriber object in any tree.

The following sections provide steps to create and configure an External Subscriber:

- "Creating an External Subscriber Object" on page 166
- "Configuring the External Subscriber Object" on page 166

Creating an External Subscriber Object

1 In ConsoleOne, select the container to hold the External Subscriber object, click *File > New > Object*, then select *TED External Subscriber*.

2 Provide a name for the External Subscriber object.

Make the name unique to help identify the server from the other tree.

3 Provide the server's TCP/IP address or DNS name, then click *OK*.

This must be a valid TCP/IP address or fully distinguished DNS name.

4 Continue with "Configuring the External Subscriber Object" on page 166.

Configuring the External Subscriber Object

1 In ConsoleOne, right-click an External Subscriber object, then click *Properties*.

2 Click *General > Settings* and fill in the Setting fields:

Use policy: Select this check box if you want to use the values set in the Tiered Electronic Distribution policy that is being enforced on the External Subscriber's server.

If you select this option, the Parent Subscriber field is dimmed and the policy settings are used instead.

Parent Subscriber: Specifies a parent Subscriber from which all Distributions are received.

Because the routing hierarchy in a Distributor object's properties only accounts for parent Subscribers, this field is where you can connect an end-node Subscriber to the routing hierarchy. These end-node Subscribers (which in this case are External Subscribers) cannot be used to pass Distributions to other Subscribers.

3 Select the *Network Address* tab and verify the IP address of the External Subscriber's server.

IP address: You provided this IP address when you created the object. Verify that it is correct.

4 Select the *Channels* tab and fill in the fields:

- ◆ **Channels this Subscriber is subscribed to**

  Lists the Channels the External Subscriber is subscribed to.

  Active: To activate a Channel for this External Subscriber so it can receive the Channel's Distributions, select a Channel, then select the check box to enable it. To deactivate a Channel so that the External Subscriber does not receive the Channel's Distributions, deselect the check box to disable it.

  Channel: Click Add to create a Channel. Click Details to edit a Channel.

- ◆ **Channels subscribed to through Subscriber Group memberships**

  Lists the Subscriber Groups that the External Subscriber is a member of, paired with which Channels the External Subscriber is subscribed to by virtue of membership in a Subscriber Group.

  These columns are for display only. The Details, Add, and Delete buttons do not apply.

  Active: Indicates whether the Channel subscribed to is active.

  Channel: Displays the Channel subscribed to through membership in a group.

  Subscriber Groups: Displays the groups the External Subscriber is a member of. You can sort the listing by clicking the column heading.

5 To include this External Subscriber in a group, click *Group Membership*, click *Add*, browse for a Subscriber Group object, click *Select*, then click *OK*.

6 When you are finished configuring the External Subscriber object, click *OK* to exit the object's properties.

3.9 Configuring Multiple Tiered Electronic Distribution Objects

When you have the same configuration change to make to several Tiered Electronic Distribution objects, you can save time by modifying the properties of multiple objects.

You can perform multiple object modifications for the following Tiered Electronic Distribution objects:

Distributor	Subscriber Group	Distribution	Policy Package
Subscriber	External Subscriber	Channel	

For more information, see:

- ◆ Section 3.9.1, "Issues with Modifying Multiple Tiered Electronic Distribution Object Properties," on page 168
- ◆ Section 3.9.2, "Modifying Multiple Tiered Electronic Distribution Object Properties," on page 169
- ◆ Section 3.9.3, "Property Tabs Available for Multiple-Object Modifications," on page 170

3.9.1 Issues with Modifying Multiple Tiered Electronic Distribution Object Properties

- **Available properties:** Although the purpose is to provide a means to make the same changes to multiple objects, not all properties for the Tiered Electronic Distribution objects can be modified using this method.

  The Schedule and Other property tabs are not available for editing the properties of multiple-selected Tiered Electronic Distribution objects. For the Distribution object, the Type tab is also not available. For changes to these property tabs, you must edit each Tiered Electronic Distribution object individually.

- **Modified fields:** The fields where you make changes in the Properties of Multiple Objects dialog box are the only modifications that are made for the selected objects. In other words, if you leave a field blank (you do not modify it), no change is made in that field for all of the selected objects. Each object retains its original field entry.

  Where objects have different information in a given field, that field is blank in the Properties of Multiple Objects dialog box.

- **Removing information:** In some fields, a space is a valid entry. You can use this as a method for removing varied existing entries for each of the selected Tiered Electronic Distribution objects when you want the field to be blank for all of the selected objects.

- **Policy defaults:** If you have a Tiered Electronic Distribution policy in force, the Use Policy check box is displayed in each Tiered Electronic Distribution object's properties, but only selected for the individual Tiered Electronic Distribution objects where the policy applies (because their properties have never been edited, or you selected that check box).

  For multiple object properties, if the Use Policy check box is displayed and selected, the policy's contents are displayed in dimmed text in the applicable fields. These attributes are only applicable to those Tiered Electronic Distribution objects whose individual properties contain a selected Use Policy check box.

  You can deselect the Use Policy check box when editing multiple properties to disable the Tiered Electronic Distribution policy for the selected Tiered Electronic Distribution objects that were previously using the policy. Any changes you make are replicated to all selected Tiered Electronic Distribution objects and the Tiered Electronic Distribution policy are no longer in force for any of those objects.

  IMPORTANT: If the Working Directory field for an object received its location from the Tiered Electronic Distribution policy, and you deselect the Use Policy check box when editing multiple properties, the Working Directory field is then left blank for that object. Therefore, the next time you access the properties for that object, you will be required to provide a working directory location.

3.9.2 Modifying Multiple Tiered Electronic Distribution Object Properties

To modify the properties of multiple Tiered Electronic Distribution objects:

1 In ConsoleOne, select a number of Tiered Electronic Distribution objects.

They must be of the same type, such as all Distributor objects. The Properties of Multiple Objects menu option do not display if you select multiple objects of different types.

You can select multiple objects using the Shift and Ctrl keys.

2 Right-click the selected objects and click *Properties of Multiple Objects*.

Each of the selected objects is listed in the Objects to Modify tab on the Properties of Multiple Objects dialog box. These are the objects that have their properties modified when you make changes.

3 To change the objects displayed in the list, click *Add* or *Remove*.

The Add button allows you to browse for other Tiered Electronic Distribution objects. Only objects of the type you have previously selected are displayed for adding to the list.

Before selecting the Remove button, you must first select one or more objects in the list. This only removes the objects from the list, not from eDirectory.

4 Select a tab containing the property that you want to modify.

For descriptions of the property tabs available for the various Tiered Electronic Distribution objects, see Section 3.9.3, "Property Tabs Available for Multiple-Object Modifications," on page 170.

5 Edit the property.

The changes are made to all of the objects listed in the *Objects to Modify* tab.

For more information on individual property fields, see the descriptions within the steps in the following sections:

- "Configuring Distributors" on page 107
- "Creating a Distribution" on page 124
- "Creating and Configuring Channels" on page 147
- "Configuring Subscribers" on page 151
- "Creating and Configuring Subscriber Groups" on page 158
- "Creating and Configuring External Subscribers" on page 166

6 Repeat Step 4 and Step 5 until you have finished modifying the various properties for the selected objects.

7 When finished modifying properties, click *OK* to close the Properties of Multiple Objects dialog box.

All changes that you have made are updated for all of the selected objects.

3.9.3 Property Tabs Available for Multiple-Object Modifications

The tables in the following sections list the property tabs that are available in the multiple object editing mode for each Tiered Electronic Distribution object.

IMPORTANT: Generally, if you change information, it is changed for all of the selected objects. Exceptions are noted in the explanations.

- "Distributor Object" on page 170
- "Distribution Object" on page 171
- "Channel Object" on page 171
- "Subscriber Object" on page 172
- "External Subscriber Object" on page 173
- "Subscriber Group Object" on page 173
- "Policy Package Object" on page 174

Distributor Object

Table 3-9 *Distributor Object Property Tabs*

Property Tabs Available	Explanation
Objects to Modify	You can add or remove Distributor objects from the list of objects to be modified. This does not add or remove the objects from eDirectory.
General	This includes the Settings and Messaging subtabs. For the Settings subtab, none of the fields display information, even if it is identical between the selected Subscriber objects. However, dimmed text is displayed in fields where the Tiered Electronic Distribution policy is in effect for one or more of the selected Tiered Electronic Distribution objects. In the Settings subtab, you can only add new information that applies to all of the selected Subscriber objects. In the Messaging subtab, you can edit existing entries.
Routing	If there are any differences in routing hierarchies between the selected Distributor objects, nothing is displayed for this tab. You can only edit routing hierarchies for multiple Distributor objects when they are identical.
NDS Rights	This tab includes the New Trustees and the Inherited Filter Rights subtabs.

Distribution Object

Table 3-10 *Distribution Object Property Tabs*

Property Tabs Available	Explanation
Objects to Modify	You can add or remove Distribution objects from the list of objects to be modified. This does not add or remove the objects from eDirectory.
General	This includes the Settings and Restrictions subtabs. For the Settings subtab, none of the fields display information, even if it is identical between the selected Subscriber objects. However, dimmed text is displayed in fields where the Tiered Electronic Distribution policy is in effect for one or more of the selected Tiered Electronic Distribution objects. In the Settings subtab, you can only add new information that applies to all of the selected Subscriber objects. In the Restrictions subtab, you can edit existing entries.
Channels	Channels do not automatically display on this tab. You can only browse for Channels to add to each of the selected Distribution objects, or browse for a Channel to be removed from each of the selected Distribution objects that are associated with that Channel. Adding or removing a Channel in the list on this tab does not add or remove the Channel object from eDirectory.
NDS Rights	This tab includes the New Trustees and the Inherited Filter Rights subtabs.

Channel Object

Table 3-11 *Channel Object Property Tabs*

Property Tabs Available	Explanation
Objects to Modify	You can add or remove Channel objects from the list of objects to be modified. This does not add or remove the objects from eDirectory.
General	This includes the Settings subtab (with the Active check box and the Description field). For the Settings subtab, none of the fields display information, even if it is identical between the selected Subscriber objects. However, dimmed text is displayed in fields where the Tiered Electronic Distribution policy is in effect for one or more of the selected Tiered Electronic Distribution objects. In the Settings subtab, you can only add new information that applies to all of the selected Subscriber objects.
Distributions	Distributions do not automatically display on this tab. You can only browse for Distributions to add to each of the selected Channel objects, or browse for a Distribution to be removed from each of the selected Channel objects that are associated with that Distribution. Adding or removing a Distribution in the list on this tab does not add or remove the Distribution object from eDirectory.

Property Tabs Available	Explanation
Subscribers	Subscribers do not automatically display on this tab. You can only browse for Subscribers to add to each of the selected Channel objects, or browse for a Subscriber to be removed from each of the selected Channel objects that are associated with that Subscriber.

Adding or removing a Subscriber in the list on this tab does not add or remove the Subscriber object from eDirectory. |
| NDS Rights | This tab includes the New Trustees and the Inherited Filter Rights subtabs. |

Subscriber Object

Table 3-12 *Subscriber Object Property Tabs*

Property Tabs Available	Explanation
Objects to Modify	You can add or remove Subscriber objects from the list of objects to be modified. This does not add or remove the objects from eDirectory.
General	This includes the Settings, Messaging, and Working Context subtabs.

For the Settings subtab, none of the fields display information, even if it is identical between the selected Subscriber objects. However, dimmed text is displayed in fields where the Tiered Electronic Distribution policy is in effect for one or more of the selected Tiered Electronic Distribution objects.

In the Settings subtab, you can only add new information that applies to all of the selected Subscriber objects. In the Messaging subtab, you can edit existing entries. |
| Channels | Channels do not automatically display on this tab. You can only browse for Channels to add to each of the selected Subscriber objects, or browse for a Channel to be removed from each of the selected Subscriber objects that are associated with that Channel.

Adding or removing a Channel in the list on this tab does not add or remove the Channel object from eDirectory. |
| Variables | You can only add a new variable for all of the selected objects. Variables that are common among all of the selected objects are not displayed for editing. You must visit each Subscriber object individually to modify existing variables. |
| Group Membership | Group Memberships do not automatically display on this tab. You can only browse for Group Memberships to add to each of the selected Subscriber objects, or browse for a Group Membership to be removed from each of the selected Subscriber objects that are associated with that Group Membership.

Adding or removing a Group Membership in the list on this tab does not add or remove the Group Membership object from eDirectory. |
| NDS Rights | This tab includes the New Trustees and the Inherited Filter Rights subtabs. |

External Subscriber Object

Table 3-13 *External Subscriber Object Property Tabs*

Property Tabs Available	Explanation
Objects to Modify	You can add or remove External Subscriber objects from the list of objects to be modified. This does not add or remove the objects from eDirectory.
General	This includes the Settings subtab. For the Settings subtab, only the Parent Subscriber field exists. If you make an entry here, all selected External Subscribers will have the same parent Subscriber.
Channels	Channels do not automatically display on this tab. You can only browse for Channels to add to each of the selected External Subscriber objects, or browse for a Channel to be removed from each of the selected External Subscriber objects that are associated with that Channel. Adding or removing a Channel in the list on this tab does not add or remove the Channel object from eDirectory.
Group Membership	Group Memberships do not automatically display on this tab. You can only browse for Group Memberships to add to each of the selected Subscriber objects, or browse for a Group Membership to be removed from each of the selected Subscriber objects that are associated with that Group Membership. Adding or removing a Group Membership in the list on this tab does not add or remove the Group Membership object from eDirectory.
NDS Rights	This tab includes the New Trustees and the Inherited Filter Rights subtabs.

Subscriber Group Object

Table 3-14 *Subscriber Group Object Property Tabs*

Property Tabs Available	Explanation
Objects to Modify	You can add or remove Subscriber Group objects from the list of objects to be modified. This does not add or remove the objects from eDirectory.
General	This includes the Settings and Messaging subtabs.
Channels	Channels do not automatically display on this tab. You can only browse for Channels to add to each of the selected Subscriber Group objects, or browse for a Channel to be removed from each of the selected Subscriber Group objects that are associated with that Channel. Adding or removing a Channel in the list on this tab does not add or remove the Channel object from eDirectory.

Property Tabs Available	Explanation
Group Members	Group Members do not automatically display on this tab. You can only browse for Group Members to add to each of the selected Subscriber objects, or browse for Group Members to be removed from each of the selected Subscriber objects that are associated with that Group Membership. Adding or removing a Group Membership in the list on this tab does not add or remove the Group Membership object from eDirectory.
NDS Rights	This tab includes the New Trustees and the Inherited Filter Rights subtabs.

Policy Package Object

Table 3-15 *Policy Package Object Property Tabs*

Property Tabs Available	Explanation
Objects to Modify	You can add or remove Policy Package objects from the list of objects to be modified. This does not add or remove the objects from eDirectory.
Policies	This includes the various supported platform subtabs. For more information on the policies available on these platforms, see Section 4.1.6, "Server Policy Descriptions," on page 204.
Distributions	Distributions do not automatically display on this tab. You can only browse for Distributions to add to each of the selected Policy Package objects, or browse for a Distribution to be removed from each of the selected Policy Package objects that are associated with that Distribution. Adding or removing a Distribution in the list on this tab does not add or remove the Distribution object from eDirectory.
NDS Rights	This tab includes the New Trustees and the Inherited Filter Rights subtabs.

3.10 Sending Distributions

For information on sending Distributions, see the following:

3.10.1 Understanding the Distribution Processes

Following are the processes for creating and sending a Distribution, generally done in this order:

1. **Configure and schedule the Distributors.** You must use the installation program on the *ZENworks 7 Server Management with Support Pack 1 Program* CD to create a Distributor.

   For information on Distributors, see Section 3.3, "Distributors," on page 95 and "Distributor Object's Refresh Schedule" on page 329.

2. **Configure and schedule the Subscribers.** You must use the installation program on the *ZENworks 7 Server Management with Support Pack 1 Program* CD to create a Subscriber.

   One of the primary configurations that you must do for Subscribers is to associate them with the Channels that hold the Distributions they need. For more information, see Section 3.6.5, "Associating Subscribers with Channels," on page 155.

   For information on Subscribers, see Section 3.6, "Subscribers," on page 149 and "Subscriber Object's Extract Schedule" on page 332.

3. **Configure the necessary policies.** Policy Packages that contain the desired policies must be created in ConsoleOne or iManager before they are distributed.

   For information on policies, see Section 4.3, "Configuring Server Policies," on page 209.

4. **Create, configure, and schedule the Distributions.** You can use either ConsoleOne or iManager to create Distribution objects.

   This could be the most time-consuming portion of the whole process, depending on the complexity of the Distribution to be configured. After you set up your Distributors and Subscribers and create the Distribution objects, you only need to utilize the Distributors' routing hierarchies for distributing the files and policies to your Subscriber servers.

   The Distribution object's schedule is the best place to prevent an individual Distribution from being sent.

   For information on Distributions, see Section 3.4, "Distributions," on page 111 and "Distribution Object's Build Schedule" on page 330.

5. **Create, configure, and schedule the Channels.** You can use either ConsoleOne or iManager to create Channel objects.

   Usually, you create a new Channel for each Distribution. It is generally easier to manage your distribution system by matching Channels with what they distribute. However, you can include multiple Distributions in a Channel, such as when they are related and all Subscribers subscribing to the Channel need all of those Distributions. For example, a Channel could hold several Distributions that each contain a different virus pattern update.

   The Channel object is normally the best object to use for controlling whether Distributions should be sent. Setting its schedule to Never effectively stops the distribution process for all of the Distributions listed in it.

   For information on Channels, see Section 3.5, "Channels," on page 146 and "Channel Object's Send Schedule" on page 331.

The Distributions are built, sent, and extracted according to the schedules that you set for each of the Tiered Electronic Distribution objects involved.

For information on the distribution processes, see Section 3.2.2, "The Basic Distribution Process," on page 88.

You might have accomplished some of the above processes during installation of Server Management and during your initial system configuration (see Chapter 1, "Post-Installation Setup," on page 29).

3.10.2 Forcing a Single Distribution To Be Sent

If you want to send a single Distribution outside of the normal Refresh, Build, and Send schedules, and the Channel's Send schedule is not ready to fire, you can manually force this distribution process using only the ZENworks Server Management role in iManager.

To force a single Distribution to be sent, do one of the following:

- If the Send Distribution Immediately After Building option is selected in the Distribution's properties, go to iManager, click Distribution, then click Build Distribution.

  Even if there are other Distributions in the Channel where this Distribution is listed, only this Distribution is sent.

- If the Send Distribution Immediately After Building option is not selected in the Distribution's properties, go to iManager, click Distribution, click Build Distribution, click Channel, then click Distribute Channel.

  All other Distributions in the Channel are also be sent if needed by the Subscribers.

As soon as a Subscriber receives an entire Distribution, it extracts it according to the Subscriber's Extract schedule.

3.10.3 Sending Distributions Through Parent Subscribers

Subscribers can receive and extract Distributions, and they can also pass on Distributions to other Subscribers. Subscribers that pass on Distributions are known as parent Subscribers.

Parent Subscribers do not need to be subscribed to the Distributions they are passing on. They simply receive a Distribution for passing it on to a subordinate Subscriber that has done two things:

- Subscribed to the Channel listing the Distribution
- Identified the parent Subscriber in the subordinate Subscriber's object properties

To set up parent Subscribers for passing on Distributions:

1 Determine a Subscriber object (hereafter referred to as "child Subscriber") that cannot receive a certain Distribution because this child Subscriber is not contained in the Distributor's routing hierarchy (the Distributor owning this Distribution).

2 In that Subscriber object's properties, click *General > Settings*, in the *Parent Subscriber* field browse for and select a Subscriber object that is contained in the Distributor's routing hierarchy, then click *OK*.

  This establishes the Subscriber selected as a parent Subscriber. This distinction is not kept in the parent Subscriber's object properties, but only in the child Subscriber's.

3 Create a *Channel* object where only the child Subscriber is associated.

4 Create a Distribution, then associate it with the child Subscriber's Channel.

5 Send this Distribution.

Because this Distribution is associated only with the Channel where the child Subscriber is subscribed, the parent Subscriber does not extract it, but only passes it on to the child Subscriber.

Because the parent Subscriber is in the routing hierarchy of the Distributor, it has access to the Distribution for passing it on. However, the child Subscriber does not have any access to the Distributor, so it needs the parent Subscriber to provide access to the Distribution.

Although you can establish a parent Subscriber for a child Subscriber, the child Subscriber can still be subscribed to a Channel where the parent Subscriber is subscribed. Both Subscribers can receive and extract that Channel's Distributions without the parent Subscriber passing it on to the child Subscriber, because the child can have access to that particular Distributor's routing hierarchy. The key is whether the Distributor owning the desired Distribution can send it to the child Subscriber without using a parent Subscriber.

3.10.4 Sending Distributions between Trees

Using External Subscribers, you can send Distributions from one tree to another. To accomplish this, do the following:

1 Make sure Tiered Electronic Distribution is installed to both trees.

In the remaining steps, TREE1 represents the tree where the Distribution is created and TREE2 represents the other tree where you want the Distribution sent.

The server in TREE2 that is to receive the Distribution from TREE1 must have the Subscriber software installed on it (meaning it is a Subscriber in TREE2).

For information on installing Tiered Electronic Distribution, see "Installation on NetWare and Windows Servers" in the *Novell ZENworks 7 Server Management Installation Guide*.

2 In TREE1, create an External Subscriber object.

Make sure that the IP address or DNS name you provide for this object matches the Subscriber server in TREE2 where you want the Distribution to be sent.

For steps in creating External Subscribers, see Section 3.8.3, "Creating and Configuring External Subscribers," on page 166.

3 In TREE1, create the Channel for the Distribution.

For steps in creating Channels, see Section 3.5.2, "Creating and Configuring Channels," on page 147.

4 Associate the External Subscriber object you created in Step 2 with the Channel you created in step Step 3.

Other Subscribers from TREE1 can already be associated with this Channel.

For steps in associating Subscribers with Channels, see Section 3.6.5, "Associating Subscribers with Channels," on page 155.

5 In TREE1, create the Distribution.

For steps in creating Distributions, see Section 3.4, "Distributions," on page 111.

6 Associate this Distribution with the Channel you created in Step 3.

7 Verify that the External Subscriber server in TREE2 received the Distribution.

3.10.5 Sending Distributions: Firewall and Cluster Issues

To send Distributions across a firewall, you must enable both the primary and secondary IP addresses of the servers running the Site List server or Distributor server software. If you only allow the secondary IP address to pass through the firewall, the Distribution cannot be sent because Tiered Electronic Distribution uses the primary IP addresses of its recipient servers.

If you are running ZENworks in a cluster, you also need to allow access to all primary IP addresses of all nodes involved.

3.11 Miscellaneous Tiered Electronic Distribution Issues

- Section 3.11.1, "Directory Sync Granularity for File Distributions," on page 178
- Section 3.11.2, "Understanding Dependencies in Tiered Electronic Distribution," on page 184
- Section 3.11.3, "System Resources and Server Behavior," on page 184
- Section 3.11.4, "Controlling I/O Rates and Concurrent Distributions," on page 185
- Section 3.11.5, "Minimizing Messaging Traffic," on page 186
- Section 3.11.6, "Changing DNS Names or IP Addresses for Tiered Electronic Distribution Servers," on page 188
- Section 3.11.7, "When a Tiered Electronic Distribution Process Fails," on page 188

3.11.1 Directory Sync Granularity for File Distributions

The File Distribution has been enhanced with directory sync granularity:

- "Understanding Synchronization and Directory Sync Granularity" on page 178
- "How the Synchronization and Directory Sync Granularity Processes Work" on page 179
- "Synchronizing Directories for a File Distribution" on page 182

Understanding Synchronization and Directory Sync Granularity

A File Distribution, with or without synchronization enabled, adds or updates files and directories on a Subscriber server. However, with synchronization enabled it also causes the deletion of files and directories. Therefore, file and directory deletion on the Subscriber server is the main function of synchronization.

Directory sync granularity allows you to specify synchronization at any directory level in the Distribution to provide synchronization "from here down."

How the Synchronization and Directory Sync Granularity Processes Work

Table 3-16 illustrates what a synchronized File Distribution does to the Subscriber server's file system if synchronization is enabled in the Distribution:

Table 3-16 *Directory Sync Granularity Comparison*

Files and directories located on the Distributor server that are contained in the File Distribution:	Applicable directories on the Subscriber server before the Distribution is received and extracted:

`data:\zenworks\viruspatterns`
`data:\zenworks\nw65sp\nw65sp1.exe`
`data:\zenworks\nw65sp\nw65sp2`
	`data:\zenworks\viruspatterns`
	`data:\zenworks\nw65sp`

Each of the end items in the above paths are synchronized in this Distribution.	One of the files is missing from the `\viruspatterns` directory on the Subscriber, and it is also missing the `\nw65sp2` directory.

Upon extraction of the File Distribution, the following occurs on the Subscriber server's file system:

1. The missing virus pattern file is restored in the `\viruspatterns` directory.

   The `\viruspatterns` directory is also made to exactly match the files and subdirectories contained in the Distribution by deleting any files or subdirectories on the Subscriber that are not contained in the Distribution.

2. The `nw65sp1.exe` file is updated in the `\nw65sp` directory. Nothing else is synchronized in that directory, because synchronization was not enabled for the `\nw65sp` directory itself.

3. The `\nw65sp2` directory and its files and subdirectories are restored from the Distribution under the `\nw65sp` directory on the Subscriber.

Directory sync granularity also allows you to retain unsynchronized directories while synchronizing their peer directories. For example, you could select to synchronize the `data:\zenworks\viruspatterns` and `data:\zenworks\nw65sp\nw65sp2` directories, but not the `data:\zenworks\nw65sp`directory.

However, if you synchronize the parent `data:\zenworks`directory, all of its subdirectories must also be synchronized, because synchronization occurs from the specified directory and downward. Therefore, when you select directories to be synchronized, you cannot select a parent directory to be synchronized, then select some of its child directories to not be synchronized.

All child directories are automatically synchronized when a parent directory is set to be synchronized, and a parent directory automatically loses its synchronization when one of its child directories has synchronization turned off for it.

For example, Table 3-17 illustrates this:

Table 3-17 *Directory Sync Granularity Plan Comparisons*

Incorrect Plan	Correct Plan

Synchronize:
`data:\zenworks`
	Synchronize:
	`data:\zenworks\viruspatterns`
	`data:\zenworks\nw65sp\nw65sp1.exe`
	`data:\zenworks\nw65sp\nw65sp2`

Do not synchronize:
`data:\zenworks\nw65sp`
	Do not synchronize:
	`data:\zenworks`
	`data:\zenworks\nw65sp`

This does not work, because by synchronizing the `\zenworks` directory, the `\nw65sp` directory must also be synchronized.	This works, because the directories desired to not be synchronized are higher in the path than those that are desired to be synchronized.

The next few sections describe synchronization scenarios:

- "Synchronizing All Directories Under the Distribution's Target Directory" on page 180
- "Using Directory Sync Granularity to Synchronize Directories at Various Levels" on page 181
- "Synchronizing a Subscriber Server Directory with Certain Distributor Server Files" on page 181

Synchronizing All Directories Under the Distribution's Target Directory

For the source, choose the directories on the Distributor server's file system to be synchronized. For the destination, the directories to be synchronized might or might not already exist in the Subscriber server's file system.

If the files already exist, when the Distribution is sent, their content is made to match that of the corresponding directory on the Distributor server's file system. If they do not exist, they are added on the Subscriber server's file system.

Determine where these directories should exist on the Subscriber server's file system. In other words, there is a parent directory under which the synchronized directories are located, or there are the synchronized directories located at the root of the Subscriber's file system. Variables can be used to specify the target on the Subscriber server's file system where the Distribution is to be extracted.

In Figure 3-34, the `\viruspatterns` and `\supportpacks` directories on the Distributor server are created and synchronized under the `vol1:\servers` directory on the Subscriber server.

Figure 3-34 *Synchronizing All Directories*
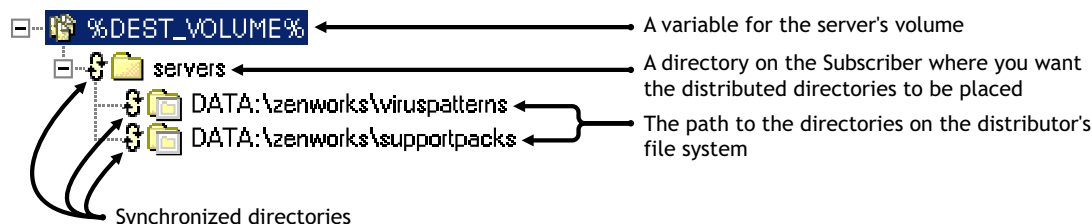
Using Directory Sync Granularity to Synchronize Directories at Various Levels

You can synchronize at the target level (%DEST_VOLUME%). In the example above, we have defined it as `vol1:\servers`. In that case, the only subdirectories that will exist under that directory are viruspatterns and supportpacks. All other existing subdirectories are deleted.

To retain other directories under `vol1:\servers`, you would not enable synchronization at the target level (%DEST_VOLUME%). Instead, you would drop down to the subordinate directories and synchronize those. For example, Figure 3-35 illustrates synchronizing only the `\viruspatterns` directory. That means there could be other directories under `vol1:\servers` that would be unsynchronized, such as `\supportpacks`.

Figure 3-35 *Using Directory Sync Granularity*
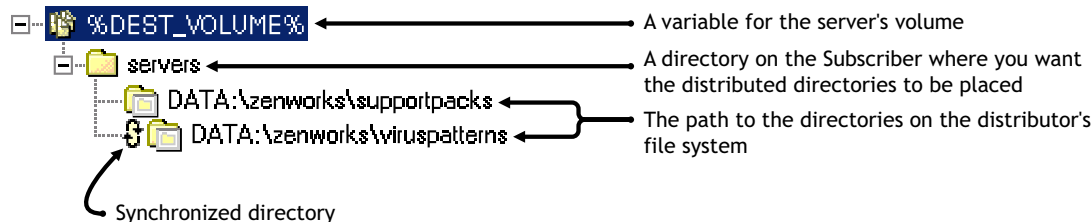
Synchronizing a Subscriber Server Directory with Certain Distributor Server Files

If the File Distribution has certain files on the Distributor server selected to be associated with a directory in the Distribution, you would not normally synchronize that directory in the Distribution. If you did, you could lose valuable data in that directory on the Subscriber server.

For example, Figure 3-36 illustrates this:

Figure 3-36 *Synchronizing a Subscriber Server Directory*
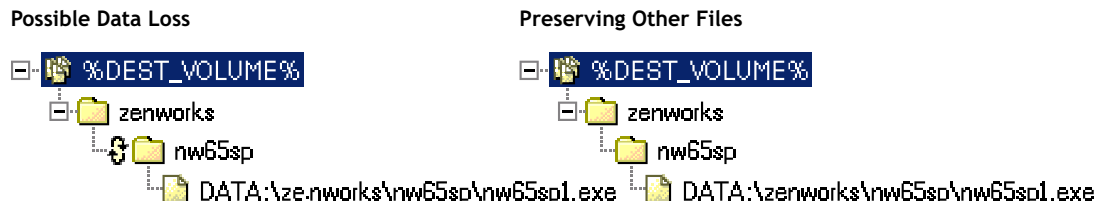
The Possible Data Loss method would cause all files in the \nw65sp directory and any of its subdirectories to be deleted from the Subscriber server, except for the nw65sp1.exe file. The Preserving Other Files method just updates the nw65sp1.exe file in the \nw65sp directory, leaving all other files and subdirectories unchanged on the Subscriber server.

Synchronizing Directories for a File Distribution

1 In ConsoleOne®, right-click a Distribution object, then click *Properties*.

2 Select the *Type* tab, then select *File* for the type of Distribution.
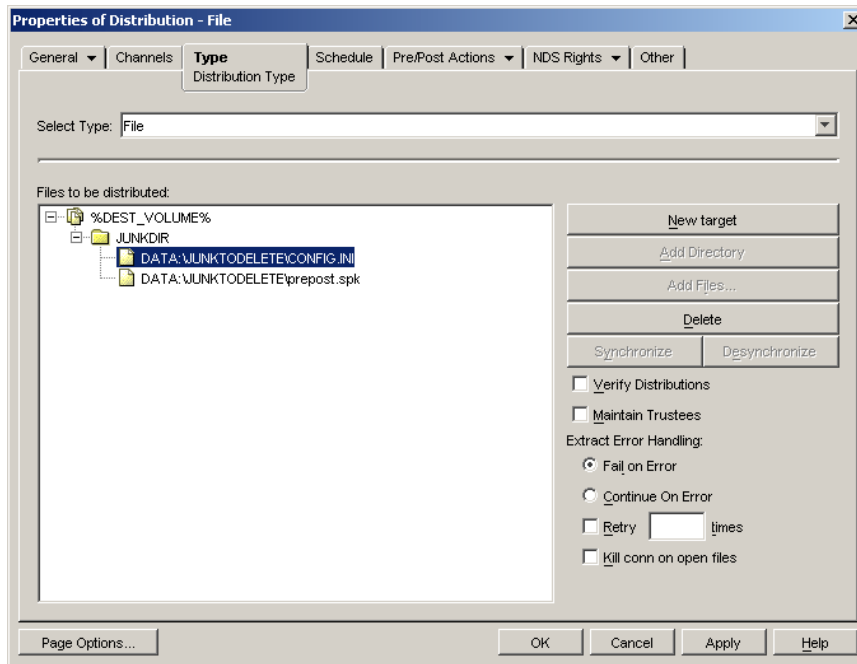
3 Click *New Target* and %DEST_VOLUME% as the default variable that contains the target path on each Subscriber server.

WARNING: If you want to synchronize at the target level, make sure this variable does not contain the root of the Subscriber server's file system.

4 Select the *Add Directory* and *Add Files* buttons to create the directory structure in the *Files To Be Distributed* box.

The *Add Directory* button is for the Subscriber server's target paths and directories, and the *Add Files* button is for browsing the distributor's file system and selecting directories and files that are to be included in the Distribution.

5 If necessary, click the plus signs to expand the directory structure.

By default, no directories in the listing are selected for synchronization.
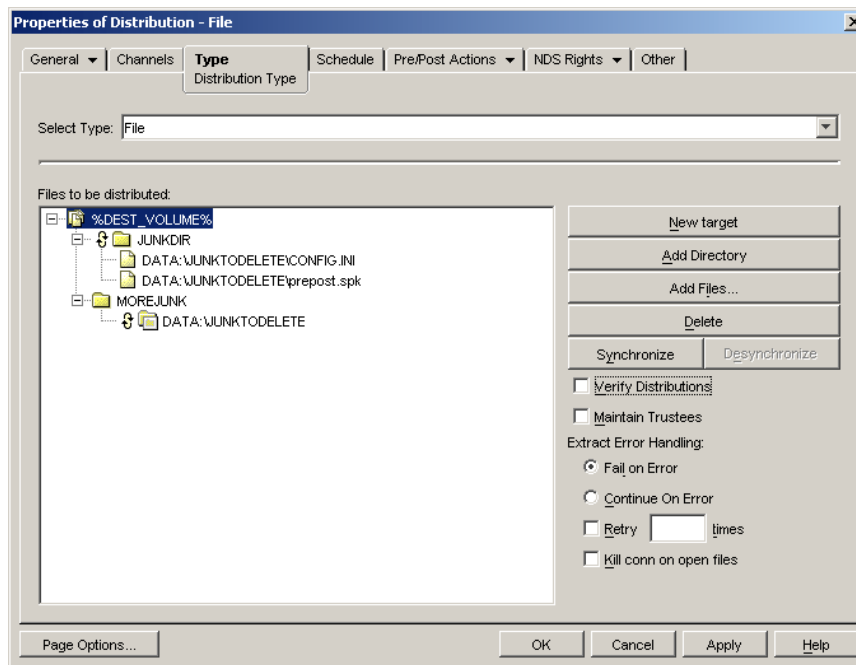
6 To specify a directory to synchronize, select the directory in the *Files To Be Distributed* box, then click the *Synchronize* button.

The *Synchronize* icon (🕀) is placed in front of that directory name.

The *Synchronize* and *Desynchronize* buttons are dimmed when you select a filename instead of a directory name. You can only synchronize directories.

Because directory synchronization is done for the directory you select and all of its subdirectories (in other words, "from here down"), there is no need to synchronize any directories below a directory that you select for synchronization.

When you synchronize a directory and expand the directories below it, the *Synchronize* icon is displayed before each directory's name.

7 To reverse your selection of a directory to be synchronized, select the directory name that has a *Synchronize* icon, then click the *Desynchronize* button.

If you desynchronize a directory and its parent directory was synchronized, the parent directory is also automatically desynchronized. This includes all directories (grandparents) up the path that might have been synchronized. In other words, you cannot synchronize a directory, then desynchronize any directory below it without also causing the directory to be desynchronized.

8 Repeat Step 6 for each directory to synchronize.

9 Continue with configuring the Distribution.

For more information on configuring Distributions, see "Creating and Configuring the Distribution" on page 57.

10 Click *OK* when finished configuring the Distribution.

For this Distribution, directory synchronization occurs with only the directories where you placed the *Synchronize* icon (🕀), including all of their subdirectories.

3.11.2 Understanding Dependencies in Tiered Electronic Distribution

Policy and Distribution Services agents (Policy/Package Agent and Distributor Agent) are dependent on one another and upon eDirectory. It is important to understand the following dependencies when using Policy and Distribution Services to manage your network:

Synchronization of Tiered Electronic Distribution Objects in eDirectory

Server Management uses eDirectory as the repository for information needed by the Tiered Electronic Distribution and Server Policies components. Because eDirectory is a distributed database and can have partitions and replicas throughout the network, it takes time to synchronize all of the replicas each time Server Management objects are created or modified.

Unloading Parent Subscribers

You must change the parent Subscriber attribute in the Subscriber object to change the parent Subscriber. Then, the next time a Distribution is sent, the distribution route to the Subscriber reflects the new parent Subscriber.

If a parent Subscriber Java process is unloaded (exited), the subordinates of the parent Subscriber do not renegotiate to another parent Subscriber. The subordinates wait until that parent Subscriber is loaded again and continue to use it. The reason for this is that if the parent Subscriber was the only server between twenty Subscribers and the Distributor (which is located across the WAN), you do not want all of the Subscribers to go across the WAN to get their Distributions if the parent Subscriber is unavailable.

3.11.3 System Resources and Server Behavior

Using Policy and Distribution Services can affect the behavior of your system:

* Tiered Electronic Distribution usage can affect system behavior because of the traffic created in sending Distributions
* Some server policies are designed to control the behavior of servers, such as how a server should be brought down
* Some server policies are designed for NetWare server configuration, such as SET parameters, content of the `autoexec.ncf` file, and so on

Installing and using Tiered Electronic Distribution can affect any of the following:

* CPU utilization
* Disk space resources
* Network traffic
* Other I/O activity

To optimize your installation of Tiered Electronic Distribution, you should consider the following issues when selecting Distributor and Subscriber servers:

- Which servers are the best candidates for the heavy workload of a Distributor?

  Consider CPU speed for building and sending Distributions, and sufficient disk space for storing all of the Distributor's Distributions.

  The server can perform other non-Server Management network functions, be running other Server Management or non-Server Management software, or be solely dedicated to the Distributor function.

- Which servers do you want to manage using server policies?

  Consider installing the Subscriber software to each server that you want to manage with policies, or where you want to distribute software packages. The policy engine is installed with the Subscriber software; also, the Subscriber software is used to extract and install software packages.

- Which servers could best handle the additional workload of being a parent Subscriber? (A parent Subscriber is a Subscriber that acts as a proxy for the Distributor to store and pass Distributions so that the Distributor does not need to send its Distributions to every Subscriber.)

  Consider CPU speed for sending the Distributions, and free disk space for storing the Distributions that the parent Subscriber passes on.

- Does each of your LAN segments have servers that are capable of being a parent Subscriber?

  Consider WAN traffic when deciding where to locate parent Subscribers.

- Do you have other processes using up bandwidth on some LANs and WAN links?

  Consider Distribution priorities and setting sending and receiving rates to minimize the affect Distributions can have on bandwidth for WAN links.

3.11.4 Controlling I/O Rates and Concurrent Distributions

If you need to control bandwidth usage for Distribution traffic, you can set the I/O rates and the maximum number of concurrent Distributions for Distributors and/or Subscribers.

Attributes of both the Distributor and Subscriber objects provide the following controls:

- **Input rate:** For sending and receiving Distributions, you can set the maximum bytes per second. The Distributor Agent sends the Distributions, and the Policy/Package Agent receives and extracts them. This allows you to have some control over the bandwidth used by these agents.The default is the maximum that the connection can handle. However, this does not control the rate at which FTP, HTTP, and RPM Distributions are built by the Distributor.

- **Output rates based on Distribution's priority:** Sets the default output rate to minimize network traffic for Tiered Electronic Distribution objects. This determines the send rate for Distributors and parent Subscribers. The default value is the maximum that the connection can handle. Blank means that bandwidth is taken from third-party applications.

  There are three output priorities where you can specify a rate:

  - **High priority:** These Distributions are sent before any Medium or Low priority Distributions.

  - **Medium priority:** These Distributions are sent after all High priority and before any Low priority Distributions.

- **Low priority:** These Distributions are sent after all High and Medium priority Distributions.

  For more information, see Section 3.4.5, "Prioritizing Distributions," on page 127.

- **Maximum number of concurrent Distributions:** This determines how many simultaneous Distributions the Distributor Agent can send. The default is unlimited (blank field). The Subscriber always receives as many Distributions as it is sent; however, it only concurrently passes on the number that you choose here.

If there is only one Subscriber, the Distributor sends Distributions at the selected rate. If there are two Subscribers, the Distributions are sent at one half the rate. In other words, to determine the slowest distribution rate, divide the Distributor's output rate by the maximum number of concurrent Distributions.

Because Subscribers always receive another concurrent Distribution, the rate applies even though you cannot limit the number of incoming connections.

3.11.5 Minimizing Messaging Traffic

Tiered Electronic Distribution provides message notifications so that administrators and selected end users are kept informed. Notifications are sent in several ways:

- Information written to log files
- Notifications sent via e-mail messages
- SNMP traps used and displayed on both local and remote consoles

The following sections explain message notification usage:

- "Message Notification Levels" on page 186
- "Managing Message Notification Level Log Files" on page 187
- "Sending Notifications Over LANs and WANs" on page 187

Message Notification Levels

There are seven levels of message notification available. Each level adds its own information to the previous level.

Messaging Level	Description
Level 0 - No messages	Messages are not sent.
Level 1 - Errors	Reports unusual or unexpected situations that can cause an operation to fail. They often require user intervention to correct the problem.
Level 2 - Successes and Level 1 messages	Reports completion of a successful operation.
Level 3 - Warnings and Level 2 messages	Reports unusual but not unexpected runtime conditions. These messages usually do not require user intervention, but in some situations an unusual runtime condition does.
Level 4 - Information and Level 3 messages	Informs the user about what has happened or is currently happening. They usually do not require any action from the user.

Messaging Level	Description
`Level 5 - Trace and Level 4 messages`	Reports detailed trace information that is used to troubleshoot unusual or unexpected situations that cause an operation to fail. This information might only be useful with the guidance of Novell Support..
`Level 6 - Developer trace and Level 5 messages`	Used for debugging code. This information is useful only to Novell Support and Development.

Regardless of the destination for a message, resources are directly affected by the level you choose.

For information on setting message notification levels, see:

- **Distributor object:**
- **Subscriber object:**

Managing Message Notification Level Log Files

The level you choose for a log file affects the rate at which the log file grows. Because log files have no maximum size, you can control the size of a log file by choosing to delete entries after *x* number of days. For information on setting message notification levels for log files, see:

- **Distributor object:**
- **Subscriber object:**

Sending Notifications Over LANs and WANs

The greatest impact on network traffic can come from the levels you choose for SNMP traps and for the remote console. For information on setting message levels for SNMP traps, e-mail messages, and the server's console, see:

- **Distributor object:**
- **Subscriber object:**

SNMP Traps

SNMP messages are sent only if there is an SNMP policy in effect for the receiving server, regardless of the level you choose for the messages. SNMP traffic is affected by both the level you choose and by the SNMP configuration in the policy on the server. There is one SNMP packet per message per destination in the SNMP Trap Target policy. IPX™ addresses are not supported for trap targets.

E-Mail Messages

E-mail messages can also affect network traffic. Like SNMP, e-mail sends only one e-mail per message per e-mail user defined. E-mail is also configured by a server policy. You must define and enable the policy on the sending server for e-mail messages to be sent.

3.11.6 Changing DNS Names or IP Addresses for Tiered Electronic Distribution Servers

Whenever there is a change to the identity of either a Distributor or Subscriber server, you must perform certain tasks so that the distribution processes for these servers can continue as before.

In the distribution process, Tiered Electronic Distribution servers identify themselves to each other by their DNS names or IP addresses. The following sections explain situations that can arise from changing these server identifiers.

If You Are Using DNS Names to Identify Your Servers

- If you change the DNS name of a Distributor server, Subscriber servers cannot recognize the Distributor as a valid source for receiving Distributions.

- If you change the DNS name of a Subscriber server, the Distributor cannot locate the Subscriber server for sending Distributions to it. This is because the Distributor obtains the Subscriber server's address from the eDirectory object.

If you change the IP address of a Distributor or Subscriber server when you are using its DNS name to identify it to Server Management, this change does not affect the distribution processes.

If You Are Using IP Addresses to Identify Your Servers

- If you change the IP address of a Distributor server, Subscriber servers cannot recognize the Distributor as a valid source for receiving Distributions.

- If you change the IP address of a Subscriber server, the Distributor cannot locate the Subscriber server for sending Distributions to it. This is because the Distributor obtains the Subscriber server's address from the eDirectory object.

Because reinstating valid certificates is involved in resolving server identity changes, see for instructions.

3.11.7 When a Tiered Electronic Distribution Process Fails

It is possible, for many common computer-related reasons, that a Tiered Electronic Distribution process could fail. The following are a few possibilities:

- **A Distribution could be interrupted.** If so, when it restarts it picks up where it left off.

  Before distribution, the Distribution package resides at the Distributor. After distribution, the Distribution package still resides at the Distributor with a copy now at the Subscriber. It is during the distribution process that an interruption could halt copying. When the Distributor tries to re-send the Distribution (the next time the Channel schedule starts), it picks up where it left off and does not re-send the entire Distribution.

  If re-sending a Distribution is interrupted, the sender retries every two minutes for 30 minutes. If it is not successful in reestablishing connection to the target server, it stops retrying. The next time the Channel's schedule starts, it picks up where it left off in sending the Distribution when it was originally interrupted.

- **An extraction could be interrupted.** If so, the extraction does not pick up where it left off.

  Distributions are made across the wire from server to server, while extractions are performed on the server from Distributions already sent. Therefore, when an extraction is interrupted, it

simply fails. The Subscriber does not roll back (or undo) the failed extraction, unless the Distribution was a software package (`.cpk` file). It tries the extraction again the next time the Subscriber's extraction schedule starts.

Files are extracted to the volume and directory specified when the Distribution package was created. File groupings and software packages both allow you to specify to which volume and directory the package should be extracted. Therefore, when an interruption occurs during extraction, it fails in the same way as if you were copying a file in the operating system.

- **The File type offers the following:**

   - Retry *X* times
   - Kill the connection on files that are open
   - Error handling (Fail on error; perform a routine on error)

   All options deal with extraction and how to handle it.

3.12 Working Directories

Distributors and Subscribers use working directories on the servers for Distributions, patches, status files, and temporary working files. The size of a working directory is determined by the size and number of Distributions.

The working directories default to the sys: volume on NetWare servers or the C: Drive on Windows servers. Because of disk space considerations on NetWare servers, we recommend that you select a different location on the server, such as a data: volume.

The default working directory names for NetWare and Windows servers are *path*`\zenworks\pds\ted\dist` for the Distributor and *path*`\zenworks\pds\ted\sub` for the Subscriber. For Linux and Solaris servers, the paths are `/var/opt/novell/zenworks/zfs/pds/ted/working/dist` and `/var/opt/novell/zenworks/zfs/pds/ted/working/sub`. You can change working directory names in the properties of the Tiered Electronic Distribution object.

The following sections describe the Tiered Electronic Distribution directory structures:

- Section 3.12.1, "NetWare Distributor Directories," on page 189
- Section 3.12.2, "NetWare Subscriber Directories," on page 191
- Section 3.12.3, "Windows Distributor Directories," on page 192
- Section 3.12.4, "Windows Subscriber Directories," on page 192
- Section 3.12.5, "Linux or Solaris Distributor Directories," on page 193
- Section 3.12.6, "Linux or Solaris Subscriber Directories," on page 193

3.12.1 NetWare Distributor Directories

The following directories are used by NetWare Distributors:

***volume:\installation_path*\zenworks\pds\ted**

Contains the Tiered Electronic Distribution software for the Distributor.

volume:*installation_path*\zenworks\pds\ted\security\private

Contains the Distributor's private key.

volume:*working_directory*

Contains one directory for each Distribution that belongs to the Distributor. The working directory name is user-defined in the Distributor object.

volume:*working_directory**distribution_directory*

Each Distribution has its own directory that is created under the working directory. The Distribution directory's name is derived from the following syntax: *Tree_DN_of_Distributio*n. For example, `TestTree_Files.Distributions.ZENworks.Novell`.

volume:*working_directory**distribution_directory**time_stamp_directory*

Each Distribution directory contains multiple time-stamp directories, which are named according to the date and time the Distribution was built.

Each time a Distribution is built, the Distributor checks to see if anything has changed since the last time the Distribution was built. If so, a new time-stamp directory is created.

The number of time-stamp directories kept is determined by the Maximum Number of Revisions to Keep field in the Distribution object's properties. There are occasions when the number of time-stamp directories exceeds the maximum number specified, because the Distributor does not delete a time-stamp directory that is in use. The Distributor removes the oldest time-stamp directories first.

Sometimes a time-stamp directory name has _temp appended to it. When a Distributor builds a Distribution, it creates a `\*_temp` directory before it determines if anything has changed. If changes are discovered, `\_temp` is removed and the directory is used for the new build.

A Distributor's time-stamp directories contain the files listed in Table 3-18:

Table 3-18   *Files in the Distributor's Time-Stamp Directories*

Filename	Description
`distfile.ted`	The Distribution that was built. All Distributions have the same filename. They are distinguished by their time-stamp directory's name and path.
digest_file	This file only exists if the Distributor Agent creates it (optional).
	Digests are used by Distributors and Subscribers to verify that Distributions have not been tampered with while in transit. The digest provides an MD5 checksum for the Subscriber to compare.
	Digests also detect corruption in a Distribution's package. In the case of corruption, the Subscriber renames the `distfile.ted` Distribution file to `distfile.corrupt` and the Distribution is rebuilt and sent the next time the Channel's schedule fires.
	The syntax for creating the digest filename is:
	%AGENT%AgentDigest.ted
	For example:

	`FTPAgentDigest.ted`
	`HTTPAgentDigest.ted`
	`FileAgentDigest.ted`
	`CPKAgentDigest.ted`

3.12.2 NetWare Subscriber Directories

The following directories are used by NetWare Subscribers:

volume:\installation_path\zenworks\pds\ted

Contains the Tiered Electronic Distribution software for the Subscriber and/or Distributor.

volume:\installation_path\zenworks\pds\ted\security

Contains certificates received from Distributors.

volume:\working_directory

Contains one directory for each Distribution that it receives from a Distributor. The working directory name is user-defined in the Subscriber object.

volume:\working_directory\distribution_directory

Each Distribution has its own directory that is created under the working directory. The Distribution directory's name is derived from the following syntax: *Tree_DN_of_Distributio*n. For example, `TestTree_Files.Distributions.ZENworks.Novell.`

volume*:*working_directory**distribution_directory**time_stamp_directory

Each Distribution directory contains multiple time-stamp directories, which are named according to the date and time the Distribution was built.

The number of time-stamp directories kept is determined by the Maximum Number of Revisions to Keep field in the Distribution object's properties.

After a threshold is met, the Subscriber receives the maximum revision information and deletes the oldest time-stamp directories first.

A Subscriber's time-stamp directories contain the files listed in Table 3-19:

Table 3-19 *Files in the Subscriber's Time-Stamp Directories*

Filename	Description
distfile.ted	The Distribution that was built. All Distributions have the same filename. They are distinguished by their time-stamp directory's name and path.
diststatus.ted	After a Distribution has been successfully received, this file is created.
digest_file	This file only exists if the Distributor Agent has created it (optional). Digests are used by Distributors and Subscribers to verify that Distributions have not been tampered with while in transit. The digest provides an MD5 checksum for the Subscriber to compare. Digests also detect corruption in a Distribution's package. In the case of corruption, the Subscriber renames the distfile.ted Distribution file to distfile.corrupt and the Distribution is rebuilt and sent the next time the Channel's schedule fires.

3.12.3 Windows Distributor Directories

The following directories are used by Windows Distributors:

***installation_path*\\zenworks\\pds\\ted**

Contains the Tiered Electronic Distribution software for the Distributor.

***installation_path*\\zenworks\\pds\\ted\\security\\private**

Contains the Distributor's private key.

3.12.4 Windows Subscriber Directories

The following directories are used by Windows Subscribers:

***installation_path*\\zenworks\\pds**

Contains the Tiered Electronic Distribution software for the Subscriber.

***installation_path*\zenworks\pds\ted\security\private**

Contains certificates received from Distributors.

local_drive*:*working_directory**distribution_directory**time_stamp_directory

Each Distribution directory contains multiple time-stamp directories, which are named according to the date and time the Distribution was built.

3.12.5 Linux or Solaris Distributor Directories

The following directories are used by Linux or Solaris Distributors:

/var/opt/novell/zenworks/zfs/pds/ted/working/dist

Contains the Tiered Electronic Distribution software for the Distributor.

/var/opt/novell/zenworks/zfs/pds/ted/security/private

Contains the Distributor's private key.

Each Distribution directory contains multiple time-stamp directories, which are named according to the date and time the Distribution was built.

3.12.6 Linux or Solaris Subscriber Directories

The following directories are used by Linux or Solaris Subscribers:

/var/opt/novell/zenworks/zfs/pds/ted/working/sub

Contains the Tiered Electronic Distribution software for the Subscriber.

/var/opt/novell/zenworks/zfs/pds/ted/security/private

Contains certificates received from Distributors.

Each Distribution directory contains multiple time-stamp directories, which are named according to the date and time the Distribution was built.

3.13 Editing the Tednode.properties File

If you should install the Subscriber software to a server that does not have a Subscriber object in any eDirectory tree, such as a Windows server in a Microsoft domain, the `tednode.properties` file is used by such a server for its configuration information. When you have Subscriber configuration changes, you need to edit the server's `tednode.properties` file using the information in this section.

The `tednode.properties` file is located in the `\zenworks\pds\ted` directory on the server.

Table 3-20 shows the required format of the file, including comments on some of the entries. The information on the right side of an = symbol is only an example and not the required value for that line. However, the examples are intended to show the correct syntax for the values.

Table 3-20 *Tednode.Properties File Fields*

Line Content	Comments
workingdir = d:\ted\tran	Subscriber's working directory
io.input = 100	Receive rate in bytes per second
io.output = -1	Send rate in bytes per second
variable1 = vol=sys:	Define the variable "vol" with the value "sys:"
variable1.description = Destination Volume	A description of the variable's function
console.level = 6	Message level for the server's console
log.level = 1	Message level for log file
log.days = 1	Number of days to save log file entries
log.path = d:\\ted\\tran\\log.txt	Path for log file and log filename
workorder.timeout = 0	Number of seconds to wait for reply from the Distributor before dropping connection; 0 = wait forever
workorder.concurrent = 0	Concurrent Distributions
email.level = 0	Message level for e-mail
smtp.host = email.novell.com	Location of SMTP host
snmp.level = 0	Message level for SNMP traps
email.target1 = johndoe@novell.com	E-mail address for the messages

For the remaining `tednode.properties` file entries, remove the # (comment) symbol from a line to enable it. This makes that line effective for the schedule type that it is listed under. However, do not remove the # symbol from the first line for a schedule type because it is only a description that indicates the schedule type. You can change the default values that are listed.

The following sample has the Daily schedule enabled because the appropriate # symbols have been removed:

Line Content
Yearly schedule and associated keys (with default values specified)
#schedule.type=yearly
#schedule.month=1
#schedule.day=1
#schedule.begin.hour=8
#schedule.begin.minute=0
#schedule.end.hour=17
#schedule.end.minute=0
#schedule.random=false

```
# Monthly schedule and associated keys (with default values specified)
#schedule.type=monthly
#schedule.day=1
#schedule.begin.hour=8
#schedule.begin.minutes=0
#schedule.end.hour=17
#schedule.end.minute=0
#schedule.random=false

# Daily schedule and associated keys (with default values specified)
schedule.type=daily
schedule.days=Monday,Tuesday,Wednesday,Thursday,Friday,Saturday,Sunday
schedule.begin.hour=8
schedule.begin.minutes=0
schedule.end.hour=17
schedule.end.minute=0
schedule.repeat.days=0
schedule.repeat.hours=0
schedule.repeat.minutes=0
schedule.random=false

# Immediate schedule and associated keys (with default values specified)
#schedule.type=immediately
#schedule.repeat.days=0
#schedule.repeat.hours=0
#schedule.repeat.minutes=0

# Interval schedule and associated keys (with default values specified)
#schedule.type=interval
#schedule.repeat.hours=0
#schedule.repeat.minutes=0

# Never schedule and associated keys (with default values specified)
#schedule.type=never

# Time schedule and associated keys (with default values specified)
#schedule.type=time
#schedule.date.year=2001
#schedule.date.month=1
#schedule.date.day=1
#schedule.begin.hour=8
#schedule.begin.minutes=0
```

Server Policies

4

Novell® ZENworks® Server Management provides server policies for managing server configurations, processes, and behaviors.

The following sections will help you to understand, set up, and configure the policies:

4.1 Understanding Server Policies

In ZENworks 7 Server Management, most policies are enforced through the distribution of policy packages. However, a few policies used by the Distributor are enforced by being associated with Novell eDirectory™ containers.

Review the following sections to understand polices in ZENworks 7 Server Management:

4.1.1 Configuration and Behavioral Management through Server Policies

The Server Policies component provides configuration and behavioral management of your servers. Server policies are divided into three packages for the convenience of scheduling policies and distributing the policies to their applicable servers:

- **Container Package:** Holds the Search policy that determines how Policy and Distribution Services searches eDirectory for objects associated with policies.
- **Service Location Package:** Holds policies specific to running Policy and Distribution Services.

◆ **Distributed Server Package:** Has a generic set of policies that can be applied to all servers, as well as policy package sets for servers on specific platforms. This package provides policies that are distributed for enforcement.

Configuration policies hold information in eDirectory that creates a similar type of configuration on a server, such as enforcing selected SET parameters. Behavioral policies hold a set of rules to be followed under certain situations, such as when a server goes down.

Through server policies you can automate the management of your servers, and through ConsoleOne® and the ZENworks Server Management role in Novell iManager you can configure policies and manage your servers from a single workstation.

4.1.2 Server Policies and Policy Packages

Server policies provide you with the ability to set, standardize, and automate configuration parameters on any given set of servers. You can control the behavior of servers in given situations, such as when downing a server.

The following sections

Creating Policies

To use server policies, you must first create the appropriate Policy Package objects in ConsoleOne, configure the policies that your server needs, enable them, and distribute the package to the applicable Subscriber servers where the package's policies are to be enforced.

Scheduling Policies

When you set up server policies, you can individually schedule them to run daily, weekly, monthly, yearly, by an event, at a specific date and time, relative to a date and time, by an interval of time, or even immediately. The default schedule for the individual policies is the default for the policy package's schedule. Therefore, when you change the package's default schedule, any policy in the package that doesn't have a schedule specified then uses the package's new schedule.

All Enabled Policies Are Enforced

You can implement (enable) any or all of the Policy and Distribution Services policies in a policy package. You can also create a Policy Package object for each different configuration set that you need. For example, you might want some of your servers to be brought down differently, so they would use different policy packages.

All policies enabled in a package are enforced on any servers where the Policy Package Distribution has been received and extracted. In other words, you cannot selectively enforce certain policies in a package. All policies in the package that are enabled are enforced on the server.

Individual Policy Changes Are Not Tied to the Policy Package

Because each policy in a policy package has its own (hidden) object in eDirectory, any changes you make to a policy that are saved when you exit the policy's dialog box (by clicking either OK or clicking Apply then Close), are not undone if you then click Cancel on the policy package's dialog box.

Therefore, clicking Cancel on the properties page for the policy package applies only to the changes you might have made for the package. For example, enabling or disabling a policy, adding or removing added policies.

Disabling a policy does not undo any configurations you made previously in the policy. The policy's configuration changes remain, but are not used because the policy is disabled.

4.1.3 Policy Characteristics

There are two different aspects of policies that determine how you use them:

- "Plural and Cumulative Policies" on page 199
- "Configuration and Behavioral Policies" on page 199

Plural and Cumulative Policies

Policy packages can contain both plural and cumulative policies. All plural policies are also cumulative, but cumulative policies are not necessarily plural. For more detail, review:

- "Plural Policies" on page 199
- "Cumulative Policies" on page 199

Plural Policies

Plural policies are those where there can be more than one per policy package per platform.

For example, in the same policy package, you can add and configure a Scheduled Down policy and name it "Scheduled Down for Time A." Then you could add and configure another Scheduled Down policy, this time naming it "Scheduled Down for Time B."

You can tell if a policy is plural by viewing the Policies tab and clicking Add, because all plural policies are listed in the Add dialog box.

Cumulative Policies

Cumulative policies are those that allow multiples of the same policy to be in effect when multiple policy packages are distributed to a server. For example, a Text File Changes policy distributed to Server A could be accumulated with a differently configured Text File Changes policy distributed to Server A. All of the text file changes from both policies would be effective for Server A.

Configuration and Behavioral Policies

A single configuration policy can affect the configuration of a single server or many servers. For example, you can schedule a policy to run at regular intervals to ensure that the server's configuration continues to be set correctly.

Behavioral policies hold a set of rules to be followed in certain situations. The policy engine carries out these rules, along with any of its supporting modules.

For example, the Server Down Process policy defines criteria that must be met before you bring the server down, such as:

- How soon before the server is brought down should users be notified
- Who is notified when the policy is being enforced
- Which peer server is to send SNMP alerts if the server does not come back up

IMPORTANT: For Linux and Windows servers, any downing command entered locally on those servers cannot be intercepted by the Server Down Process policy. NetWare servers use APIs that enable the policy to intercept the action. For the Server Down Process policy to work for the Linux and Windows server platforms, they must be downed using iManager where the action can be detected by the policy.

Behavioral policies are designed to make servers act more intelligently, to handle situations an administrator might not even be aware of, and to reduce complexity for administrators.

In summary, the benefits of configuration and behavioral policies include:

- Automating tasks that an administrator would normally perform
- Notifying specified users through e-mail messages that a server is going down
- Allowing a server down process to abort on certain conditions

4.1.4 Server Policies Architecture

To understand how server policies are used to manage your servers, you must understand its eDirectory objects and its agent:

-
-

eDirectory Schema Extensions for Server Policies

The eDirectory schema extensions included in the Server Policies component define the class of eDirectory objects that are created in your eDirectory tree, including which information is required or optional at the time the object is created. Every object associated with the Server Policies component in an eDirectory tree has a class defined for it in the tree's schema.

Server Management objects for the eDirectory schema are:

Container Package
Server Package
Service Location Package
Distributed Server Package
ZENworks Database

Note the following concerning policy enforcement:

- All of the policies in the Distributed Server Package must be distributed to be enforced

◆ All of the policies in the Container Package, Server Package, and Service Location Package must be associated to be enforced

Existing eDirectory classes that are modified with the addition of Server Management attributes are:

Country
Group
Locality
Organization
Organizational Unit
Server

The following sections summarize the primary eDirectory objects that are added to eDirectory from the schema extensions provided with the Server Policies component:

◆ "Container Package Object" on page 201
◆ "Server Package Object" on page 201
◆ "Service Location Package Object" on page 201
◆ "Distributed Server Package" on page 202
◆ "ZENworks Database Object" on page 202

For basic information about the types of objects in an eDirectory tree, see the Novell NetWare Documentation Web site (http://www.novell.com/documentation/lg/nw5/docui/index.html) and select *Procedures > Planning > Directory Services > eDirectory Planning*.

Container Package Object

The Container Package object is an eDirectory object that manages the Search policy object. This policy is used by the Distributor and Subscriber objects for all versions of Server Management, and must be associated to be enforced.

Server Package Object

The Server Package object is an eDirectory object that manages the following policy objects for ZENworks Server Inventory:

Rollup Policy
zeninvDictionaryUpdatePolicy
ZENworks Database

All policies in this package must be associated to be enforced.

Policy and Distribution Services does not use this package.

Service Location Package Object

The Service Location Package object is an eDirectory container object that manages the following policy objects:

SMTP Host
SNMP Trap Targets
Tiered Electronic Distribution
ZENworks Database

Service Location Package policies provide general Policy and Distribution Services configuration and location information.

All policies in this package must be associated to be enforced.

All policies are used by ZENworks 7 Server Management Distributors and Subscribers.

Distributed Server Package

The Distributed Server Package object is an eDirectory object that manages the following policy objects (ZENworks 7 Server Management only):

Copy Files
NetWare Set Parameters
Prohibited File
Scheduled Down
Scheduled Load/Unload
Server Down Process
Server Scripts
SMTP Host
SNMP Community Strings
SNMP Trap Targets
Text File Changes
ZENworks Database
ZENworks Server Management

Distributed Server Package policies are used for configuring servers, controlling server behavior, and providing general Server Management configuration and location information.

All policies in this package must be distributed to be enforced.

ZENworks Database Object

Provides the location of the `zfslog.db` file for logging reporting information. You can install the database file on only NetWare® and Windows servers.

The ZENworks Database object can exist multiple times in a tree, each with its own associated database file; however, there can only be one database file installed per server.

The Server Policies component writes policy information to the Server Management database file (`zfslog.db`). Because every server in your network can be running the Policy/Package Agent, they can each write to the database, even across WAN links. If you do not need consolidated server policies reports on all servers, you can install a database to each WAN segment.

If you require consolidated server policies reports, you can have just one `zfslog.db` file where all servers running the Policy/Package Agent can log information. The amount of data a Policy/Package Agent writes to the database might not create excessive WAN traffic, depending on the number of servers and speeds of the WAN links.

Because you can install the Server Management database to multiple servers, to minimize WAN traffic you should coordinate the placement of Policy Package and ZENworks Database objects in containers on the WAN segments.

Policy/Package Agent

Policy and Distribution Services allows you to manage your network servers using the Policy/Package Agent. This agent is installed on each server where you select the Subscriber/Policies installation option.

The Policy/Package Agent does the following:

- Extracts (installs) a software package's contents.
- Extracts the policy information from a Policy Package Distribution.
- Enforces the enabled policies from the extracted policy information based on their enforcement schedules.

There are a number of server policies that provide configuration and behavioral management of your servers. The Policy/Package Agent must be running on each server you want to manage with policies or have software packages to extract and install.

You should install the Policy/Package Agent to every server in your network. Exceptions might be servers where you do not need to distribute software packages, or servers that you do not want to manage using policies.

4.1.5 Enforcing Policies

Most ZENworks 7 Server Management policies are enforced by creating the policy package, enabling and configuring the policy, scheduling the package, distributing the package, and extracting the policies on servers.

Some ZENworks 7 Server Management policies are enforced by creating the policy package, enabling and configuring the policy, scheduling the package, and associating the package with the containers where the Distributor or Subscriber objects reside.

For more information, review the following:

Scheduling Policies

Some server policies must be scheduled before they can be enforced.

The following schedules are available:

- Activate by the Default Package Schedule (which you can set to any of the schedules)
- Activate on a specified event (such as running at system startup or shutdown)
- Activate once relative to a period of time
- Activate at a specified date and time
- Activate once per year at a specified time
- Activate once each month at a specified time
- Activate on one or more days of the week at specified times
- Activate on one or more days of the week, repeating at a specified interval of time

- Continuously repeat at a specified interval of time
- Run immediately
- Run immediately, repeating at a specified interval of time

IMPORTANT: If you enable a policy, but do not schedule it, it activates according to the schedule currently specified in the Default Package Schedule.

The Default Package Schedule provides a default for unscheduled policies in the policy package. The default schedule is the Run At System Startup event.

Distributing Policies

After you have enabled and configured a policy contained in the Distributed Server Package, you must distribute its policy package to the Subscriber servers where the enabled policies are placed into effect. In other words, configuring and enabling a policy only sets up the policy. It is enforced through its distribution to and extraction on the applicable servers that are running Policy and Distribution Services.

Associating Policies

After you have enabled and configured a policy contained in the Service Location Package, you must associate its policy package with the containers where Distributor or Subscriber objects reside so that the enabled policies are placed into effect. This association can be directly with a container where the Distributor or Subscriber objects reside, or with a container higher in the tree from where the container holding these objects reside.

Because configuring and enabling a policy only sets up the policy, it is enforced through its association with the applicable servers that are running Policy and Distribution Services.

4.1.6 Server Policy Descriptions

The tables in the following sections list the server policies by policy package. The second column indicates whether a policy is a configuration or behavioral policy, and whether it is cumulative, plural, or both.

- "Container Package" on page 205
- "Service Location Package" on page 205
- "Server Package" on page 206
- "Distributed Server Package" on page 207

Container Package

Table 4-1 *Container Package Policy*

Policy Name	Policy Type Keys	Policy Function
Search	Behavioral	If you don't set a Search policy, the default is to search from the parent container to the root every hour. This can create unnecessary search traffic. Therefore, we recommend that you make effective use of the Search policy.
		This Search policy can only be administered in ConsoleOne. A Search policy created in NetWare Administrator for ZENworks is not recognized in Server Management.

Because most policies in Server Management are distributed rather than associated for enforcement and a Distributor does not receive Distributions, the Search policy is used in Server Management to enable the Distributor Agent to locate and use policies in the Service Location Package. For example, the Distributor Agent can use the package's ZENworks Database policy to write reporting information to the ZENworks Server Management Database file.

Also, Distributors read the Service Location Package policies for their Subscribers. That means Subscribers receive their Service Location Package policies through associations, as well.

Service Location Package

Table 4-2 *Service Location Package Policies*

Policy Name	Policy Type Keys	Policy Function
SMTP Host	Configuration	Sets the TCP/IP address of the relay host that processes outbound Internet e-mail. This policy must be enabled if you select the E-Mail option for notifying or logging messages in any of the other policies.
SNMP Trap Targets	Configuration	Sets SNMP trap targets for associated eDirectory objects.
		In ZENworks 7 Server Management, you can schedule this policy for when you want it to be refreshed.
		IPX™ addresses are not supported for SNMP trap targets. You can only use IP addresses and DNS names.

Policy Name	Policy Type Keys	Policy Function
Tiered Electronic Distribution	Configuration	Sets defaults for the Distributor and Subscriber objects, including:
		I/O rates
		Maximum concurrent Distributions
		Connection time-out in minutes
		Working directory
		Parent Subscriber
		Messaging levels for a server's console, SNMP traps, log files, and e-mail notification
		Extraction Schedule
		Refresh Schedule
		Variables
		Any defaults set here override unchanged defaults in a Tiered Electronic Distribution object. However, if a Tiered Electronic Distribution object's properties are modified, those modifications have precedence over any defaults set in the Tiered Electronic Distribution policy.
ZENworks Database	Configuration	Sets the DN for locating the ZENworks Database object and the database file. The database is used for logging successes and failures that are used in creating reports.
		This policy can be created to override the database settings that might have been established during installation of Policy and Distribution Services.
		The Policy/Package Agent and the Distributor Agent both write to `zfslog.db`. For information on having these agents write to different database files, see Section 10.1.7, "Coexisting Databases," on page 361.

Server Package

The Server Package exists in ZENworks 7 Server Management only for use by Server Inventory. The ZENworks Database policy contained in this package is automatically created by the installation program when Server Inventory is installed to enable automatic location of the database for logging inventory data.

Policy and Distribution Services does not use this package.

Although other policies exist in this package, Table 4-3 only lists the ZENworks Database policy.

Table 4-3 *Server Package Policy*

Policy Name	Policy Type Keys	Policy Function
ZENworks Database	Configuration	Sets the DN for locating the ZENworks Database object. This policy must be in effect for Server Inventory to locate a database for logging inventory data.

Distributed Server Package

This package contains the policies the must be distributed to Server Management servers to be enforced on them.

Table 4-4 *Distributed Server Package Policies*

Policy Name	Policy Type Keys	Policy Function
Copy Files	Plural Cumulative Configuration	Enables copying of files on a server from one location to another by using policy configurations.
NetWare Set Parameters	Plural Cumulative Configuration	Specifies and optimizes selected Set Parameters for a server or group of servers. For the NetWare platform only.
Prohibited File	Plural Cumulative Configuration	Monitors and enforces the deletion or moving of unauthorized files from a specified volume/drive or directory/folder.
Scheduled Down	Plural Cumulative Configuration Behavioral	Schedules when a server should go down, and whether it should be automatically brought back up. The policy includes which command to use in bringing it down (RESET, RESTART, or DOWN).
Scheduled Load/ Unload	Plural Cumulative Configuration	For automating the loading and unloading order of NLM™ and Java Class processes for the selected servers, and for starting and stopping Windows services. NLM files that require user input to unload cannot be automated.
Server Down Process	Behavioral	For controlling which processes to follow and which conditions to meet before downing a server.
Server Scripts	Plural Cumulative Configuration	For automating script usage on your servers.
SMTP Host	Configuration	Sets the TCP/IP address of the relay host that processes outbound Internet e-mail. This policy must be enabled if you select the E-Mail option for notifying or logging messages in any of the other policies.
SNMP Community Strings	Configuration	Allows you to receive and respond to SNMP requests.
SNMP Trap Targets	Configuration	Sets SNMP trap targets for associated eDirectory objects. You can schedule this policy for when you want it to be refreshed. IPX addresses are not supported for SNMP trap targets. You can only use IP addresses and DNS names.
Text File Changes	Plural Cumulative Configuration	For automating changes to text files.

Policy Name	Policy Type Keys	Policy Function
ZENworks Database	Configuration	Sets the DN for locating the ZENworks Database object and the database file. The database is used for logging successes and failures that are used in creating reports.
		This policy can be created to override the database settings that might have been established during installation of Policy and Distribution Services.
		The Policy/Package Agent and the Distributor Agent both write to `zfslog.db`. For information on having these agents write to different database files, see Section 10.1.7, "Coexisting Databases," on page 361.
ZENworks Server Management	Configuration	Basic configuration parameters for Policy and Distribution Services, such as status logging, defining the server console prompt for the Policy/Package Agent, setting its working path, and setting a database purging limit.
		You can enable this policy on each server where you want to enforce server policies. However, if you do not enable the policy, Policy and Distribution Services works from pre-programmed defaults.

4.2 Creating a Policy Package

Policy and Distribution Services groups its server policies into three Policy Package objects:

- ◆ Container Package
- ◆ Service Location Package
- ◆ Distributed Server Package (ZENworks 7 Server Management only)

You can place policy packages anywhere in the tree. For ease of management, we recommend that you create an OU container for grouping the policy packages. For example, Policies.

However, if you install ZENworks Desktop Management to your tree, you could keep the Server Management and Desktop Management policies in separate containers, such as Server_Policies and Desktop_Policies.

IMPORTANT: If you have partitions that are accessed across a WAN, make sure that the Policy Package objects are in the same partition as the Server object to ensure that the Policy/Package Agent loads. Also make sure that the Search policy does not require searching outside the partition where the Server object exists.

To determine which Policy Package objects to create, first determine which policies you need.

To create Policy Package objects, review the instructions in the following sections:

- ◆ Section 4.2.1, "Creating a Policies Container," on page 209
- ◆ Section 4.2.2, "Creating a Policy Package Object," on page 209

4.2.1 Creating a Policies Container

To create the OU container object for holding your Policy Package objects:

1 In ConsoleOne, right-click the container where you want the policies container located.

> **IMPORTANT:** Where you create the OU, and how many characters you use to name it, directly affects the number of characters that you have available for naming the plural policies. eDirectory has a 64-character limit for the full name and path in the tree for a policy.
>
> Because you can have many different versions of one plural policy in a single policy package, you want to be able to name them descriptively. Therefore, place the OU as high in the tree as is logical, and give it a short name to provide as many characters as possible for naming the policies.

2 Click *New > Object*, then select *Organizational Unit*.

3 Provide a name for the OU, then click *OK*.

4.2.2 Creating a Policy Package Object

To create a Policy Package object:

1 In ConsoleOne, right-click the container you created for the Policy Package objects, click *New*, then select *Policy Package*.

The Policy Package Wizard opens.

2 Under *Policy Packages*, select a policy package, then click *Next*.

Available packages include: Container, Server, Service Location, and Distributed Server.

3 Provide a name for the package, then click *Next*.

Because you can have multiples of the same package type, use a unique, informative name for each package.

> **IMPORTANT:** Because of the eDirectory 64-character path/name limit, and the package name you provide here is part of the path for plural policies that you can create later, provide a brief, but unique, Policy Package object name so that you can have as many characters as possible to be available for giving descriptive plural policy names.

4 Repeat Step 2 and Step 3 for each package to be created.

Select the *Create Another Policy Package* check box to save repeating Step 1.

4.3 Configuring Server Policies

You can configure server policies for containers, servers, and service locations. The policies allow you to automate use of NetWare functionality. See your NetWare documentation for specific information.

To configure server policies, review the instructions in the following sections:

- Section 4.3.1, "Compiling Zentrap.mib," on page 210
- Section 4.3.2, "Configuring the Container Package Policy," on page 210

For information on scheduling server policies, see .

4.3.1 Compiling Zentrap.mib

The SNMP Community Strings and SNMP Trap Targets policies utilize SNMP. `Zentrap.mib` is located on the *Program* CD under `\zfs\tedpol\sfiles\mibs`.

To receive SNMP traps on your SNMP management console, you must copy the `zentrap.mib` file from the *ZENworks 7 Server Management with Support Pack 1 Program* CD to the location that your management console uses to manage MIBs, then compile it. Your SNMP management console can then receive and interpret SNMP traps from Server Management.

4.3.2 Configuring the Container Package Policy

The Search policy is used by the Distributor for information on how to read the eDirectory tree when the Distributor has been refreshed.

IMPORTANT: If you do not use the Search policy, Server Management searches up to [Root] and reads the objects every hour. Be sure to configure and enable the Search policy to limit unnecessary search traffic.

To configure the Search policy:

1 In ConsoleOne, right-click the *Container Package*, click *Properties*.

2 Select the *Policies* tab, select the check box for *Search Policy*, click *Properties*, then select the *Search Level* tab.

If the box under the *Enabled* column is not selected for the Search policy, select it before clicking *Properties*. A policy must be enabled to activate the *Properties* button.

3 To determine the upper limits of the search policy, select one of the following:

Search Location	Description
Object Container	Search to the parent container of the Server object
Partition	Search to the Partition Root
Selected Container	Search to the selected container
[Root]	Search to the root of the tree

If you chose *Selected Container*, browse to select the container.

To determine searching limits in either direction of the item selected, enter a number. For example:

#	Description
0	Limits the search to the current level (as set in the Search For Policies Up To field).
1	Limits the search to one level above the current level (as set in the Search For Policies Up To field). For example, if you specify the server's parent container in the Selected Container field, +1 would limit the search to one level above the parent container.
–1	Limits the search to one level below the chosen search level (as set in the Search For Policies Up To field). For example, if you select [Root] in the Search For Policies Up To field, -1 would allow searching up to one level below [Root].

4 To determine the search order, select the *Search Order* tab.

Type	Description
Object	Server
Group	Server Group
Container	Container of Servers

Use the arrow keys to change the order. You can also click *Add* or *Remove* to change which object types are used.

5 (Optional) Because policies are refreshed when they are received at the Subscriber, specify a refresh frequency.

The default is once every hour.

If you leave both time increments at zero (days and hours), policies are not refreshed from eDirectory, even if you have *Policy Manager Will Refresh Policies From eDirectory* selected.

Changes made to enabled policies are not enforced until they are refreshed at the given refresh interval. However, you can manually refresh all policies using the POLICY REFRESH command at the server console. The refresh rate is listed in seconds at the server console (1 hour = 3600 seconds).

6 Click *OK* to close the policy.

If you click *Cancel*, none of the Search policy changes made on any of the tabs are saved.

7 To associate the policy package so that the Search policy is enforced on the Distributor, select the *Associations* tab, then click *Add*.

8 Browse to select the container where the Distributor object resides (or any container above it), then click *OK*.

If you click *Cancel*, the association you made is not saved.

4.3.3 Configuring Service Location Package Policies

Because the Distributor does not receive Distributions, policies for a Distributor must be associated with the container where its object resides. The Service Location Package contains policies used by the Distributor.

To configure Service Location Package policies, review the following sections:

- "SMTP Host" on page 212
- "SNMP Trap Targets" on page 212
- "Tiered Electronic Distribution" on page 213
- "ZENworks Database" on page 218

SMTP Host

Sets the TCP/IP address of the SMTP relay host that processes outbound Internet e-mail. This policy must be enabled if you select the E-Mail option for notifying or logging messages for the Distributor.

To configure the SMTP Host policy:

1 In ConsoleOne, right-click the Service Location Package, then click *Properties*.

2 Select the SMTP Host policy's check box, then click *Properties*.

If the box under the *Enabled* column is not selected for the SMTP Host policy, select it before clicking *Properties*. A policy must be enabled to activate the *Properties* button.

3 Provide the TCP/IP address or DNS name of the relay host server, then click *OK*.

4 To associate the policy package so that the SMTP Host policy is enforced on the Distributor, select the *Associations* tab, then click *Add*.

5 Browse to select the container where the Distributor object resides (or any container above it), then click *OK*.

If you click *Cancel*, the association you made is not saved.

SNMP Trap Targets

Use this property page to establish the targets (or locations) where you want SNMP traps sent from the Distributor. Each target must be a valid TCP/IP address or DNS name.

Make sure that you have compiled `zentrap.mib` (see Section 4.3.1, "Compiling Zentrap.mib," on page 210).

To configure the SNMP Trap Targets policy:

1 In ConsoleOne, right-click the *Service Location Package*, then click *Properties*.

2 Select the SNMP Trap Targets policy, then click *Properties*.

If the box under the *Enabled* column is not selected for the SNMP Trap Targets policy, select it before clicking *Properties*. A policy must be enabled to activate the *Properties* button.

3 To add items to the *SNMP Trap Targets* list on the *SNMP Trap Policy* tab, click *Add*.

4 On the SNMP Target dialog box, provide valid a TCP/IP address or DNS name, then click *OK*.

5 Repeat Step 3 and Step 4 for each trap target to be added.

6 To schedule the policy, select the *Schedule* tab, select a type in the *Schedule Type* field, then configure the schedule:

7 Click *OK* when finished.

8 To associate the policy package so that the SNMP Trap Targets policy is enforced on the Distributor, select the *Associations* tab, then click *Add*.

9 Browse to select the container where the Distributor object resides (or any container above it), then click *OK*.

If you click *Cancel*, the association you made is not saved.

Tiered Electronic Distribution

This policy allows you to set default values for the attributes of Distributors and Subscribers.

How the Policy Works

The default values set in the Tiered Electronic Distribution policy become effective when you associate the Service Location Package that contains this policy to a container above where the Distributor and Subscriber objects reside, or to the container where Subscriber objects reside.

The values in the attributes of the Tiered Electronic Distribution policy automatically replace the similar values for the Distributor and Subscriber objects, but only if the default values of those attributes have never been changed in the object's properties.

After you have changed the values of the attributes in the Distributor or Subscriber objects and you want to use the values in the Tiered Electronic Distribution policy, then you must edit the Distributor or Subscriber object's properties and select the Use Policy check box at the top of each tab in the object's properties that contains the check box. Then the Tiered Electronic Distribution policy values will appear in the Distributor or Subscriber object's attributes.

Cumulative Policies

Tiered Electronic Distribution policies are not cumulative, meaning:

* **One at a time:** You cannot have more than one Service Location Package (containing the Tiered Electronic Distribution policy) associated to the same container.

* **Closest wins:** If the Subscriber's container is associated with a Tiered Electronic Distribution policy (in the Service Location Package) and a parent container also has a Tiered Electronic Distribution policy (in the Service Location Package) associated with it, the Tiered Electronic Distribution policy of the closest container (the Subscriber's own container) prevails.

Replacing, Adding, or Losing Property Values

The following information applies only where the Tiered Electronic Distribution policy is in effect:

* You can add the Variables defined in the Tiered Electronic Distribution policy to the Distributor or Subscriber's list of variables. They do not replace the variables already defined in the Distributor or Subscriber object.

* For all other policy fields that coincide with values in a Distributor's or Subscriber's properties, the Tiered Electronic Distribution policy replaces, not supplements, them, including the possibility of replacing property values with empty fields. Therefore, if you create a Tiered Electronic Distribution policy, make sure you fill in all of the fields on every tab in the policy that you want to be applied to the affected Distributors or Subscribers.

  For example, if your Subscriber has a working directory entered in its object's properties, and you do not provide a working directory in the Tiered Electronic Distribution policy, then later apply the policy by selecting the Use Policy check box on the Subscriber's properties, the Subscriber will no longer have a working directory available to it.

Multiple Policies for Platform Configurations

You can have multiple instances of the Tiered Electronic Distribution policy for your Subscriber objects for the purpose of defining different policy settings for different server platforms. To do this, you must have created the Subscriber objects in different containers representing their respective operating systems.

Subscriber attributes that could require operating system-specific values are:

working directories
messaging settings
variables definitions

Configuring the Tiered Electronic Distribution Policy

1 In ConsoleOne, right-click the *Service Location Package*, then click *Properties*.

2 Select the Tiered Electronic Distribution policy, then click *Properties*.

  If the box under the *Enabled* column is not selected for the Tiered Electronic Distribution policy, select it before clicking *Properties*. A policy must be enabled to activate the *Properties* button.

3 Click *General > Settings* and fill in the fields:

Input rate: Sets the default input rate to minimize network traffic for Tiered Electronic Distribution objects. This determines the receive rate for Subscribers and Distributors. The default value is the maximum that the connection can handle. You can use this rate to control the use of narrow bandwidth links.

Output rate: Sets the default output rate to minimize network traffic for Tiered Electronic Distribution objects. This determines the send rate for Distributors and parent Subscribers. The default value is the maximum that the connection can handle. Blank means that bandwidth is taken from third-party applications.

There are three output priorities where you can specify a rate:

- ◆ **High priority:** These Distributions are sent before any Medium or Low priority Distributions.

- ◆ **Medium priority:** These Distributions are sent after all High priority and before any Low priority Distributions.

- ◆ **Low priority:** These Distributions are sent after all High and Medium priority Distributions.

For more information, see Section 3.4.5, "Prioritizing Distributions," on page 127.

Maximum concurrent Distributions to build: Specifies the maximum number of distribution threads that can be running concurrently for building Distributions. The default value is 5. Valid values are from 1 to 10.

This number can help in load-balancing a Distributor's building activity.

Maximum concurrent Distributions to send: Specifies the maximum number of distribution threads that can be running concurrently for sending Distributions. The default value is unlimited (a blank field).

This number can help in load-balancing a Distributor's sending activity and spread network traffic over an entire scheduling window.

Connection time-out: Specifies a default number of seconds before the Distributor times out when connecting to another node, or specifies the number of seconds a Subscriber waits for a response from a Distributor (receiving) or a Subscriber (sending) before ending the connection.

After the time has transpired, a Distributor ends the connection and does not retry until the Channel's Send schedule starts again. If a connection is ended during sending or receiving, a Subscriber does not start again until the next time the Channel's Send schedule starts.

The default value is 300 seconds (five minutes). The available range in seconds is 1 to 60,000. You should select a reasonable time to wait for a response from one node to another.

IMPORTANT: This interval must be increased on slow or busy links where longer delays are frequent.

Working directory: Provide a default Tiered Electronic Distribution directory to store Distributions, persistent status, and temporary files on a server. The directory needs to be located where there is enough free space to handle processing of Distributions.

The Working Directory field allows the use of variables to specify the volume/drive and directory names. However, variables only work with Subscribers.

IMPORTANT: Distributors are not able to resolve variables and use exactly what is specified in the Working Directory field. For example, if the value was `%VOL%ted1\working`, the Distributor would create a working directory on the sys: volume named `sys:\%VOL%\ted\working`, because it could not resolve %VOL%.

For more information, see .

Parent Subscriber: Subscribers should generally not receive their Distributions directly from a Distributor. You can browse for a Subscriber to be the default parent Subscriber for your whole network, which passes on Distributions when a Subscriber object might not have a parent Subscriber defined in its properties.

Disk space desired to be left free: Use this as the default value to ensure there is enough free disk space for receiving Distributions where you might not have this value defined in a Subscriber object's properties. A Subscriber does not attempt to receive a Distribution if the disk space value set here is insufficient.

4 Click *General > Messaging* and fill in the fields:

Server console: Procedure to follow when displaying messages at the server console. The default is Level 4 (Information & Level 3 Messages).

SNMP trap: Procedure to follow when sending SNMP traps. The default is Level 0 (No Messages).

Log file: Procedure to follow when recording information to a log file. The default is Level 5 (Trace Information & Level 4 Messages).

Filename: By default, this field is blank. Whatever log filename you select, it replaces `ted.log` for the servers where this policy is enforced.

To create a log file, specify the log file's filename using the following format:

installation_path*directory_path**filename*.*filename_extension*

The *installation_path* is not required for ZENworks to locate the log file, but it is easier for you to locate the file if the path is included.

IMPORTANT: Because the log file can become quite large, for NetWare servers we recommend that you do not use the sys: volume.

Use filename extensions such as `.log` or `.txt`.

Delete log entries older than __ days: Controls disk space usage. For log files, it is important to set the message levels at minimal detail and to purge entries older than six days (the default).

E-mail: Procedure to follow when sending e-mail messages. None or Errors Only are recommended to minimize unnecessary e-mail traffic. The default is Level 0 (No Messages).

Users: Add users, groups, or e-mail addresses.

Address attribute: Displays the attribute of the associated user or group. You can change the attribute from the drop-down list, which displays over three dozen options.

Following are some of these options:

CN	Given Name	Postal Code
Description	Initials	Postal Office Box
EMail Address	Internet EMail Address	Surname
Full Name	Mailbox ID	Telephone Number
Employee ID	NSCP:mailHost	Title
Entrust:User	OU	uniqueID
Generational Qualifier	Physical Delivery Office Name	

5 To assign default values to variables used by the Subscriber, select the *Variables* tab, click *Add*, then fill in the fields:

Variable: Name of the variable. It should indicate how the variable is used. For example, WORKINGVOL.

The variable name can be derived from predefined and user-defined variables.

Value: The value that the Subscriber uses when this variable is specified. For example, data:.

A value can be another variable name. You can nest variables using this method.

To ensure that extraction takes place, provide an absolute path to the Subscriber. For example, if the path is only the data: volume, make sure the colon (:) is included, because it is a necessary part of the full path.

Description: Describes how the variable is used. For example:

```
Volume for the working directory.
```

If a variable defined here does not exist in a Subscriber's variables list, it is automatically added. However, if the variable does exist in the Subscriber's variables list, the definition in the Subscriber prevails.

6 To assign a default refresh schedule for all Distributors, select the *Schedule* tab, click *Distributor Refresh Schedule*, select a schedule in the *Schedule Type* field, then configure the schedule:

For information on the refresh schedule, see .

IMPORTANT: We recommend the Distributor's Refresh schedule be daily, unless changes to Distributions warrant a more frequent refresh. However, do not refresh the Distributor more often than every five minutes. The following can need up to five minutes to complete their processes: Distribution building, eDirectory replication, and tree walking (when no Search policy is defined).

7 To assign a default extraction schedule for all Subscribers, select the *Schedule* tab, click *Subscriber Extract Schedule*, select a schedule in the *Schedule Type* field, then configure the schedule:

8 Click *OK* to close the policy.

9 To associate the policy package so that the Tiered Electronic Distribution policy is enforced on the Distributor, select the *Associations* tab, then click *Add*.

10 Browse to select the container where the Distributor object resides (or any container above it), then click *OK*.

If you click *Cancel*, the association you made is not saved.

11 To associate the policy package so that the Tiered Electronic Distribution policy is enforced on a Subscriber, select the *Associations* tab, then click *Add*.

12 Browse to select the container where Subscriber objects reside (or any container above it), then click *OK*.

This should be the Subscribers where you want the Tiered Electronic Distribution policy's default information to be available.

If you are creating this policy for a particular operating system, make sure you select the correct platform-specific container, and the policy applies only to the Subscribers under that container.

If you click *Cancel*, the association you made is not saved.

13 Repeat Step 12 for each container where Subscribers exist that you want to use this policy.

ZENworks Database

Sets the DN for locating a ZENworks Database object. If you did not establish this information when installing Policy and Distribution Services, you can create this policy to enable Server Management to locate a database file for logging successes and failures that are used in creating reports. You can also create this policy to override the information established during installation.

Use this property page to select the database object to be associated with the current ZENworks Database policy. The policy is not in effect until you have distributed the policy to the Subscribers, or associated the policy with the Distributor.

The Server Management database is used to store reporting information for Distributions and Server Policies.

To configure the ZENworks Database policy:

1 In ConsoleOne, right-click the *Service Location Package*, then click *Properties*.

2 Select the ZENworks Database policy, then click *Properties*.

If the box under the *Enabled* column is not selected for the ZENworks Database policy, select it before clicking *Properties*. A policy must be enabled to activate the *Properties* button.

3 Select the *Policy/Distribution Management* tab.

4 In the *Database DN* field, browse for the ZENworks Database object that represents the database for this policy, then click *OK*.

5 To associate the policy package so that the ZENworks Database policy is enforced on the Distributor, select the *Associations* tab, then click *Add*.

6 Browse to select the container where the Distributor object resides (or any container above it), then click *OK*.

If you click *Cancel*, the association you made is not saved.

4.3.4 Configuring Distributed Server Package Policies

You can configure Distributed Server Package policies to automate control of various server behaviors and processes and to automate control of SMTP Host TCP/IP addresses, SNMP Trap Targets, and the ZENworks Database object's DN.

There are several Policies tab options for server policies, one for each supported operating system. The policies that are available on the General tab apply to servers on all platforms. The policies available on the specific platform tabs apply only to the servers for those platforms.

Platform-specific policies, such as those on the NetWare tab, always override similar policies on the General tab for a particular policy package.

All policies are contained in the NetWare policies. Therefore, only the NetWare policies are documented here. The information applies equally to each platform.

To configure Distributed Server Package policies, review the following sections:

- "Copy Files" on page 220
- "NetWare SET Parameters" on page 221
- "Prohibited File" on page 222
- "Scheduled Down" on page 225
- "Scheduled Load/Unload" on page 225
- "Server Down Process" on page 226
- "Server Scripts" on page 228
- "SMTP Host" on page 229
- "SNMP Community Strings" on page 229
- "SNMP Trap Targets" on page 233
- "Text File Changes" on page 234
- "ZENworks Database" on page 235
- "ZENworks Server Management" on page 236

Copy Files

The Copy Files policy enables copying of files on a server from one location to another by using policy configurations. You can either copy or move the files.

To configure the Copy Files policy:

1 In ConsoleOne, select the Distributed Server Package's container, right-click the *Distributed Server Package*, then click *Properties*.

2 Select the *Policies* tab, then select the platform from:

General
Windows
NetWare
Linux
Solaris

3 Click *Add*, click *Copy Files*, provide a policy name, then click *OK*.

4 Click *Properties*.

The *Copy Files* tab displays.

5 Click *Add*.

Local File Copy #1 defaults. You can edit that name.

6 Fill in the fields:

Source path: Provide the full path where the files to be copied are located.

You can use wildcards in the path:

* = any number of characters
? = any single character in that position
??? = any characters in those positions

Target path: Provide the full path where the copied files are to be placed.

You can use wildcards in this path. This path does not need to mirror the source path. However, you could mirror an existing target path.

Include subdirectories: Includes all subdirectories and their files beginning from the directory at the end of the path; otherwise, only the files in the directory at the end of the path are copied.

Maintain attributes: Maintains the file attributes in the target's file system that exist in the source's file system.

Overwrite destination files: Overwrites files of the same name in the destination directories, regardless of differences in file dates. If you do not select this option, files of the same name is not replaced.

Maintain trustees: Maintains the file's trustee attributes.

When a file is locked: Select one or both:

◆ **Retry __ times:** Retries overwriting a locked file the number of times you select before failing to replace the file. Leave this check box deselected to not replace locked files on the target file system.

◆ **Kill connection of open files:** (NetWare only) Attempts to kill the connection of locked files so they can be overwritten. This applies only to files being extracted, not to files being accessed to build the Distribution. If a file belonging to a Distribution is locked

when the Distribution is being built, the build fails. Server and NLM connections cannot be killed.

Error processing: Fail On Error is selected by default. This stops the file copying process when an error is encountered in copying. To continue file copying when an error is encountered, select Continue On Error.

Operation: Sets whether to copy or move the files identified in the Source Path.

7 Select the *Schedule* tab, then schedule the policy (see).

8 Click *OK* to close the policy.

NetWare SET Parameters

You can automate the use of SET parameters by your servers.

To configure NetWare SET parameters:

1 In ConsoleOne, select the Distributed Server Package's container, right-click the *Distributed Server Package*, then click *Properties*.

2 Click *Policies > NetWare* (or *General*).

3 Click *Add*, then select *NetWare Set Parameters*.

4 Provide a name for this SET parameters policy, then click *OK*.

Because the policies selected from this dialog box are plural, you can have multiple SET parameter policies listed on the Policies tab. Therefore, provide a unique name for this policy.

When you click OK after naming the SET parameters policy, it is selected on the Policies tab.

5 Click *Properties*.

The *Set Commands* tab displays.

6 Click *Add*.

The NetWare Server SET Command Wizard opens.

7 Select the server containing the SET parameters, then click *Next*.

IMPORTANT: The Policy/Package Agent must be running.

---

8 Select all of the commands you want to configure in the policy.

You can select whole categories by selecting the check box for the category, or clicking the plus sign to expand a SET command category and selecting the check boxes for individual commands to be included.

WARNING: Do not select the Set Developer Option SET command and change the default of Off to On. This parameter is meant to help developers debug server abends. It disables some of the operating system checking to prevent certain abends from occurring. Also, if the Set Developer Option is enabled, running NCP™ scripts that require keyboard entry could abend the server.

---

9 Click *Finish* when you are finished selecting the commands.

The selected commands are now displayed in the Set Commands tab for the policy.

10 To edit a SET command, click its plus sign to expand its attributes.

11 To edit an attribute, select the attribute, then click *Edit*.

A dialog box is displayed in which you can make changes to the attribute.

12 Repeat Step 11 for each attribute to edit for a given SET command.

13 Repeat Step 10 through Step 12 to edit another SET command's attributes.

14 Schedule the policy (see Section 4.7, "Scheduling Policies," on page 239).

15 Click *OK* to close the policy.

If you click *Cancel*, neither the schedule or the SET parameter changes are saved.

Prohibited File

This policy allows you to monitor and enforce the deletion or moving of unauthorized files from a specified volume/drive or directory/folder. For example, you can automate deletion of `.jpg`, `.mp3`, and `.avi` files from a server.

All platforms are supported (NetWare®, Windows, Linux, and Solaris), including the use of the General tab.

With this policy, you can:

- Specify one or more volumes/drives or directories to monitor. You have the option to include all subdirectories.
- Specify which file types to monitor using wildcard combinations.
- Specify the action for all encountered files as follows:
    - Delete
    - Move to specified location
- Specify a schedule for enforcement of the policy.

To configure a policy to manage prohibited files:

1 In ConsoleOne, right-click the *Distributed Server Package*, then click *Properties*.

2 Click *Policies > NetWare* (or other platform).

3 Click *Add*, then select *Prohibited File*.

4 Provide a unique name for the policy, then click *OK*.

The following property page is displayed:
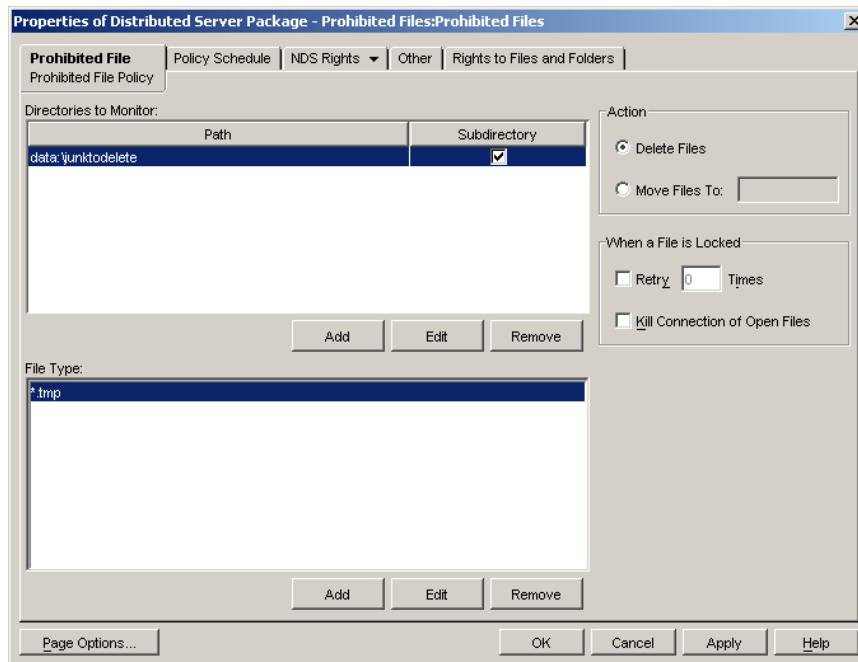
5 Fill in the fields:

Directories to monitor: For this instance of the policy, you can specify the paths to be monitored:

- ◆ **Path:** This can be a volume, drive, or directory name. It must be the full path when a directory is given.

  You can add multiple paths. For each path that you enter, files matching the file types that you define in the File Type field are either deleted or moved according to which Action button you select.

  Variables are supported in the paths.

- ◆ **Subdirectory:** Select the check box to specify that all subdirectories be included.

  If you want only a certain subdirectory, you should create another policy just for that subdirectory by giving its full path in the Path field. However, you cannot move files to a directory that is being monitored, or to any of its subdirectories.

- ◆ **Add:** Opens a dialog box where you can select a path. This field cannot be browsed, so you must know the full path to the files to be moved or deleted.

- ◆ **Edit:** Allows you to edit the selected path.

- ◆ **Remove:** Removes the selected path entry from the list.

Files to manage: You can specify the type of files you want to monitor:

- ◆ **Add:** Opens a dialog box where you specify a file type. You can use wildcards in the path:

  * = any number of characters
  ? = any single character in that position
  ??? = any multiple characters in those positions

This field cannot be browsed, so you must specify the correct information to identify the files to be moved or deleted.

IMPORTANT: The ? wildcard acts differently in ZENworks than in DOS. For example, the search string *.htm? finds only files that end in `.html`, whereas DOS finds files that end in both `.htm` and `.html`. In other words, use of the ? wildcard in ZENworks means that you expect a character to occupy its position in the filename.

- ◆ **Edit:** Allows you to edit the selected file type.
- ◆ **Remove:** Removes the selected file type from the list.

Action: You have two options for how to handle the files you've specified in the Directories to Monitor and the Files to Manage boxes:

- ◆ **Delete files:** Select the option to delete the specified files from the locations you have identified.
- ◆ **Move files to:** Select the option to move the specified files to the path that you specify in this field. This field cannot be browsed, so you must know the full path to where you want the files to be moved.

  If you move files:

  - ◆ The full paths of the files are preserved (meaning if the path doesn't exist at the target, it is created there)
  - ◆ Files are overwritten if they exist in the same path
  - ◆ File or directory attributes and trustees are not transferred
  - ◆ File ownership is preserved

IMPORTANT: If a directory is being monitored, you cannot move files into it or any of its subdirectories.

When a file is locked: Occasionally, files you might be trying to delete or move might be open. For these files, you can specify one of the following resolutions:

- ◆ **Retry ___ times:** Select the check box and enter a number for how many times you want to retry deleting or moving the file before continuing with the next file. Valid entries are from 1 to 10. The time used by each increment depends on the various hardware and software speeds involved in your system.

  Use this field to allow enough time for a temporarily opened file to be closed, such as a file that is only opened long enough for the application to either obtain a copy for editing or write a new copy of the file.

- ◆ **Kill connection of open files:** (NetWare only) Kills the connection that is holding the file open so that the file can be deleted or moved, even if opened by a user at the time.

IMPORTANT: You can only kill connections to files on workstations. Server files cannot be disconnected from the process that has them open.

---

6 Click *OK* to close the policy.

Scheduled Down

You can automate when and how you want a server to go down, and whether it should be automatically brought back up.

To configure a scheduled downing for a server:

1 In ConsoleOne, right-click Distributed Server Package, then click *Properties*.

2 Click *Policies > NetWare* (or other platform).

3 Click *Add*, then select *Scheduled Down*.

4 Provide a unique name for the policy, then click *OK*.

Because the policies selected from this dialog box are plural, you can have multiple Scheduled Down policies listed on the *Policies* tab. Therefore, provide a unique name for this policy.

When you click *OK* after naming the Scheduled Down policy, the policy is selected on the *Policies* tab.

5 Click *Properties*.

The *Up Procedure* tab displays.

6 Select the downing method:

Downing Option	Description
Reset Server	Downs the server and then does a warm boot
Restart Server	Downs the server and then restarts it
Down Server	Downs the server, does not restart it

7 Schedule the policy (see).

8 Click *OK* to close the policy.

If you click *Cancel*, neither schedule for your newly scheduled Down policy is saved.

Scheduled Load/Unload

You can automate scheduled loading and unloading of NLM files and Java Class processes, and Linux and Solaris executables.

To configure the schedules:

1 In ConsoleOne, select the Distributed Server Package's container, right-click the *Distributed Server Package*, then click *Properties*.

2 Click *Policies > NetWare* (or other platform).

3 Click *Add*, then select *Scheduled Load/Unload*.

4 Provide a name for this Load/Unload policy, then click *OK*.

Because the policies selected from this dialog box are plural, you can have multiple Load/Unload policies listed on the *Policies* tab. Therefore, provide a unique name for this policy.

When you click *OK* after naming the Load/Unload policy, it is selected on the *Policies* tab.

5 Click *Properties*.

The *Scheduled Load/Unload* tab displays.

6 Click *Add*.

7 Select one of the following options:

Select an item for further instructions on configuring it.

8 Repeat Step 6 and Step 7 for each NLM or process to be included.

9 To rearrange the order, use the arrow keys.

10 Schedule the policy (see Section 4.7, "Scheduling Policies," on page 239).

11 Click *OK* to close the policy.

If you click *Cancel*, your newly scheduled Load/Unload policy is not saved.

Server Down Process

You can automate the procedures your servers use when they are downed.

IMPORTANT: For the Windows, Linux, and Solaris platforms, if you down the server from its console, this policy is not recognized. Instead, you must down the server using the *Actions* option in *Remote Web Console* in iManager so that this policy can be applied.

To configure the downing process for a server:

1 In ConsoleOne, select the Distributed Server Package's container, right-click the *Distributed Server Package*, then click *Properties*.

2 Click *Policies* > *NetWare* (or other platform).

3 Select the Server Down Process policy, then click *Properties*.

If the box under the *Enabled* column is not selected for the Server Down Process policy, select it before clicking *Properties*. A policy must be enabled to activate the *Properties* button.

4 To configure procedures for downing, select the *Down Procedure* tab, then click *Down Procedures*.

5 To enable the policy's options, select the check box labeled *Follow this procedure when a down server is triggered*, then enter the number of minutes to wait before downing the server.

6 To disable login before downing, select the check box, then enter the number of minutes before downing to disable login.

7 To drop connections before downing, select the check box, then enter the number of minutes before downing the server to drop connections.

8 To configure an order for unloading, select the *Down Procedure* tab, then click *Ordered Unload*.

  8a To include NLM files and processes, select the *Unload these NLMs and kill these processes in this order before downing* check box.

  8b Click *Add*.

  8c Select either *NLM* or *Process*, provide the name, then click *OK*.

8d To change the order, use the arrow keys.

9 To configure reporting, select the *Notification* tab, then click *Reporting*.

    9a To have another server send an SNMP alert if the server is not up after a specified time, select the *Send SNMP Alert* check box, then enter the number of minutes.

    For information about displaying SNMP traps on your management console, see Section 4.3.1, "Compiling Zentrap.mib," on page 210.

    9b To specify which servers can watch for the restart and send the alert in case of failure, click *Add* to display an ordered list of candidate servers.

    Policy and Distribution Services starts at the top of the list to communicate with the first server and use it for the alert notification. If Policy and Distribution Services cannot communicate with a server, the next one on the list is tried. The first server that can be used is the one that is scheduled to send the alert.

    9c Browse to select a server.

    9d Repeat Step 9a through Step 9c for each server needed.

    9e To change the order, use the arrow keys.

10 To configure broadcast messages, select the *Notification* tab, click *Broadcast Messages*, then click *Send messages to connected users*.

    10a Enter the number of times to send the message.

    10b To broadcast custom text, enter it in the box.

    10c To include the predefined message containing a time as the last line of your broadcast, select the check box.

    The *x* minutes is derived from dividing the number of times from Step 10a into the number of minutes remaining before the server can be downed, then subtracting that amount (in whole minutes) for the amount to display in each broadcast. For example, if there are 10 minutes remaining and you select 5 in Step 10a, the message is broadcast every two minutes. The number of minutes remaining after each broadcast will be two minutes less than at the last broadcast.

11 To configure targeted messages, select the *Notification* tab, click *Targeted Messages*, then click *Send e-mail to selected users when server is going down*.

    11a To specify the users, groups, or e-mail addresses to receive the targeted messages, click *Add*.

    11b Select either *User*, *Group*, or *E-Mail Address*.

    11c Browse to select the user or group, or provide the e-mail address.

    11d Repeat Step 11a through Step 11c for other users, groups, or e-mail addresses.

12 To configure the conditions for downing a server, select the *Conditions* tab, then click *Use Conditions*.

    12a To specify the conditions, click *Add*.

    12b Select from the following conditions to specify when not to bring the server down:

    Some of these conditions require you to enter valid names. Others use the Select Object dialog box to browse for them.

    File open: If the files that you specified are open. For example, a `.exe`.

    NLM loaded: If the NLM files that you specified are running.

    Server connected: If the server that you specified is connected.

> **User connected:** If the users that you specified are connected.
>
> **Number of user connections:** If the number of users connected exceeds the number you specify. In other words, don't bring the server down if too many users would be affected.
>
> **Workstation connected:** If the workstations that you specified are connected.

12c Repeat Step 12a and Step 12b for each condition to add to the list.

12d To change the order, use the arrow keys.

13 Click *OK* to close the policy.

If you click *Cancel*, none of the Server Down Process policy changes made on any of the tabs are saved.

Server Scripts

You can automate script usage by your NetWare servers.

To configure server scripts:

1 In ConsoleOne, select the Distributed Server Package's container, right-click the *Distributed Server Package*, then click *Properties*.

2 Click *Policies > NetWare* (or other platform).

3 Click *Add*, then select *Server Scripts*.

4 Provide a unique name for the policy.

Because the policies selected from this dialog box are plural, you can have multiple Script policies listed on the *Policies* tab. Therefore, provide a unique name for this policy.

When you click *OK* after naming the Script policy, it is selected on the *Policies* tab.

5 Click *Properties*.

The *Script* tab displays.

6 Click *Add*, then select *Server Scripts*.

7 Provide a script name.

Script #1 displays.

8 Select the script type (NCF, NetBasic*, PERL).

> **IMPORTANT:** NetBasic is not supported on NetWare 6.5 servers.

9 Enter the script text.

10 Repeat Step 6 through Step 9 for each script to be added.

11 Use the arrow keys to arrange the order to execute the scripts.

12 Schedule the policy (see Section 4.7, "Scheduling Policies," on page 239).

13 Click *OK* to close the policy.

If you click *Cancel*, neither the schedule or any of the scripts entered are saved.

SMTP Host

You can set the TCP/IP address of the relay host that processes outbound Internet e-mail.

To configure the SMTP Host policy:

1 In ConsoleOne, right-click the *Service Location Package*, then click *Properties*.

2 Select the SMTP Host policy, then click *Properties*.

   If the box under the *Enabled* column is not selected for the SMTP Host policy, select it before clicking *Properties*. A policy must be enabled to activate the *Properties* button.

   The *SMTP Host* tab defaults.

3 Provide the TCP/IP address or DNS name (such as `mail.novell.com`), then click *OK* to close the policy.

   If you click *Cancel*, the TCP/IP address is not saved.

SNMP Community Strings

This policy provides configuration and scheduling of SNMP community strings.

Make sure that you have compiled `zentrap.mib` (see Section 4.3.1, "Compiling Zentrap.mib," on page 210).

IMPORTANT: Running INETCFG does not show that the policy has been applied to the server. Instead, use TCPCON to verify. See "Verifying Community String Changes" on page 230.

To configure the SNMP Community Strings policy:

1 In ConsoleOne, select the Distributed Server Package's container, right-click the *Distributed Server Package*, then click *Properties*.

2 Click *Policies* > *NetWare* (or other platform).

3 Select the SNMP Community Strings policy, then click *Properties*.

   If the box under the *Enabled* column is not selected for the SNMP Community Strings policy, select it before clicking *Properties*. A policy must be enabled to activate the *Properties* button.

   The *SNMP Community Policy* tab displays.

4 Fill in the *Community Strings* fields:

   Monitor
   Control
   Trap

   Community strings are case sensitive. Enter a string for each field as needed.

5 Select the *Schedule* tab, then schedule the policy (see Section 4.7, "Scheduling Policies," on page 239).

6 Click *OK* to close the policy.

Verifying Community String Changes

To confirm that the SNMP Community Strings policy has been successfully applied to a server, do the following on any NetWare server:

1 At the server's main command prompt, enter `tcpcon` to display the following menu:

```
TCP/IP Console  6.00k                          NetWare Loadable Module

  Host:    Local System
  Uptime: 0 Days  0 Hours  9 Minutes 42 Seconds
  System: Novell NetWare 5.60.04  December 12, 2003

  IP Received:  915            TCP Received:   9,222
  IP Sent:      391            TCP Sent:       9,250
  IP Forwarded: DISABLED       TCP Connections: 52

                              Available Options

                       SNMP Access Configuration
                       Protocol Information
                       IP Routing Table
                       Statistics
                       Interfaces
                       Display Local Traps

  View and change the TCPCON options.
  ENTER=Select ESC=Exit Menu                              F1=Help
```

2 Select *SNMP Access Configuration* to display "Local System" in the *Transport Protocol* field:

```
TCP/IP Console  6.00k                          NetWare Loadable Module

  Host:    Local System
  Uptime: 0 Days  0 Hours 11 Minutes 33 Seconds
  System: Novell NetWare 5.60.04  December 12, 2003

  IP Received:  1,108          TCP Received:   10,108
  IP Sent:      512            TCP Sent:       10,139
  IP Forwarded: DISABLED       TCP Connections: 51

                         Available Options

                      SNMP Access Configuration

  Transport Protocol: Local System
  Host:
  Community Name:     public
  Timeout:            5    (seconds)
  Poll Interval:      1    (seconds)

  The transport protocol for remote SNMP access.
  ENTER=Select ESC=Previous Menu                          F1=Help
```

3 Press Enter to display the *Transport* options:

```
┌─────────────────────────────────────────────────────────────────────┐
│ TCP/IP Console   6.00k                        NetWare Loadable Module │
│ ┌───────────────────────────────────────────────────────────────┐    │
│ │ Host:    Local System                                          │    │
│ │ Uptime: 0 Days   0 Hours 11 Minutes 59 Seconds                 │    │
│ │ System: Novell NetWare 5.60.04  December 12, 2003              │    │
│ │                                                                │    │
│ │ IP Received: 1,159          │ TCP Received:    10,276          │    │
│ │ IP Sent:       543          │ TCP Sent:        10,308          │    │
│ │ IP Forwarded: DISABLED      │ TCP Connections: 51              │    │
│ └────────────────────────┌─────────────────────┐─────────────────    │
│                          │ Transport           │                      │
│                          ├─────────────────────┤ s                    │
│              SNMP Acc    │ Local System        │ n                    │
│ ┌───────────────────────│ TCP/IP              │─────────────────┐    │
│ │ Transport Protocol: Local Syst │ IPX         │                │    │
│ │ Host:                 └─────────────────────┘                 │    │
│ │ Community Name:     public                                     │    │
│ │ Timeout:            5    (seconds)                             │    │
│ │ Poll Interval:      1    (seconds)                             │    │
│ └───────────────────────────────────────────────────────────────┘    │
│ The transport protocol for remote SNMP access.                        │
│ ENTER=Select ESC=Previous Menu                          F1=Help       │
└─────────────────────────────────────────────────────────────────────┘
```

4 Select the *TCP/IP* option to display the TCP/IP transport protocol information:

```
┌─────────────────────────────────────────────────────────────────────┐
│ TCP/IP Console   6.00k                        NetWare Loadable Module │
│ ┌───────────────────────────────────────────────────────────────┐    │
│ │ Host:    Local System                                          │    │
│ │ Uptime: 0 Days   0 Hours 12 Minutes 17 Seconds                 │    │
│ │ System: Novell NetWare 5.60.04  December 12, 2003              │    │
│ │                                                                │    │
│ │ IP Received: 1,237          │ TCP Received:    10,437          │    │
│ │ IP Sent:       563          │ TCP Sent:        10,469          │    │
│ │ IP Forwarded: DISABLED      │ TCP Connections: 53              │    │
│ └───────────────────┌──────────────────────┐──────────────────────    │
│                     │  Available Options    │                         │
│ ┌───────────────────┴──────────────────────┴──────────────────────┐  │
│ │               SNMP Access Configuration                          │  │
│ │ Transport Protocol: TCP/IP                                       │  │
│ │ Host:               lb                                           │  │
│ │ Community Name:     public                                       │  │
│ │ Timeout:            5    (seconds)                               │  │
│ │ Poll Interval:      5    (seconds)                               │  │
│ └──────────────────────────────────────────────────────────────────┘  │
│ The name or address of the host. DEL=Local System INS=Display host names.│
│ ENTER=Select ESC=Previous Menu                          F1=Help       │
└─────────────────────────────────────────────────────────────────────┘
```

5 Replace `lb` with the IP address of the NetWare server where you want to verify the string changes, and replace `public` with a valid monitor read string:

```
TCP/IP Console  6.00k                              NetWare Loadable Module

    Host:    Local System
    Uptime: 0 Days  0 Hours 13 Minutes 44 Seconds
    System: Novell NetWare 5.60.04  December 12, 2003

    IP Received:  1,430              TCP Received:    11,118
    IP Sent:        690              TCP Sent:        11,157
    IP Forwarded: DISABLED           TCP Connections: 51

                          Available Options

                      SNMP Access Configuration

Transport Protocol: TCP/IP
Host:               192.68.1.203█
Community Name:     myreadstring
Timeout:            5█  (seconds)
Poll Interval:      5   (seconds)

The timeout interval for the request reply.
ENTER=Select ESC=Previous Menu                              F1=Help
```

6 Press Esc to display the *Save TCP/IP Console Option?* menu, then select *Yes* to continue:

```
TCP/IP Console  6.00k                              NetWare Loadable Module

    Host:    Local System
    Uptime: 0 Days  0 Hours 14 Minutes 14 Seconds
    System: Novell NetWare 5.60.04  December 12, 2003

    IP Received:  1,475              TCP Received:    11,298
    IP Sent:        722              TCP Sent:        11,337
    IP Forwarded: DISABLED           TCP Connections: 51

                          Available Options

                      SNMP Access Configuration

Transport Protocol
Host:               Save TCP/IP Console Options?
Community Name:
Timeout:            No
Poll Interval:      Yes

The timeout interval for the request reply.
ENTER=Select ESC=Previous Menu                              F1=Help
```

7 At this point, you should see the statistics being updated; however, if the community string changes are not displayed (as depicted below), make sure that the correct monitor string was entered in Step 5.

```
TCP/IP Console  6.00k                        NetWare Loadable Module

    Host:    192.68.1.203
    Uptime:  Unknown
    System:  Unknown

    IP Received:   0            TCP Received:    0
    IP Sent:       0            TCP Sent:        0
    IP Forwarded: Unknown       TCP Connections: 0

                        Available Options
                    SNMP Access Configuration
                    Protocol Information
                    IP Routing Table
                    Statistics
                    Interfaces
                    Display Local Traps

    View and change the TCPCON options.
    ENTER=Select ESC=Exit Menu        _                    F1=Help
```

8 Another way to see that the policy is actually applied when the policy is deployed is to change the messaging level for the server's Subscriber object to Level 4 or Level 5 (see the *SNMP trap* field in Step 3 under Section 3.6.3, "Configuring Subscribers," on page 151), then view the new and old string values in the TCP/IP Console screen as the changes occur.

SNMP Trap Targets

You can set targets for SNMP traps for the Policy/Package Agent.

- "Understanding How the Windows Trap Target Policy Enforcer Behaves" on page 233
- "Configuring the SNMP Trap Target Policy" on page 234

For information about displaying SNMP traps on your management console, see Section 4.3.1, "Compiling Zentrap.mib," on page 210.

Understanding How the Windows Trap Target Policy Enforcer Behaves

The following abbreviations are used in this section to represent these Windows registry locations:

- **AGENT_KEY:**

  HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\SNMP\Parameters

- **ZFS_KEY:** HKEY_LOCAL_MACHINE\SOFTWARE\Novell\Zenworks\ZfS

The Windows SNMP trap target policy enforcer performs in the following sequence:

1. The policy enforcer first verifies an installation of an SNMP agent. This is done by checking if AGENT_KEY exists. If it exists, the enforcer assumes that an SNMP agent is installed and continues with the following steps. Otherwise, an error is returned and the processing stops.

2. The enforcer keeps track of all trap targets added by the ZENworks Server Management policy by placing the trap targets in ZFS_KEY. The trap targets are organized like the trap targets in AGENT_KEY with a subkey of TrapConfiguration. The subkey TrapConfiguration contains community strings that are represented as registry subkeys. These community strings contain the trap target values associated with each community string.

3. Each trap target in the ZENworks Server Management policy is put into AGENT_KEY, unless it already exists. The policy enforcer ensures that each Server Management trap target is found, or is added to each community string. If no community strings exist in AGENT_KEY, a community string named "public" is created.

4. Any previously added trap targets found in ZFS_KEY that are removed from the ZENworks Server Management policy are removed from AGENT_KEY. Trap targets not added by Server Management are not removed.

5. If Microsoft's SNMP agent is installed, the agent's trap targets are automatically updated with registry changes.

Configuring the SNMP Trap Target Policy

To configure the SNMP Trap Targets policy:

1 In ConsoleOne, right-click the Service Location Package, then click *Properties*.

2 Select the SNMP Trap Targets policy, then click *Properties*.

If the box under the *Enabled* column is not selected for the SNMP Trap Targets policy, select it before clicking *Properties*. A policy must be enabled to activate the *Properties* button.

3 Click *Add*.

4 Provide a new target, then click *OK*.

TIP: Provide the TCP/IP address or DNS name of the target server. IPX addresses are not supported.

---

5 Repeat Step 3 through Step 4 for each new trap target.

6 Select the *Schedule* tab, then schedule the policy (see Section 4.7, "Scheduling Policies," on page 239).

7 Click *OK* to close the policy.

If you click *Cancel*, none of the targets that you provided are saved.

Text File Changes

You can automate changes to text files on your servers.

To configure text file changes:

1 In ConsoleOne, select the Distributed Server Package's container, right-click the *Distributed Server Package*, then click *Properties*.

2 Click *Policies > NetWare* (or other platform).

3 Click *Add*, then select *Text File Changes*.

4 Provide a unique name for the policy.

Because the policies selected from this dialog box are plural, you can have multiple text file policies listed on the Policies tab. Therefore, provide a unique name for this policy.

When you click *OK* after naming the text file policy, it is selected on the *Policies* tab.

5 Click *Properties*.

The *Text Files* tab defaults.

6 Click *Add*.

After one text file has been added, you are given the opportunity to select whether you are adding another text file or another change item for the selected text file.

To add another text file, select *Text File*. It does not matter which text file or change item is selected in the left pane—the text file is added to the far left level.

To add another change to a text file, in the left pane select the text file for the change, click *Add*, then select *Change*. The change item is added under the selected text file.

7 If you are adding a text file, provide the name of the text file.

8 Accept the default name (such as Change #1) or rename it; if you are adding a text file, click *OK*.

9 Click the down-arrow for the *Change Mode* field, then select the change mode from the drop-down list.

10 Click the down-arrow for the *Search Type* field, then select the search type from the drop-down list.

11 Enter the exact search string.

12 Select the check box if you want the string search to be case sensitive.

13 To find all occurrences of the search string, make sure the box is selected, or deselect the box to find only the first occurrence.

14 Click the down-arrow for the *Result Action* field, then select the action from the drop-down list that should result if a string is matched.

15 If you are replacing a string or entering a new one, enter the text in the *New String* text box.

16 Repeat Step 6 through Step 15 for each text file to add or each change to be made.

17 To reorder the text files and change items, use the arrow keys.

18 Schedule the policy (see Section 4.7, "Scheduling Policies," on page 239).

19 Click *OK* to close the policy.

If you click *Cancel*, neither the schedule or any of the text files entered are saved.

ZENworks Database

If you installed the Server Management database during installation, but the database file is not associated with a Database object, you can set its object's DN so that the server this policy is associated with can find the database file for logging information.

To configure the ZENworks Database policy:

1 In ConsoleOne, right-click the *Service Location Package*, then click *Properties*.

2 Select the ZENworks Database policy, then click *Properties*.

If the box under the *Enabled* column is not selected for the ZENworks Database policy, select it before clicking *Properties*. A policy must be enabled to activate the *Properties* button.

3 Select the *Policy/Distribution Management* tab.

The *Inventory Management* tab defaults. Make sure you are using the correct tab.

4 Provide the DN of your ZENworks Database object, or browse to select the DN, then click *OK* to close the policy.

If you click *Cancel*, the DN is not saved.

ZENworks Server Management

This policy provides basic configuration parameters for Policy and Distribution Services.

To configure the ZENworks Server Management policy:

1 In ConsoleOne, select the Distributed Server Package's container, right-click the *Distributed Server Package,* then click *Properties*.

2 Click *Policies > NetWare* (or other platform).

3 Select the ZENworks Server Management policy, then click *Properties*.

If the box under the *Enabled* column is not selected for the ZENworks Server Management policy, select it before clicking *Properties*. A policy must be enabled to activate the *Properties* button.

The *General – Status* tab displays.

4 To determine the policy's general status:

    4a Select the procedure to follow when displaying messages at the server console.

    4b Select the procedure to follow when sending SNMP traps.

    For information about displaying SNMP traps on your management console, see Section 4.3.1, "Compiling Zentrap.mib," on page 210.

    4c Select the procedure to follow when recording information to a log file.

Logging Procedure	Description

Log File	Select this option to enable it and provide the log file's filename. Include its full path. By default, Policy and Distribution Services uses `\zenworks\zfs-startup.log`, unless you enter a filename here. Then, for the servers where this policy is enforced, the log file you specify here is used instead of `zfs-startup.log`.
	
	 Some examples:
	
	 `sys:\zenworks\polpack.log`
	 `sys:\zenworks\polpack.txt`
	 `data:\zenworks\policies.log`

Delete Log Entries Older Than__Days	Use this option to control disk space usage.
E-Mail Messages	Select whether to send e-mail messages. The None or Errors Only options are recommended.
◆ Users	You can add users, groups, or e-mail addresses.
◆ Address Attribute	After you select users or groups, this field displays the attribute of the associated user or group. You can change the attribute from the drop-down list.

IMPORTANT: Set the E-Mail Messages option to either None or Errors Only. If you set this to a more detailed level, performance degrades because of the extra e-mail messages that are created.

5 To determine the policy's configuration, select the *ZENworks Server Management* tab, then click *Configuration*.

5a Provide a console prompt.

You can customize the prompt using plain text and variables. The default is:

`%SERVER_DN% - ZENworks Server Management >`

You can use any of the predefined or user-defined variables (for more information, see Section 9.2, "Types of Variables," on page 353).

5b Provide a working path.

This is for Policy and Distribution Services temporary and backup files. The default directory is `\zenworks\pds\smanager\working`.

5c To determine how old database information should be before purging, enter the number of days.

All policy-related information older than the number of days entered is purged when Server Management is started on the same server where `zfslog.db` resides.

IMPORTANT: The database can only be purged if Server Management is running on the same server where `zfslog.db` is located.

Tiered Electronic Distribution information is purged manually from the database. For more information, see Section 10.5, "Purging the Database," on page 368.

6 To set a port number for the ZENworks Web Server, select the *Port Configuration* tab and select or enter a port number.

7 Click *OK* to close the policy.

If you click *Cancel*, none of the policy changes on any of the tabs are saved.

4.3.5 Creating Custom Log Files Using Policies

If you want to create custom log files, you can use either the Tiered Electronic Distribution policy (Service Location Package) or the ZENworks Server Management policy (Distributed Server Package):

- **Tiered Electronic Distribution policy:** With this policy, you associate its Service Location Package to an eDirectory container, and all Distributor and Subscriber objects under it can use this policy. The Use Policy check box that is displayed in each of the object's properties allows you to individually select whether that Distributor or Subscriber should use the policy. The check box is disabled by default.

  Using this policy, the Distribution Agent logs Tiered Electronic Distribution information to your custom log file for the selected Distributors and Subscribers.

- **ZENworks Server Management policy:** With this policy, you distribute its Distributed Server Package to the servers where you want the policy enforced.

  Using this policy, the Policy/Package Agents for these servers log policy and software package information to your custom log file.

When you are creating and configuring one of these policies, the Path and Filename field for the log file is blank by default.

For information on how to create and configure these policies, see:

- "Tiered Electronic Distribution" on page 213
- "ZENworks Server Management" on page 236

4.4 Enabling Policies

A policy must be enabled before it is in effect for the policy package. You can disable a policy without removing it from the package.

To enable a policy:

1 In ConsoleOne, right-click the Policy Package object containing the policy to be enabled, then click *Properties*.

2 To enable a policy, select its check box under the *Enabled* column.

If you enable a policy, make sure it is correctly configured.

3 To cause an enabled policy to be enforced, distribute the policy package.

For more information, see Section 4.5, "Distributing Policies," on page 238.

4.5 Distributing Policies

You must distribute a Distributed Server Package before its policies are in effect. When you do distribute the package, its enabled policies are only in effect for the server where it is distributed after the Subscriber has extracted the Distribution.

To distribute policies to a server:

1. Create a Distribution that is a Policy Package type.

2. Configure the policies in the policy package.

3. Select a Channel for the Policy Package Distribution.

4. Subscribe the Subscribers to the selected Channel.

5. Send the Distribution.

The Policy/Package Agent on the receiving server extracts the enabled policies and enforces them on the server.

For more information on creating Policy Package Distributions, see "Creating and Configuring the Distribution" on page 57.

4.6 Associating Policies

Because Distributors do not receive polices through Distributions, the Distributor object needs to be associated with the Container Package object so that it can use the Search policy for how to read the eDirectory tree when the Distributor is refreshed.

The Distributor object also needs to be associated with the Service Location Package. This package contains the ZENworks Database policy, which enables the Distributor Agent to locate the database

file for writing report information. It also contains other policies the Distributor uses (see Section 4.3.3, "Configuring Service Location Package Policies," on page 212).

To associate policy packages with the Distributor object's container:

- Section 4.6.1, "Associating a Policy Package to the Distributor Object," on page 239
- Section 4.6.2, "Associating the Distributor Object to a Policy Package," on page 239

4.6.1 Associating a Policy Package to the Distributor Object

To associate a policy package to the Distributor object's container:

1 In ConsoleOne, right-click the policy package, then click *Properties*.

2 Select the *Associations* tab, then click *Add*.

3 Browse to select the container where the Distributor object resides (or any container above it), then click *OK*.

   If you click *Cancel*, the association you made is not saved.

4.6.2 Associating the Distributor Object to a Policy Package

To associate the Distributor object's container with a policy package:

1 In ConsoleOne, right-click the container where the Distributor object resides (or any container above it), then click *Properties*.

2 Select the *ZENworks* tab, click *Associated Policy Packages*, then click *Add*.

3 Browse to select the policy package, then click *OK*.

   If you click *Cancel*, the association you made is not saved.

4 Repeat Step 2 and Step 3 for additional policy packages to be associated with the Distributor object's container.

4.7 Scheduling Policies

For information, see Section 8.3, "Scheduling and Server Policies," on page 346.

4.8 Viewing Effective Policies

To view which ZENworks 7 Server Management policies are in effect for the current server object:

1 At the `ZENworks Server Management` prompt on the server, enter `Policy List`.

   Displays the policies that are currently in effect for the server.

4.9 Changing Policy Enforcement

You might need to change or stop policy enforcement for a particular server or a group of servers.

You can change policy enforcement in several ways:

- Section 4.9.1, "Modifying a Policy That Is Being Enforced," on page 240

4.9.1 Modifying a Policy That Is Being Enforced

To change a policy that is being enforced:

1 In ConsoleOne, right-click the Distributed Server Package object containing the policy to be modified, then click *Properties*.

2 Modify the policy as needed, then click *OK* to exit the policy package properties.

The next time the Distribution containing this policy package is built, the following transpires:

1. A new version of the Distribution is created because it had changed.

2. The Policy Package Distribution is sent according to the Send schedule of the Channel.

3. The Subscribers subscribed to the Channel each receive and extract the Policy Package Distribution according to their extraction schedules.

4. The modified policy is enforced on the Subscribers where the Policy Package Distribution was extracted.

4.9.2 Stopping a Specific Policy From Being Enforced

To stop a specific policy from being enforced:

1 In ConsoleOne, right-click the Distributed Server Package object containing the policy to be stopped, then click *Properties*.

2 Select the policy to be stopped, then do one of the following:

  2a Select the check box under the Enabled column to disable the policy.

  2b Click *Remove* to remove the plural policy.

    You can delete plural policies from a policy package because they were previously added using the Add button.

3 Click *OK* to save the change and exit the policy package properties.

The next time the Distribution containing this policy package is built, the following transpires:

1. A new version of the Distribution is created because it had changed.

2. The Policy Package Distribution is sent according to the Send schedule of the Channel.

3. The Subscribers subscribed to the Channel each receive and extract the Policy Package Distribution according to their extraction schedules.

4. The disabled/removed policy is no longer enforced on the Subscribers where the Policy Package Distribution was extracted.

4.9.3 Removing Policy Enforcement for a Specific Subscriber

If you want to stop a distributed policy from being enforced on a specific Subscriber server, rather than on all Subscribers receiving that Distribution, do the following:

1 In ConsoleOne, right-click the Subscriber object, then click *Properties*.

2 Select the *Channels* tab, select the Channel containing the policy to be removed from enforcement, click *Remove*, then click *OK*.

3 Click *OK* to close the Subscriber object's properties.

4 On the Subscriber server's file system, delete the following files:

   ◆ The Distribution directory containing the policy's Distribution file

   ◆ The related Policy file (`.pol`) from the `\smanager\policy` directory (which was created when the Policy Package Distribution was extracted)

5 Reset the Subscriber server to refresh its policy configuration.

The Subscriber no longer receives the Policy Package Distribution containing that policy, nor does it continue to enforce the policy previously distributed to the Subscriber.

4.9.4 Stopping Enforcement of a Policy Package Distribution

If you need to stop enforcement of a Policy Package Distribution for all of the Subscribers where it was distributed, you must follow certain steps. Because the policy package was distributed, each Subscriber that received the Distribution can still enforce that policy if you only delete the policy package object.

To stop enforcement, do the following:

1 In ConsoleOne, delete the Distribution object for the Policy Package type.

   IMPORTANT: If the policy package has other policies that you do not want to stop, then do not delete the package. Instead, just disable the policy that you want to stop.

2 On the Subscriber server's file system, delete the `.pol` file that was created by the Policy Package Distribution.

   The `.pol` file is located under the `\zenworks\pds\smanager\policies` directory.

3 Refresh the policies on each Subscriber.

   You can do this from each Subscriber server's console using the Policy Refresh command, or from iManager using the Refresh option.

The policies in the Policy Package Distribution are no longer enforced on the Subscriber after its policies have been refreshed. The refresh process clears its memory of all policies, then reloads them from the Policy Package Distributions existing in its file system.

Server Software Packages

<div style="text-align: right; font-size: 3em;">5</div>

Novell® ZENworks® Server Management provides the Server Software Packages component for managing files and applications on your network. Using software packages, you can automate the installation and upgrading of software on your servers.

The real value in using software packages is to set up processes to be done on a server before and after installation of the package.

The following sections give you an understanding of how you can benefit from using the Server Software Packages component:

5.1 Software Management through Server Software Packages

Software management is done by creating Server Software Packages and distributing them using Tiered Electronic Distribution. You can configure Server Software Packages so that a server must meet certain minimum requirements before a package is installed on it. Software packages can consist of multiple software package components.

Each software package component can also be configured so that minimum requirements must be met before that component can be installed on the server.

5.2 Understanding Server Software Packages

Policy and Distribution Services provides the means to automate and standardize the distribution and installation of server files and applications. This includes your ability to standardize NLM™ versions, configuration files, databases, and more. Review the following sections:

5.2.1 Understanding Server Software Packages and Components

To distribute server files and applications for installation on a server, you must include the software in a software package. You create the software packages under the Server Software Packages namespace in ConsoleOne®. Creating software packages is like building a software installation executable.

Figure 5-1 illustrates the relationship between software packages and package components:

Figure 5-1   *Server Software Packages Sample Tree*
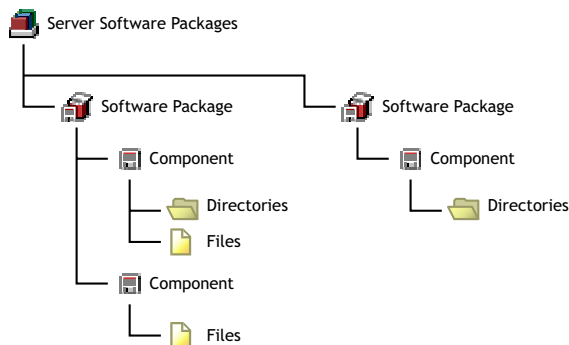
Note the following:

- Software Package objects are displayed under the Server Software Packages namespace
- A Software Package object can contain multiple Component objects
- Component objects can contain files and directories
- Each software package can include all of the files for one or several applications
- Software Package configuration files (`.spk` and `.cpk`) are stored on a server or workstation file system

5.2.2 Understanding Software Package and Component Configurations

Software packages and their components contain configuration information and installation requirements. Because each Component object is governed by its own set of configuration parameters and installation requirements, you might have multiple components for a software package, such as pre-installation actions, installation actions, and post-installation actions.

You can configure every aspect of the distribution and installation of server files and applications, including the following:

- Requiring a specific operating system
- Specifying how much RAM the target server needs
- Specifying how much disk space the target server needs

- Requiring certain SET commands on the target server
- Making changes to the target server's registry
- Replacing files on the target server
- Requiring specific `products.dat` entries

Requiring Software Package Installation Prerequisites

Not only can a software package have installation prerequisites, but each of its components can also have its own installation prerequisites. The hierarchy for adhering to prerequisites to determine installation eligibility is:

- If the prerequisites for the package are not met, none of the components are installed.
- If the prerequisites for the package are met, the components are eligible to be installed.
- If the prerequisites for a component are not met, that component is not installed.

Because some components can be installed while others are not, a partial installation of the software package is possible.

IMPORTANT: When you specify prerequisites, be sure to create prerequisites at the software package level that would apply equally to all of its components, and create prerequisites at the component level that are specific to that component.

Naming Software Packages

When you create a software package, you initially give it a `.spk` extension, which represents a software package that has not yet been compiled. This file contains all of the installation requirements for the software package and all of its components.

WARNING: Do not use double-byte characters in the software package name. This causes an error in any report you run on the software package.

5.2.3 Determining the Installation Order of Software Packages

There are two issues concerning the ordering of Server Software Packages in Distributions:

Forcing the Software Package Distribution Order Using Multiple Distributions

If you want to include multiple software packages in one Distribution, consider the following:

- Multiple software packages are not gathered into a Distribution in any particular order when the Distribution is built
- Multiple software packages are not applied to a server in any guaranteed order when the Distribution is extracted and installed
- Multiple software packages that are contained in one Distribution and start their installations in a certain order might not all finish in that same order

To install software packages in a particular order:

1 Place each software package in its own Distribution (one software package per Distribution).

2 Control the order of software package installations by scheduling the order when the Distributions are sent and extracted.

Forcing the Software Package Distribution Order Using Dependencies

Another way to ensure software package installation order is to use dependencies with multiple Software Package Distributions, such as placing a dependency in a subsequent software package that is established in previous software package.

For example:

1. Create Software Package Distribution 1.

2. Create Software Package Distribution 2 with a dependency on something that is installed from Software Package Distribution 1.

3. Send both Distributions.

4. If the Subscriber attempts to extract Software Package Distribution 2 first, it will fail.

5. The Subscriber extracts Software Package Distribution 1, which provides the dependency on the Subscriber required by Software Package Distribution 2.

6. The Subscriber can now successfully extract Software Package Distribution 2.

With this scenario, you do not need to use schedules to control the installation order.

How Rollback Is Affected by Software Package Ordering

Rollback is also affected by the fact that multiple software packages contained in one Distribution won't necessarily finish extracting in the same order that they started.

Although you can specify the order for processing software packages that are contained in a Distribution, this order is not guaranteed. This is because the length of time it takes for software packages to finish processing can be different for each package, and it is the finishing time for a software package that determines its rollback order.

In other words, you can only roll back the last software package that was successfully processed, and then other software packages only in the reverse order of when they finished processing.

You can use the Package List command to view the order in which software packages finished processing. An asterisk marks the next package that is available for rollback.

For more information on rollback, see Section 5.2.11, "Rolling Back Software Package Installations," on page 255.

5.2.4 Executing Extracted Files

In a Software Package Distribution, some of the files that the software package copies to a server might be executables that you want to have execute in conjunction with extracting the Software Package Distribution.

To run executable files that are delivered through a software package, configure the pre or post execution actions, including order of execution, of the files in the software package. Pre and post

actions are available when creating the Server Software Package and when creating the Software Package Distribution.

For more information, see Section 3.4.6, "Pre and Post Processing for Distributions," on page 127, and "Configuring the Software Package Components" on page 263.

5.2.5 Compiling Software Packages

After you have defined your software packages, including configuring the components, you must compile the software package. This process compresses the files and applications and their configurations into one file for distribution.

The default extension for a compiled software package is `.cpk`. The compiled version contains all of the files necessary to install the files and applications that the software package represents.

IMPORTANT: If you provide the path and filename of the `.spk` when you are prompted for the compiled filename, the `.spk` is overwritten and can no longer be edited. Be sure to use the `.cpk` extension when naming the compiled version.

A `.cpk` file has the potential to be very large (hundreds of megabytes), because software packages can include many large files to be copied. Therefore, `.cpk` files should generally be stored on a server where you have sufficient free disk space.

However, software packages can perform simple functions, which would make the `.cpk` files' sizes relatively small, so that you could store them on a workstation. For example, a software package could be configured to just delete directories on a file server (see Section 5.5, "Using Server Software Packages to Delete Directories on Servers," on page 274).

When a rollback-enabled software package is successfully installed, a rollback package is created on the server. Processing this rollback package returns the server to its original state (before the package was installed). For more information, see Section 5.2.11, "Rolling Back Software Package Installations," on page 255.

5.2.6 Accessing Software Packages

Where you save software packages (on workstations or on servers) depends on how you want to manage the software packages.

Because the Server Software Packages component uses a namespace in ConsoleOne, it enables you to have access to software packages from any workstation or server where you are running ConsoleOne.

However, you should be aware of the following issues:

- "Running ConsoleOne from a Workstation" on page 248
- "Running ConsoleOne from a Server" on page 248

For information on managing software packages from multiple workstations, see Section 5.2.9, "Managing Server Software Packages," on page 249.

Running ConsoleOne from a Workstation

If you run ConsoleOne from a workstation and save a software package to that workstation, the package is not available in ConsoleOne to other workstations or servers running ConsoleOne.

Running ConsoleOne from a Server

You must have the same drive mapping to a server on different workstations if you run ConsoleOne from the server at those workstations. Otherwise, any software package you save to that server cannot be read at the different workstations.

For example, the following scenario illustrates when a package can be found:

1. You run ConsoleOne from Workstation A to access Server A.
2. Server A is mapped as drive S: for Workstation A.
3. You save `pkg_a.spk` to Server A.
4. You run ConsoleOne from Workstation B to access Server A.
5. Server A is also mapped as drive S: for Workstation B.
6. `Pkg_a.spk` can be found because both workstations were mapped to drive S:.

The following scenario illustrates when a package cannot be found:

1. You run ConsoleOne from Workstation A to access Server A.
2. Server A is mapped as drive S: for Workstation A.
3. You save `pkg_a.spk` to Server A.
4. You run ConsoleOne from Workstation B to access Server A.
5. Server A is mapped as drive T: for Workstation B.
6. `Pkg_a.spk` cannot be found because you are looking for the package on drive T: when it was previously saved to drive S:.

The only difference between the scenarios is the drive letter mappings to Server A for each workstation.

5.2.7 Distributing Software Packages

Distributions can include software packages, which are installed, or file groupings, which are extracted.

The Policy/Package Agent extracts or installs Software Package Distributions on the Subscriber server.

When software packages are created, they can contain system requirements that must be met before you install the package on the target Subscriber's server. If the Subscriber meets these requirements, the subscription schedule determines when the package is actually installed.

5.2.8 Distributing Software Packages to a Cluster

When you send a Distribution containing software packages to a cluster to update the sys: volume for each node, the only node in the cluster that receives it is the one that currently has the Subscriber software running.

Because the machines comprising the nodes in the cluster run the Subscriber software, only one node at a time in a cluster is actively running the Subscriber software.

Therefore, if you want to use a Software Package Distribution to update files on a sys: volume for each node in a cluster, you must do this manually by updating one node, bringing it down so that the next node in the failover sequence sees that the previous node has failed and start running the Subscriber software, then update that machine, bring it down, and so on, until all of the machines in the cluster have been updated. Then restart all of the downed servers in the cluster and the primary node's machine takes over again.

You can use a Software Package Distribution to update files on the cluster machine itself, such as Tiered Electronic Distribution .ncf files, because the Subscriber software is contained on the cluster machine's shared hard drive.

5.2.9 Managing Server Software Packages

The following sections explain where to store Server Software Package files, and how to manage them:

Understanding Server Software Package Files

There are three file types associated with software packages:

- **Configuration file (.spk):** When you create a Server Software Package, you initially create a configuration file (.spk) for it. This file's configuration is created in the properties of the software package object in the Server Software Packages namespace in ConsoleOne.

  A .spk file is generally small (around 100 KB). Therefore, it can generally be stored on the workstation running the instance of ConsoleOne that you are using to create and manage software packages.

- **Compiled file (.cpk):** When you compile a software package, a .cpk file is created from the .spk file's configuration information. This provides the content of the software package, such as files or functions. The .cpk file is used to install the software package's content on a server.

  You should generally store .cpk files on a server where there is sufficient free disk space, because compiled software packages might contain many files. However, you can store small .cpk files that only contain functions on a workstation.

- **Preferences file (.ser):** The preferences file (snapinprefs.ser) is automatically created on the workstation being used to create a software package. It contains pointers to the .spk files for the software packages.

  This preferences file allows you to see the software packages in the namespace in ConsoleOne. In other words, software packages are displayed in the Server Software Packages namespace for an instance of ConsoleOne only if the .spk file's path is listed in the preferences file located on the workstation running that instance of ConsoleOne.

When you create a new software package, you specify the local path for the `.spk` file. When you compile a software package, you specify the server's path for the `.cpk` file. After you exit ConsoleOne, any time you have created, deleted, or compiled a software package, the `.spk` file paths are logged to the `snapinprefs.ser` file.

The path to the `.cpk` file is also logged to the `snapinprefs.ser` file. The next time you compile the software package, the wizard displays the `.cpk` file's previous location so that you do not need to remember it each time you compile the package. However, you need to note where you store the `.cpk` files for when you want to distribute them using Tiered Electronic Distribution, because the `.cpk` files' locations are not stored in the software package's properties.

Understanding Your Software Package Management Options

If you are using only one specific workstation for viewing, creating, and managing all of your software package files, then you can store the `.spk` files on that workstation.

It is also possible to manage your software packages from multiple workstations. This requires that you centralize your `.spk` file storage to a network server. This method also requires the use of a master `snapinprefs.ser` file so that you can view all of your software packages from any workstation.

The next two sections explain these management options.

Storing and Managing .Spk Files Using One Workstation

If you use only one workstation for viewing, creating, and managing your software packages, you can store the `.spk` files on the workstation and the `.cpk` files on a server.

Whether you are running ConsoleOne from the workstation where it is installed or from a workstation that uses an installation of ConsoleOne on a network server, the `snapinprefs.ser` file is updated on the workstation being used to run ConsoleOne.

Storing .Spk Files on a Network Server and Managing Them from Multiple Workstations

If you want to use multiple workstations for viewing, creating, and managing the same set of software packages, you need to store all `.spk` files on a network server so that they can be accessed by each workstation.

You might also want to use different workstations for managing different sets of software packages. Any workstation used to create `.spk` files has a software package preferences file of its own created on the workstation used to manage the software packages.

You can manage all of your software packages from multiple workstations if you use a master copy method for the `snapinprefs.ser` file.

Understanding the Software Package Preferences File

When you create a Server Software Package object in ConsoleOne, a software package preferences file (`snapinprefs.ser`) is created in the following location on the workstation running ConsoleOne:

`c:\documents and settings\`*user_ID*`\.consoleone` (Windows 2000)

where *user_ID* is the user directory associated with how you are logged in, such as Administrator.

The full path and filename for a software package is drive-dependent. The `snapinprefs.ser` file contains the drive letter, path, and package name for each `.spk` created by the workstation.

The `snapinprefs.ser` file is unique for each workstation. It is the preferences file that is updated whenever you add or remove `.spk` files using that workstation. Therefore, if you use three different workstations to create `.spk` files, you have three different `snapinprefs.ser` files, each on its own workstation.

When you start ConsoleOne, it checks to see if a `snapinprefs.ser` file was created for that workstation by the instance of ConsoleOne being run on the workstation, and whether ConsoleOne is installed on that workstation or is being run on that workstation from an instance installed on a server. If the file does not exist, a `snapinprefs.ser` file is created when you exit ConsoleOne. If it exists, the `snapinprefs.ser` file is updated with the full paths to any new `.spk` files.

You can copy a `snapinprefs.ser` file from one workstation to another. However, after replacing a `snapinprefs.ser` file with a copy from another workstation, you need to restart ConsoleOne to see any change.

A software package can become unusable if you change drive mappings after creating the package, because the `snapinprefs.ser` file's location to the package is then different. However, if you use a UNC path, this is not an issue as long as the workstation has access to that UNC path.

If you replace the `snapinprefs.ser` file on a workstation, you need to manually insert any software packages missing from the newly copied `snapinprefs.ser` file. Otherwise, the software packages listed in the `snapinprefs.ser` file that was replaced would be inaccessible on the workstation.

Even if a workstation has never been used to create a software package, you can copy a `snapinprefs.ser` file from another workstation to the appropriate location (`c:\....\.consoleone`). Then, when you start ConsoleOne, you can see all of the software packages that are listed in the `snapinprefs.ser` file that you copied.

For more information, see .

Managing Software Packages from Multiple Workstations

If you are using multiple workstations for creating, deleting, and compiling the same set of software package files, you should do the following:

1. Store the `.spk` files on one network server (usually the server where you are storing their corresponding `.cpk` files), so that the software packages can all be accessed from any workstation.

2. When mapping a workstation to the server where the `.spk` and `.cpk` files are stored, use the same drive letter for all workstations.

3. Create a master `snapinprefs.ser` file to use for keeping all workstations updated with their latest software package additions, deletions, and compilations (see "Setting Up the Master Snapinprefs.ser File" on page 257).

4. Create a batch file for starting and stopping ConsoleOne on a workstation (see "Creating and Using the ConsoleOne Batch File" on page 258). This batch file does two things:

   ◆ Automatically upload the latest `snapinprefs.ser` file from the storage server to the workstation any time ConsoleOne is started on that workstation.

     This allows you to see all software packages from the workstation where you started ConsoleOne.

   ◆ Automatically download the revised `snapinprefs.ser` file from the workstation to the storage server when ConsoleOne is exited on that workstation.

     This creates a new master copy of the `.ser` file containing the workstation's latest software package additions.

5. Run the batch file from any workstation where you want to manage software packages (see "Using the ConsoleOne Batch File" on page 260).

General Rules for Managing Software Packages from Multiple Workstations

Using a master copy for the `snapinprefs.ser` file works only if you exit ConsoleOne on one workstation, then start it on another workstation. This sequential method does not work for concurrently running instances of ConsoleOne where each instance is updating its local `snapinprefs.ser` file. The instance of ConsoleOne that is exited last overwrites the master copy with its local `.ser` file.

IMPORTANT: Creating, deleting, or compiling software packages in ConsoleOne are the only functions that cause logging to the `snapinprefs.ser` file. Therefore, you can use ConsoleOne to manage software packages, such as viewing and editing properties, without starting ConsoleOne from the batch file. Just make sure that you do not add, delete, or compile any `.spk` files in ConsoleOne if you do not start ConsoleOne with the batch file.

To manage software packages using this master copy/single server/multiple workstation method, observe the following general rules:

◆ Always exit ConsoleOne after creating a new software package (`.spk` file) or compiling a new package (`.cpk` file). This causes the master `snapinprefs.ser` file to contain the newest software package links.

◆ Never have two or more workstations concurrently managing software packages. The batch file used to start ConsoleOne on these workstations could cause paths to any newly created software packages to be lost.

◆ Never use the batch file to start ConsoleOne when you do not intend to manage software packages. Instead, start ConsoleOne without using the batch file.

  You need to do this because the batch file always overwrites the master copy on the software package storage server when ConsoleOne is exited (if ConsoleOne was started by the batch file). You could inadvertently overwrite the master `snapinprefs.ser` file and lose links to newly created software packages.

  For example, on Workstation_A you run the batch file to start ConsoleOne, do administrative work other than software packages, for some reason go to Workstation_B where you decide to create a new software package (so you use the batch file again), exit ConsoleOne on

Workstation_B, then later exit ConsoleOne on Workstation_A. Your new software packages created on Workstation_B no longer have links to them in the master `snapinprefs.ser` file.

The Best Scenario for Using Multiple Workstations to Manage Software Packages

The best scenario is that you have one administrator who can use multiple workstations to manage your software packages. If you have multiple administrators, they need to coordinate so that they don't overwrite each other's latest software package additions and deletions in the master `snapinprefs.ser` file.

For more information, see .

Example: Using a Master Snapinprefs.ser File

Keeping the master copy on the server properly updated is a matter of timing. For example, in the following scenario, the first `snapinprefs.ser` file was initially created on Workstation A, then copied to the network server to be the master `snapinprefs.ser` file. Both workstations are using Windows 2000.

A batch file is used to start ConsoleOne for the purpose of controlling events before and after using ConsoleOne. For example:

1. Administrator A starts the batch file on Workstation A to begin ConsoleOne.
2. The batch file running on Workstation A identifies the storage server as being mapped to drive M: (or it maps drive M: to that server).
3. The batch file copies the master `snapinprefs.ser` file from the server at drive `M:` to the `c:\documents and settings\`*`user_ID`*`\.consoleone` directory on Workstation A.
4. Administrator A creates a new software package, naming it `ssp1.spk`.
5. Administrator B starts the batch file on Workstation B to begin ConsoleOne.
6. The batch file running on Workstation B identifies the storage server as being mapped to drive M: (or it maps drive M: to that server).
7. The batch file copies the master `snapinprefs.ser` file from the server at drive M: to the `c:\documents and settings\`*`user_ID`*`\.consoleone` directory on Workstation B.

   This is the same version of `snapinprefs.ser` that Administrator A had copied to Workstation A, except that it hasn't been updated yet with Administrator A's addition of `ssp1.spk`.
8. Administrator B creates a new software package, naming it `ssp2.spk`.
9. Administrator B exits ConsoleOne, which updates `snapinprefs.ser` on Workstation B with the `ssp2.spk` path.
10. The batch file running on Workstation B updates the master `snapinprefs.ser` file on the network server at drive M: with the updated `snapinprefs.ser` file from Workstation B.

    This updated master `snapinprefs.ser` file now contains the location of `ssp2.spk`.
11. Administrator A exits ConsoleOne, which updates `snapinprefs.ser` on Workstation A with the `ssp1.spk` path.

12. The batch file running on Workstation A updates the master `snapinprefs.ser` file on the network server at drive M: with the updated `snapinprefs.ser` file from Workstation A.

This updated master `snapinprefs.ser` file now contains the location of `ssp1.spk`. However, the location for `ssp2.spk` has been lost, because Workstation B's update of the master `snapinprefs.ser` file was overwritten by Workstation A's later update.

This scenario would cause Administrator B to lose access to `ssp2.spk`, because the master `snapinprefs.ser` file no longer contains a record of `ssp2.spk`'s location. It was replaced with Administrator A's `snapinprefs.ser` file containing only `ssp1.spk`'s location. However, you can manually insert `ssp2.spk` into ConsoleOne (using the Insert Software Package option), so that it is listed in the `snapinprefs.ser` file along with `ssp1.spk`.

For this multiple-workstation management method to work, you must ensure that the master `snapinprefs.ser` file you keep on the network server is only used by one workstation at a time for creating, deleting, or compiling `.spk` files. However, you can use multiple workstations to simultaneously view or edit a Server Software Package object's properties, because the viewing and editing functions do not cause updates to a `snapinprefs.ser` file.

WARNING: You can perform edits to the properties of the Server Software Package object without affecting the `snapinprefs.ser` file. However, because Server Software Package objects are not in eDirectory™, but only in a name space, the `.spk` files might not have file-locking protection, unless the server's operating system provides this functionality. Therefore, you should devise management controls to protect against overwriting `.spk` files when using multiple workstations to manage software packages.

5.2.10 Failure of Software Package Installations

If a server fails to meet any of the software package requirements, it is not installed:

- "Failure During an Installation" on page 254
- "Failure of a Component" on page 254

Failure During an Installation

The system tracks all changes made by the installation of a software package. Except as noted under Section 5.2.11, "Rolling Back Software Package Installations," on page 255, if a server meets the requirements and the installation begins, then a failure condition halts the installation prematurely, the installation program automatically returns the server to the state it was in before the installation began, undoing what had been done to that point.

Failure of a Component

If a server meets the software package requirements, and some of the components meet the installation requirements and some do not, the installation is completed except for the components where the requirements were not met. In this case, you would have a partial installation of the package.

You should organize your software packages and their components so that if this happens, it does not leave disconnected or incomplete files or applications on the target machine.

5.2.11 Rolling Back Software Package Installations

Software package rollback is enabled by default. You should not disable rollback, unless you know the installation never needs to be undone.

Rollback Methods

There are two ways you can roll back a software package installation:

- On the server containing the package to be rolled back, enter `package rollback` at the server's `ZENworks Server Management` console prompt.
- Use a Web browser to access the ZENworks Server Management role and select the rollback option. For more information, see Chapter 2, "Novell iManager," on page 63.

The software package is uninstalled, leaving the server as if it had never been installed, except for any changes that might have been made to the server in using the installed application.

Rollback works, even if some components have not been installed during a successful package installation, because the installation program tracks what was and wasn't installed by the software package.

Rollback of Older Installations

When you roll back a software package installation, it is the last software package installed on that server. If that's not the one you need to roll back, you must roll back each installation, beginning with the more recent installations first, until you have rolled back the desired package.

For example, you installed three software packages on a server (Package1, Package2, and Package3). Package1 was installed first and Package3 was installed last. If you want to roll back Package2, you must first roll back Package3. To do so, you need to enter `package rollback` at the server's `ZENworks Server Management` console prompt once for Package3, then again for Package2.

Rollback Exceptions

You can normally undo a successful software package installation by rolling it back. However, any software package installation that runs a program such as a NetBasic script, a Java Class, or an NLM that modifies the server cannot be rolled back successfully, because those programs or services might have launched other programs that made changes on the server, which cannot be tracked for rolling back.

5.3 Planning Server Software Packages

Review each of the following sections and take notes as instructed. This information will help you to configure your software packages and their components.

- Section 5.3.1, "Which Files or Applications Do I Want to Distribute?," on page 256
- Section 5.3.2, "What Software Package Components Are Needed?," on page 256
- Section 5.3.3, "What Minimum Requirements Are Needed?," on page 257

After planning your software package, continue with Section 5.4, "Setting Up Server Software Packages," on page 257.

5.3.1 Which Files or Applications Do I Want to Distribute?

You can distribute software packages containing files and applications for servers, as well as software packages containing end-user applications for further distribution in ZENworks Desktop Management to workstations. For information on configuring a Desktop Application object, see "Application Management" in the *Novell ZENworks 7 Desktop Management Administration Guide*.

If you have ZENworks 7 Desktop Management installed, you can also distribute desktop applications using Tiered Electronic Distribution, instead of including them in software packages. For more information, see Chapter 6, "Desktop Application Distribution," on page 279.

You can include a file or application in more than one software package. For instance, a word processor application could be included in a software package designed for a secretarial group and one designed for a financial group.

Where applicable, organize the files and applications into logical groups for inclusion in software packages.

Follow the steps under Section 5.4.2, "Creating a Server Software Package," on page 261 and Section 5.4.4, "Creating the Software Package Components," on page 263 and note the information you need to know for creating the software package and its components.

5.3.2 What Software Package Components Are Needed?

You can have one or more components in a software package. For example, if you create a software package for installing virus protection software, you might want one component to be the original virus protection program, and another component a current virus pattern update file.

Components in a software package can each have the same or different installation requirements. If you give the components different requirements, they might not all be installed together. You can save time and minimize error by giving all of the components the same requirements.

IMPORTANT: You should include in the same component the files and applications that are dependent on each other. This prevents problems running the files or applications if a critical component is not installed. If you need to split an application's files into multiple components, make sure that you make each component's requirements the same, so that they all are either installed or not installed.

Follow the steps under Section 5.4.5, "Configuring the Software Package Components," on page 263 and note the information you need to know for configuring the package components.

5.3.3 What Minimum Requirements Are Needed?

Minimum requirements establish whether a software package can be installed on the target machine. If these requirements are all met, you can install the software package on that server.

However, you can establish requirements for the software package as a whole, as well as for each package component. Therefore, if the package's requirements were all met, but some component requirements were not met, only part of the package would be installed.

Follow the steps under Section 5.4.3, "Configuring the Server Software Package," on page 262 and note the information you need to know for configuring the software package.

5.4 Setting Up Server Software Packages

To set up a software package for distribution, perform the following tasks in order:

1. "Setting Up Multiple-Workstation Management for Server Software Packages" on page 257
2. "Creating a Server Software Package" on page 261
3. "Configuring the Server Software Package" on page 262
4. "Creating the Software Package Components" on page 263
5. "Configuring the Software Package Components" on page 263
6. "Compiling a Software Package" on page 274
7. "Distributing the Software Package" on page 274

5.4.1 Setting Up Multiple-Workstation Management for Server Software Packages

If you want to manage your software packages from multiple workstations, do the following in order to set up managing the replication of a master copy of the `snapinprefs.ser` file to multiple workstations; otherwise, continue with Section 5.4.2, "Creating a Server Software Package," on page 261.

1. "Setting Up the Master Snapinprefs.ser File" on page 257
2. "Creating and Using the ConsoleOne Batch File" on page 258

Setting Up the Master Snapinprefs.ser File

For the following instructions, select any workstation that you use for managing software packages. If you have already created software packages using a workstation, select that workstation so you do not lose any software package information stored in the workstation's `snapinprefs.ser` file.

1 Map a drive to the server where you want to store your `.spk` and related `.cpk` files.

This drive letter should be one that can be used by all of the other workstations you use to manage software packages. This drive letter is written to the `snapinprefs.ser` file as part of the path information for each listed `.spk` file, so it should be a fixed drive letter that all workstations use.

The drive letter is also used in the batch file that you use to start ConsoleOne, which provides each workstation access to the same `.spk` file locations.

2 If you already have Server Software Package objects created by this workstation, skip to Step 5.

or

If you have not yet created any Server Software Package objects using this workstation, start ConsoleOne.

This version of ConsoleOne must have the Policy and Distribution Services snap-ins installed.

3 In the Server Software Package namespace, create a Server Software Package object.

You do not need to fully configure the Server Software Package object at this time. Just give the package a name and provide a location and filename for the `.spk` file. Make sure you use the drive mapping you used in Step 1.

For information on creating software packages, see Section 5.4, "Setting Up Server Software Packages," on page 257.

4 Exit ConsoleOne.

This step is important to make sure that the `snapinprefs.ser` file is created for this workstation.

5 On the network server you use to store the master copy of the `snapinprefs.ser` file, create a directory named `\C1` at the root of the drive.

You can select any safe location on the server for the master `snapinprefs.ser` file.

The batch file sample provided below uses a directory named `\C1`. You can modify the batch file if you want to use a different directory name, and you can include path information; however, do not use variables for the root location.

For example,

`\zenworks\clssp`

could be used to replace the `\C1` directory name.

6 Copy the workstation's `snapinprefs.ser` file from:

`c:\documents and settings\`*user_ID*`\.consoleone` (Windows 2000)

to the `\C1` directory on the network server.

This becomes the master `snapinprefs.ser` file that is updated with new `.spk` paths, provided you are using the batch file documented in "Creating and Using the ConsoleOne Batch File" on page 258.

7 Continue with "Creating and Using the ConsoleOne Batch File" on page 258.

Creating and Using the ConsoleOne Batch File

Review the following sections to understand, create, and use the batch file:

Sample Batch File

```
@echo off
REM map a network drive
net use m: \\server1.servers.novell.com\vol1

REM create a backup copy of the workstation's .ser file
copy "%USERPROFILE%\.consoleone\snapinprefs.ser"
"%USERPROFILE%\.consoleone\snapinprefs.tmp"

REM copy the master .ser to the workstation
copy m:\c1\snapinprefs.ser "%USERPROFILE%\.consoleone\snapinprefs.ser"

REM start ConsoleOne
c:\Novell\ConsoleOne\1.2\bin\ConsoleOne.exe

REM batch file control returns after exiting ConsoleOne
REM copy the updated .ser to server
copy "%USERPROFILE%\.consoleone\snapinprefs.ser" m:\C1\snapinprefs.ser

REM restore the backup copy of the workstation's .ser file
copy "%USERPROFILE%\.consoleone\snapinprefs.tmp"
"%USERPROFILE%\.consoleone\snapinprefs.ser"

REM delete the mapped network drive
net use m: /delete
@echo on
```

What the Batch File Does

- It maps a network drive for accessing the server where you are storing `.spk` and `.cpk` files.

- It uses the %USERPROFILE% Windows variable to locate the Server Management `\consoleone` directory. This variable is also used by Server Management to determine where it creates the `\consoleone` directory and writes the `snapinprefs.ser` file.

- It creates a backup `.tmp` copy of the `snapinprefs.ser` file.

- It copies the master `snapinprefs.ser` file from the `\C1` directory on the server to the workstation's `\consoleone` directory.

- It starts ConsoleOne.

- After you have exited ConsoleOne, the batch file copies the updated `snapinprefs.ser` file from the workstation's `\consoleone` directory to replace the version in the `\C1` directory on the server. This becomes the new master `snapinprefs.ser` file.

- It restores the backed-up copy of the `snapinprefs.ser` file from the `.tmp` file.

- It unmaps the drive letter to the server.

Creating Your Batch File

1 Copy the text from the above sample batch file into a text editor.

2 Replace the `m:` drive letter with one that each of your workstations has free. Make sure you do this wherever `m:` exists in the batch file.

3 Edit the `net use m: \\server1.servers.novell.com\vol1` line by replacing it with the path to the server volume or shared folder of the server where you are storing the `.spk` and `.cpk` files.

4 Save the batch file on your workstation and give it a name, such as:

`C1SSP.BAT`

5 Copy this batch file to each workstation that you use to manage software packages.

Optional Modifications to the Batch File

- If you installed ConsoleOne to a different location on the workstation than the one indicated in the batch file sample, modify the `c:\novell\consoleone\1.2\bin\consoleone.exe` line to reflect the location of the `consoleone.exe` file on the workstation.

  You should make this modification in each individual batch file copy on a workstation where the default ConsoleOne path was not used.

- This batch file can also be used by a workstation to start an instance of ConsoleOne that is installed on a server. Modify the `c:\novell\consoleone\1.2\bin\consoleone.exe` line to reflect the location of the `consoleone.exe` file on the server. Make sure the drive letter is the one being used for accessing the server (see).

- If the `\consoleone` directory path is different between workstations because the %USERPROFILE% variable was not used, you need to edit any lines containing the variable, as necessary. Open the copy of the batch file on a workstation where the %USERPROFILE% variable was not used and edit the lines containing the variable to reflect the correct path to the `\consoleone` directory.

- If you created a directory other than `\C1` on the server, replace `\C1` wherever it exists in the batch file with the directory that you specified in .

- The batch file creates a `.tmp` version of the `snapinprefs.ser` file. This allows you to maintain the version of the `.ser` file on the workstation that existed before you used the batch file. However, if you want the workstation's version to always match the master version it copied to the server, remove the following two lines from the batch file:

  ```
  copy "%USERPROFILE%\.consoleone\snapinprefs.ser"
  "%USERPROFILE%\.consoleone\snapinprefs.tmp"

  copy "%USERPROFILE%\.consoleone\snapinprefs.tmp"
  "%USERPROFILE%\.consoleone\snapinprefs.ser"
  ```

- If you cannot use the same drive letter for all workstations, you can use the %1 argument in the batch file, but only if you are using UNC paths for all of your `.spk` files. To do this, replace all occurrences of `m:` with `%1`. Then, when you execute the batch file from a command line, add the drive letter after the batch file's name. For example,

  `C1SSP R:`

  causes the batch file to use R: as the drive for locating the master copy of the `snapinprefs.ser` file.

Using the ConsoleOne Batch File

- Before running this batch file, place a `snapinprefs.ser` file in the `\consoleone` directory of each workstation you use to manage software packages. The batch file assumes that the `.ser` file exists for copying and replacing.

- Before running this batch file, place your master copy of the `snapinprefs.ser` file in the `\C1` directory of the server where you have stored the software package files. The batch file assumes that this `.ser` file exists for copying and replacing.

- Run this batch file any time you plan to add, delete, or compile software packages.

- You do not need to use the batch file when you view or edit the properties of software packages. The add, delete, and compile functions are the only actions that causes the `snapinprefs.ser` file to be updated.

Continue with .

5.4.2 Creating a Server Software Package

1 In ConsoleOne, right-click the Server Software Packages namespace, then click *New Package*.

The Create New Server Software Package Wizard opens.

2 Read the information on the first dialog box, then click *Next*.

3 Provide a name for the software package.

Make this a descriptive name. It is displayed in ConsoleOne under the Server Software Packages object.

IMPORTANT: Do not use double-byte characters in the software package name. This causes an error in any report you run on the software package.

4 Because software packages are file-based, provide the full path and filename, including the `.spk` extension.

If you don't enter the extension, you are prompted to add it.

You can also use UNC paths.

You can store the `.spk` files on a workstation or server. The `.spk` files is typically below 100 KB in size. However, compiled software packages (`.cpk` files) can be in the hundreds of megabytes. For information on storing `.spk` and `.cpk` files, see .

WARNING: Software package full paths and filenames are drive-dependent. A software package can become unusable if you change drive mappings after creating the package. Make sure your entry in this field is not changed. However, if you used a UNC path, this is not an issue.

5 Click *Finish*.

6 Continue with .

5.4.3 Configuring the Server Software Package

After a software package has been created, you need to configure it by setting the prerequisites for installation of the files and applications contained in the package.

To configure a package:

1 In ConsoleOne, right-click a software package, then click *Properties*.

The *Identification* tab should be displayed. If not, select it.

The *Name* field should display the name you gave the package when you created it.

2 Provide a useful description for the software package.

3 If you don't want to be able to roll back to the older version of the server file or application after installing the newer version, click *Disable Rollback*. However, this is not recommended.

For information on rolling back software package installations, see Section 5.2.11, "Rolling Back Software Package Installations," on page 255.

4 Select the *Requirements* tab.

5 Click *Add*, then select a requirement:

Section E.1, "Operating System," on page 424
Section E.2, "Memory (RAM)," on page 427
Section E.3, "Disk Space," on page 427
Section E.4, "SET Commands," on page 428
Section E.5, "Registry," on page 428
Section E.6, "File," on page 429
Section E.7, "Products.dat," on page 429

6 Repeat Step 5 for each requirement.

7 If you want to use variables to customize the installation, select the *Variables* tab, then click *Add*.

8 Provide the variable name and value.

For information on variables, see Section 9.6, "Using Variables to Control File Extraction," on page 357.

9 Repeat Step 7 and Step 8 for each variable.

10 Click *OK* when you have finished configuring.

If you click *Cancel*, none of the configuration changes on any of the tabs are saved.

11 Continue with Section 5.4.4, "Creating the Software Package Components," on page 263.

5.4.4 Creating the Software Package Components

After you have created and configured a software package, you need to create the components of the package, including the individual files or applications for the package.

To create the software package components:

1 In ConsoleOne, right-click a software package (in the left pane), then select *New Component*.

2 Provide the name of the component as you want it to be displayed in ConsoleOne, then click *OK*.

The component is displayed as named under the Software Package object.

3 Repeat these steps for each component needed.

4 Continue with .

5.4.5 Configuring the Software Package Components

After you have created the software package components, you need to configure the prerequisites for each, including identifying the files or applications for the component.

Package components can each have the same prerequisites, which can save time and minimize user error.

To configure a component:

1 In ConsoleOne, right-click a component, then click *Properties*.

The *Identification* tab should be displayed. If not, select it.

2 Provide a useful description for the component.

3 Select a further action for the software package to perform after the installation process has finished from the *After package installation is complete* drop-down list.

4 To continue configuring the component, see each of the following that you might need to configure:

Do not click *OK* on this component's property page until you have finished configuring all of the above items, as needed.

5 Click *OK*.

If you click *Cancel*, none of the configuration changes on any of the tabs are saved.

6 Continue with to ready your software package for distribution.

Requirements

To specify requirements for installing the server files or applications:

1 While displaying the properties of the software package component, select the *Requirements* tab, then click *Add*.

2 Select any of the following requirement items:

For further instructions on configuring an item, see one of the above items.

Continue with the next item to configure before clicking *OK*.

Pre-Installation Load/Unload

To configure certain NLM files or processes to load or unload before installing the software package on a server:

1 While displaying the properties of the software package component, select the *Pre-Installation* tab, then click *Load/Unload*.

2 Click *Add*.

3 Select one of the following:

For further instructions on configuring an item, see one of the above items.

IMPORTANT: If you select a process to be loaded by the software package, and it is already running on the target server, the package installation fails and is rolled back (if rollback is enabled). If the process requires intervention to unload, you must remember to unload it manually before installing the software package.

To make sure that a process is not already loaded when you are including it in the software package, add an unload option for that process before adding the load option—but only if the process does not require user input from the keyboard to unload it.

4 Repeat Step 1 through Step 3 for each NLM or process to be included.

5 Use the arrow keys to arrange the order to execute the NLM files and the processes.

Continue with the next item to configure before clicking *OK*.

Pre-Installation Script

To configure running server scripts before installing the software package on a server:

1 While displaying the properties of the software package component, select the *Pre-Installation* tab, then click *Script*.

2 Click *Add*.

3 Provide the script name.

4 Select the script type (NCF, NetBasic, PERL).

IMPORTANT: NetBasic is not supported on NetWare 6.5 servers.

5 Enter the script text.

WARNING: If a software package passes all requirements and executes the script, processing done by the script cannot be undone by rollback.

6 Repeat Step 2 through Step 5 for each script to be added.

7 Use the arrow keys to arrange the order to execute the scripts.

Continue with the next item to configure before clicking *OK*.

Local File Copy

The Local File Copy component enables copying of files on a server from one location to another using a software package. You can either copy or move the files.

To configure the Local File Copy component:

1 While displaying the properties of the software package component, select the *Local File Copy* tab.

2 Click *Add*.

Local File Copy #1 is the default name. You can edit that name.

3 Fill in the fields:

Source path: Provide the full path where the files to be copied are located.

You can use wildcards in the path:

   * = any number of characters
   ? = any single character in that position
   ??? = any characters in those positions

Target path: Provide the full path where the copied files are to be placed.

You can use wildcards in this path. This path does not need to mirror the source path. However, you could mirror an existing target path.

Include subdirectories: Includes all subdirectories and their files beginning from the directory at the end of the path; otherwise, only the files in the directory at the end of the path are copied.

Maintain attributes: Maintains the file attributes in the target's file system that exist in the source's file system.

Overwrite Destination Files: Overwrites files of the same name in the destination directories, regardless of differences in file dates. If you do not select this option, files of the same name are not replaced.

Maintain trustees: Maintains the file's trustee attributes.

When a file is locked: Select one or both:

- ◆ **Retry __ times:** Retries overwriting a locked file the number of times you select before failing to replace the file. Leave this check box deselected to not replace locked files on the target file system.

- ◆ **Kill connection of open files:** (NetWare only) Attempts to kill the connection of locked files so that they can be overwritten. This applies only to files being extracted, not to files being accessed to build the Distribution. If a file belonging to a Distribution is locked when the Distribution is being built, the build fails. Server and NLM connections cannot be killed.

Error processing: *Fail On Error* is selected by default. This stops the file copying process when an error is encountered in copying. To continue file copying when an error is encountered, select *Continue On Error*.

Operation: Sets whether to copy or move the files identified in the *Source Path* field.

Continue with the next item to configure before clicking *OK*.

Copy File

You can configure the Copy File component to control how files are copied during installation of a software package. This includes adding files to existing directories, creating new directories, adding files and subdirectories to the new directories, and deleting existing files and directories.

A file group is a root item for the component's expandable tree structure. You can have multiple file groups for the Copy File component. A file group is a set of related directories and files. File groups are top-level items and cannot contain other file groups.

The other structure items are directory and file, which are contained within a file group. Directories can contain other directories or files, but not file groups.

IMPORTANT: When you add a file group or directory, you are creating the target paths where the files are to be copied, not the source paths of the files. The source paths are automatically accounted for as you select your source files or directories.

To configure copying files during installation of the software package:

1 While displaying the properties of the software package component, select the *Copy File* tab.

2 To create your first file group, do the following:

  2a Click the down-arrow on the drop-down list next to the *Add* button, select *Add File Group*, then click *Add*.

    Because files and directories must be contained within file groups, you are prompted to create a file group the first time you click *Add*, regardless of the type you are attempting to add.

You should create one file group for each specific target location. For example, `c:\files`, `c:\data\accounting`, and `c:\data\personnel` could be different locations on a `C:` drive where you want to copy different groups of unrelated files.

2b Name the file group, then provide its target path.

The file group's target path specifies the base path from where all directories and files within the group are installed.

2c To specify what to do when a file group location is locked, select the check box for one of the following:

1. Retry (enter the number of retry times)
2. Kill Connection of Open Files (NetWare only)
3. Fail With Error

Retries are about 5 seconds apart. Therefore, 12 retries would take about one minute.

3 To create a target directory under a file group or another directory, select the file group or directory, in the drop-down box select *Add Directory*, click *Add*, then do the following:

3a Because Directory is the default directory name, to rename the directory, right-click *Directory*, click *Rename*, type the desired directory name, then press Enter.

When entering information into this field, you must press Enter for the change to be saved.

To match an existing target directory for deleting or copying files, you must enter the exact name.

IMPORTANT: If you provide an existing directory name and that directory is marked as Read Only on the destination server's file system, the Software Package Distribution fails when the Subscriber tries to extract the Distribution, because it cannot write to that directory. Therefore, you must know the attributes of existing target directories and remove their Read Only directory attributes.

You must create the same directory structure in the File Copy component as exists in the target location so that the directory name you provide here is in the same sequence in the path.

3b To determine whether to create or delete the directory, select the mode from the *Copy Mode* drop-down list.

Create: If you select *Create* and the directory does not exist, the directory is created. If you select *Create* and the directory does exist, the directory is not created, and no error is encountered.

Delete: If you select *Delete* and the directory exists, the directory is deleted, including any subdirectories and files under it. If you select *Delete* and the directory does not exist, the directory is not deleted, and no error is encountered.

WARNING: If you plan to set the *Copy Mode* as Delete for any directories you add, and you do not want any parent directories that you have added to also be deleted, place those parent directories in the *Target Path* field of the file group. For example, if you want to delete `c:\winnt\cookies`, but do not want to delete the `\winnt` directory, enter `c:\winnt` in the *Target Path* field, click *Add* to enter the `\cookies` directory in the tree structure, then click *Delete* for the *Copy Mode* field. For example:

Target = `c:\winnt`

Tree structure = `cookies`

causes only cookies and all of its files and subdirectories to be deleted.

Conversely, both the \winnt and \cookies directories are deleted if you enter c:\ in the *Target Path* field, click *Add* to enter the \winnt directory in the tree structure, click *Add* to enter the \cookies directory under \winnt in the tree structure, then click *Delete* for the *Copy Mode* field.

For example:

Target = c:\

Tree structure = winnt\cookies

causes winnt and all of its files and subdirectories to be deleted.

4 To add files or source directories under a file group or directory in the tree structure, select a file group or directory, in the drop-down box select *Add File*, click *Add*, then do the following:

4a Select the files or directories using the Open dialog box.

These directories and files are displayed directly under the file group or directory you selected in Step 3.

For the destination server's file system, attributes of the copied files and directories are not maintained. For more information, see Step 4c.

If you selected a directory on the Open dialog box, it is not displayed expanded. Click the plus signs to expand the existing structure under the directory that you added.

In the Open dialog box, the Recurse Directories option is selected by default. To only select files in this directory, select the Recurse Directories check box to disable it and none of the subdirectories are selected.

To exclude files or subdirectories from being selected, select the Exclude Selected Subdirectory option, select the files or directories to be excluded (use Shift and Ctrl for multiple select), then click Open.

If you exclude files or subdirectories, it does not remove them from the file system. It only prevents them from being selected.

For information on removing files or subdirectories from the tree structure after adding files and directories, see Step 8.

4b To configure a subdirectory that was added, do the following:

◆ Select the subdirectory, then select the *Copy Mode* (whether to Create or Delete the directory).

WARNING: When you set the *Copy Mode* to Delete, it causes deletion of the target directory and all of its files and subdirectories.

◆ To rename a subdirectory that was added, right-click the subdirectory, click *Rename*, type a new directory name, then press Enter.

When entering information into this field, you must press Enter for the change to be saved.

If you rename a directory that was selected through the Open dialog box, make sure that the new name meets your expectations for the target location.

Because only selected files have their path remembered for copying, renaming a directory does not affect file selection. In other words, you can give a target directory a different name than its source, and still have the same files copied under it.

4c To configure an added file, select the file, then do the following:

◆ To determine the file's copy mode, select a mode from the *Copy Mode* drop-down list.

You must select an option for every file. You can select multiple files where you want the mode to be the same.

The options are: Copy Always, Copy If Exists, Copy If Does Not Exist, Copy If Newer, Copy If Newer and Exists, and Delete.

WARNING: When you set the *Copy Mode* to Delete, it causes deletion of the selected file from the target server.

◆ Select the check box for each attribute that should apply to the selected files.

Attributes do not default. You must set them for the destination server. They are not carried over from where you obtained the file.

IMPORTANT: Do not select all of the attributes for a file, or an exception is thrown on the server.

When setting the attribute of an executable file, set it to Read Only. Do not set it to Execute. If you mark a file as Execute, the NetWare® CLIB API does not allow you to change it to a different attribute. To change the attribute from Execute to Read Only after the software package has been installed, you need to manually delete the file, replace it, then set its attribute again.

---

5 To create another file group, do the following:

5a Click the down-arrow on the drop-down list next to the *Add* button, select *Add File Group*, then click *Add*.

It doesn't matter what you have selected in the tree structure; the file group is automatically placed at the first tree level, equal to any other file groups that are displayed.

5b Name the group.

5c Provide its target base path.

5d To indicate what to do when a group location is locked, select the check box for one of the following:

1. Retry (enter the number of retry times)
2. Kill Connection of Open Files (NetWare only)
3. Fail With Error

6 Repeat Step 5 or each additional file, directory, or file group to be added.

7 If you want the file groups to be copied in a particular order, use the arrow keys to arrange the order of the file groups.

The arrows are dimmed if the file group you have selected has no valid up or down movement available to it.

8 To remove a file group, directory, or file, select it, then click *Remove*.

You can use the *Remove* button to prune the tree structure of unwanted files or directories.

You can use the Shift and Ctrl keys to select multiple items for removal.

IMPORTANT: If you remove a file group or directory, all files and directories displayed below it are also removed, but only from this tree structure, not from the source file system.

Continue with the next item to configure before clicking *OK*.

Text Files

To configure making changes to text files during installation of the software package:

1 While displaying the properties of the software package component, select the *Text Files* tab.

2 Click *Add*.

After one text file has been added, you are given the opportunity to select whether you are adding another text file or adding another change item for the selected text file.

To add another text file: Select *Text File*. It does not matter which text file or change item is selected in the left pane—the text file is added to the far left level.

To add another change to a text file: In the left pane select the text file for the change, click *Add*, then select *Change*. The change item is added under the selected text file.

3 If you are adding a text file, provide the name of the text file.

4 Accept the default name (such as Change #1) or rename it.

If you are adding a text file, click *OK*.

5 Click the down-arrow for the *Change Mode* field, then select the change mode from the drop-down list.

6 Click the down-arrow for the *Search Type* field, then select the search type from the drop-down list.

7 Enter the exact search string.

8 Select the check box if you want the string search to be case sensitive.

9 To find all occurrences of the search string, select the check box (default); otherwise, deselect the check box to find only the first occurrence.

10 Click the down-arrow for the *Result Action* field, then from the drop-down list, select the action that should result if a string is matched.

11 If you are replacing a string or entering a new one, enter the text in the *New String* text box.

12 Repeat Step 2 through Step 11 for each text file to add or each change to be made.

13 To reorder the text files and change items, use the arrow keys.

Continue with the next item to configure before clicking *OK*.

SET Commands

For NetWare only.

To configure the target server's SET commands:

1 While displaying the properties of the software package component, select the *SET Commands* tab.

2 Click *Add* to open the NetWare Server SET Commands Wizard.

3 Select the server containing the SET commands, then click *Next*.

> **IMPORTANT:** The Server Management and Java must be running on the server where you want to obtain the SET commands.

4 Select all of the SET commands you want to configure for the target server.

You can select whole categories by selecting the check box for the category, or click the plus sign to expand a SET command category and select the check boxes for individual SET commands to be included.

WARNING: Do not select the Set Developer Option SET command and change Off to On. This parameter is meant to help developers debug server abends. It disables some operating system checking to prevent certain abends from occurring. Also, if the Set Developer Option is enabled, running NCP™ scripts that require keyboard entry could abend the server.

---

5 Click *Finish* when you have completed selecting SET commands.

The selected SET commands are now displayed in the *SET Commands* tab for the file or application component.

6 To edit a SET command, click its plus sign to expand its attributes.

7 To edit an attribute, select the attribute, then click *Edit*.

A dialog box is displayed where you can make changes to the attribute.

8 Repeat Step 7 for each attribute to edit for a given SET command.

9 Repeat Step 6 through Step 8 to edit another SET command's attributes.

Continue with the next item to configure before clicking *OK*.

Registry Settings

To configure registry changes for either NetWare or Windows servers:

1 While displaying the properties of the software package component, select the *Registry Settings* tab, then click HKEY_LOCAL_MACHINE.

HKEY_LOCAL_MACHINE is a Windows registry key. For NetWare, HKEY_LOCAL_MACHINE is also recognized by Server Management as the equivalent to My Server. Therefore, you can use this key for editing both NetWare and Windows registries.

2 Click *Add*.

3 Select from the following:

Section F.1, "Key," on page 431
Section F.2, "Binary," on page 432
Section F.3, "Expand String," on page 432
Section F.4, "(Default)," on page 432
Section F.5, "DWord," on page 433
Section F.6, "Multi-Value String," on page 433
Section F.7, "String," on page 433

For further instructions on configuring an item, see one of the above items.

4 Repeat Step 2 and Step 3 for each registry entry to be made.

5 Use the arrow keys to arrange the order in making registry entries.

Continue with the next item to configure before clicking *OK*.

Products.dat

For NetWare only.

The `products.dat` file can be updated by your software package so that future updates can identify the most recently installed version of the file or application.

WARNING: Modifying `products.dat` could prevent something from running or being installed on the NetWare server. Never modify any entries supplied by Novell.

To determine which action to take for `products.dat`:

1 While displaying the properties of the software package component, select the *Products.dat* tab.

2 Select one of the following:

Option	Description
Add	Adds a new entry
Modify Existing Entry	Searches for a matching ID and modifies the version and description
Replace Existing Entry	Searches for a specific ID and replaces it with a new one
No Action	This is the default. Nothing is done to `products.dat`

3 If you selected *Add*:

    3a Provide the ID of the item to add.

        This is case sensitive. The item is the ID of the new product for the `.dat` file.

    3b Provide the exact version number to add.

    3c Provide the description to add.

4 If you selected *Modify Existing Entry*:

    4a Provide the ID of the item to search for (case sensitive).

    4b Provide the new version number.

    4c Provide the new description.

5 If you selected *Replace Existing Entry*:

    5a Provide the ID of the item to search for (case sensitive).

    5b Provide the exact version number to match.

    5c Provide the new ID.

    5d Provide the new version.

    5e Provide the new description.

Continue with the next item to configure before clicking *OK*.

Post-Installation Unload/Load

To configure certain NLM files and processes to load or unload after installing the software package on a server:

1 While displaying the properties of the software package component, select the *Post-Installation* tab, then click *Load/Unload*.

2 Click *Add*.

3 Select one of the following:

Section D.1, "Load NLM/Process," on page 421
Section D.2, "Load Java Class," on page 421
Section D.3, "Unload Process," on page 422
Section D.4, "Start Service," on page 422
Section D.5, "Stop Service," on page 422

Select an item for further instructions on configuring it.

4 Repeat Step 2 and Step 3 for each NLM or process to be included.

Continue with the next item to configure before clicking *OK*.

Post-Installation Script

To configure running NetWare server scripts after installing the software package on a server:

1 While displaying the properties of the software package component, select the *Post-Installation* tab, then click *Script*.

2 Click *Add*.

3 Provide the script name.

4 Select the script type (NCF, NetBasic, PERL).

IMPORTANT: NetBasic is not supported on NetWare 6.5 servers.

5 Enter the script text.

WARNING: If a software package passes all requirements and executes the script, processing done by the script cannot be undone by rollback.

6 Repeat Step 2 through Step 5 for each script to be added.

7 Use the arrow keys to arrange the order to execute the scripts.

Continue with the next item to configure before clicking *OK*.

5.4.6 Compiling a Software Package

Your software packages (`.spk` files) cannot be installed by Policy and Distribution Services until they have been compiled and have the `.cpk` extension.

To compile a software package:

1 In ConsoleOne, right-click a software package, then click *Compile Package*.

The Compile Server Software Package Wizard opens.

2 Read the information on the first dialog box, then click *Next*.

3 Provide a name and path for the compiled software package (using the `.cpk` extension), then click *Next*.

Select a location where free disk space is adequate for the `.cpk` file. Compiled software packages (`.cpk` files) are generally much larger than the uncompiled (`.spk`) counterparts.

IMPORTANT: If you provide the path and filename of the `.spk` when prompted for the compiled (`.cpk`) filename, the `.spk` is overwritten and can no longer be edited. Therefore, be sure to use the `.cpk` extension when naming the compiled version.

The compiling process could take some time, depending on how many files are involved.

4 When compiling has completed, click *Finish*.

5 Continue with Section 5.4.7, "Distributing the Software Package," on page 274 to distribute your software package (`.cpk`).

5.4.7 Distributing the Software Package

After a software package is ready for distribution, you can distribute it in the following ways:

◆ Use Tiered Electronic Distribution (see Chapter 3, "Tiered Electronic Distribution," on page 85 for instructions on distributing through Tiered Electronic Distribution)

◆ Manually copy the software package file (`.cpk`) to the server and run it from the server's console prompt using the PACKAGE command (see Appendix C, "Server Console Commands," on page 415 for instructions on using the command)

After a software package is installed on a target server, you might need to reboot the server. For example, if `tcpip.nlm` is modified by the package, it cannot be downed—you must reboot the server to run that NLM again. However, you could have the software package cause the server to come down and restart automatically.

5.5 Using Server Software Packages to Delete Directories on Servers

If you want to delete certain directories from a number of different network servers (NetWare, Windows, Linux, and Solaris), you normally do not have an automated method for performing this task. However, if you are using ZENworks 7 Server Management Policy and Distribution Services, the Server Software Packages feature of Server Management provides the capability for you to delete specified directories from any Subscriber server's file system.

To automate the deletion of specified directories on multiple servers, you first set up path variables (if necessary), create a Server Software Package in its namespace in ConsoleOne, compile the

software package, then distribute the package using Tiered Electronic Distribution. No further user intervention is required.

Do the following in order to create a software package that deletes specified directories on a server:

1. "Setting Up Variables for Use With the Server Software Package" on page 275
2. "Creating the Server Software Package" on page 276
3. "Creating and Configuring the Server Software Package Component" on page 276
4. "Compiling the Server Software Package" on page 278
5. "Manually Testing that the Directories Have Been Deleted" on page 278
6. "What's Next" on page 278

5.5.1 Setting Up Variables for Use With the Server Software Package

Before you create the software package, you must set up the variables in your Subscriber objects' properties if you are using variables in paths (for instance, if your target servers have different operating systems, like NetWare and Windows).

1 Identify the directories to be deleted:

  1a Identify the root of the path, such as its volume name (NetWare), drive letter (Windows), or /var/opt/novell/zenworks (for Linux and Solaris). For example, `data:`.

  1b Identify the rest of the path, including the parent directory to the directories to be deleted, such as `\zenworks\pds\ted\dist` where dist is the parent directory.

  1c Identify the directories to be deleted, such as `olddist.Distributions.ZENworks.Novel`l.

  The resulting full path and directory to be deleted would be:

  `data:\zenworks\pds\ted\dist\olddist.Distributions.ZENworks.Nove`
  `ll`

  You might have varying path elements from server to server. You should use variables as necessary to allow for those differences (see Step 2 and Step 3).

2 In ConsoleOne, create a variable to represent data:, D:, or /var/opt/novell/zenworks for each Subscriber where the directories to be deleted reside, such as DELETEDDIRROOT.

  If you name a directory to be deleted that does not exist on a target server, nothing is done for that directory on that server.

  You can also define variables globally using the Tiered Electronic Distribution policy. There, you should define the default value for a variable and allow the exceptions to be defined in the applicable Subscriber objects' properties.

3 In ConsoleOne, create a Subscriber variable to represent where any path elements are different for the Subscriber server.

  For more information on variables, see Chapter 9, "Variables," on page 349.

4 Repeat Step 2 and Step 3 as necessary.

5 Continue with Section 5.5.2, "Creating the Server Software Package," on page 276.

5.5.2 Creating the Server Software Package

1 In the left pane in ConsoleOne where the ZENworks 7 Server Management snap-ins have been installed, right-click the *Server Software Packages* namespace.

2 Click *File > New > Software Package* to start the Create New Server Software Package Wizard.

3 Click *Next*.

4 Provide a name for the software package, such as Delete Old Directories.

5 Specify a path and filename for the software package template file (`.spk`), such as `c:\temp\deletedirs.spk`.

> **IMPORTANT:** If you save your `.spk` file to a network server, use a UNC path so that you still have access to that software package file if your drive letters change.
>
> You can also save your `.spk` files to a workstation or server, because the `.spk` file sizes do not become large. For this particular type of software package (where you are only giving instructions for deleting directories and are not compiling data files), the `.cpk` (compiled software package) version is similar in size. Therefore, for management purposes, you might want to save these `.spk` files and their corresponding `.cpk` files in the same location, which can be on a workstation or server.

6 Click *Finish*.

7 If necessary, click the plus sign to expand the *Server Software Package* namespace to view the new package.

8 Continue with Section 5.5.3, "Creating and Configuring the Server Software Package Component," on page 276.

Unless otherwise instructed, you should perform the steps in the subsequent sections from the same instance of ConsoleOne you used in the above steps, because your `.spk` files are accessible from there.

5.5.3 Creating and Configuring the Server Software Package Component

1 Right-click the software package object that you just created, then select *New Component*.

2 Provide a name for the component, such as Delete Directories.

3 If necessary, click the plus sign to expand the Server Software Package object.

4 Right-click the component and select *Properties*.

5 Select the *Copy File* tab.

6 Click the drop-down list button next to the *Add* button, then select *Add File Group*.

7 Click *Add*.

8 Provide a name for the file group, such as Delete Working Directories.

9 In the *Group Target Path* field, specify the name of the variable that you created containing the location of the directories to be deleted, and add any path information that is not contained in the variable; however, do not specify the name of the directory to be deleted as part of that path.

For example, if the location for the directories to be deleted is the same for all target servers, specify the actual volume (NetWare) or drive (Windows) with the path information (which can also contain variables).

However, if you need to use variables because the server operating systems are different, then specify the variable name (within the % symbols) plus the full path (which can also contain variables) to the directory just above the directories to be deleted. For example, %DELETEDDIRROOT% (variable name) and `%TARGET%\pds\ted\dist` (full path to the parent directory of the directories you want to delete).

IMPORTANT: When using variables, the path you provide must be the directory containing the directory to be deleted. In Step 11 you add the actual directory names to be deleted.

---

10 Click *OK* to exit the dialog box.

11 Click the drop-down list button again and select *Add Directory*.

Make sure you first select the tree item under which you want to add this directory.

12 Click *Add*.

13 To change the name ("Directory") that defaults in the tree structure to the actual directory name that you want deleted (such as `olddist.Distributions.ZENworks.Novell`), edit the directory name and press the Enter key to save the change.

If you do not press the Enter key, "Directory" is displayed again. The *Rename* button allows you to edit the directory name.

14 Click the drop-down list button next to the *Copy Mode* combo box, then select *Delete*.

15 Click *Apply*.

16 Repeat Step 10 through Step 15 for each directory you want this software package to delete using this component's file group.

You can start at Step 6 to add other file groups, or from Step 1 to add a new component. You might want to repeat from these steps if you cannot add all of your directories to be deleted under the file group that you created in Step 6.

17 When finished configuring the software package component, click *OK* or *Close*.

Using the examples from the above steps, you would have entered:
```
%DELETEDDIRROOT%
```
and
```
%TARGET%\pds\ted\dist
```
and
```
olddist.Distributions.ZENworks.Novell
```
in order to delete the directories having the following paths:
```
data:\zenworks\pds\ted\dist\olddist.Distributions.ZENworks.Novell
d:\zfs\zenworks\pds\ted\dist\olddist.Distributions.ZENworks.Novell
```

18 Continue with Section 5.5.4, "Compiling the Server Software Package," on page 278.

5.5.4 Compiling the Server Software Package

You now have a `.spk` file that serves as the template for what you want to delete. You need to compile this `.spk` file into a `.cpk` file.

1 Right-click the software package, such as Delete Old Directories.

2 Select *Compile* to start the Compile Software Package Wizard.

3 Click *Next* on the first page of the wizard.

4 Provide the full path and filename for the `.cpk` file that you are generating.

IMPORTANT: Do not use the `.spk` extension for this filename, or your template file could be overwritten by its compiled version if they are stored in the same location. This would prevent you from making further edits to the software package. You can use the same filename, such as DELETEDIRS, but you should use only the `.cpk` filename extension.

---

5 Click *Next*, then click *Finish*.

6 Continue with Section 5.5.5, "Manually Testing that the Directories Have Been Deleted," on page 278.

5.5.5 Manually Testing that the Directories Have Been Deleted

The software package is now ready for sending as a Software Package Distribution. However, for testing, you can manually process the software package on one of the target servers to determine that the directories were deleted as intended.

1 On a server where you want to delete a directory, create a directory that is contained in your software package (such as `olddist.Distributions.ZENworks.Novell`) under `\zenworks\pds\ted\dist`.

2 Copy the `.cpk` file (for example, `deletedirs.cpk`) to the `\temp` directory on that server.

3 At the server's `ZENworks Server Management` console prompt, enter the PACKAGE PROCESS command to process the software package.

For example, if it is a NetWare server, at the `ZENworks Server Management` prompt you should enter:

```
package process data:\temp\deletedirs.cpk
```

Server Management processes the package and report that it has finished processing. Check the server's file system to see that the `\olddist.Distributions.ZENworks.Novell` directory, or the directories you specified, were deleted.

5.5.6 What's Next

After you are satisfied with the result of your test, you can distribute the `deletedirs.cpk` file using Tiered Electronic Distribution to all your target Subscriber servers with your new Software Package Distribution in order to delete directories on your Subscriber servers' file systems.

Desktop Application Distribution

<div style="text-align: right; font-size: 3em;">6</div>

Novell® ZENworks® Server Management provides Policy and Distribution Services integration with ZENworks Desktop Management's Novell Application Management.

Desktop Application Distributions can be sent to only Linux, NetWare®, and Windows servers. This Distribution type is not supported on Solaris servers.

The following sections provide information on understanding, setting up, and using the integration between ZENworks Server Management and ZENworks Desktop Management:

- Section 6.1, "Understanding Desktop Application Distributions," on page 279
- Section 6.2, "Requirements," on page 295
- Section 6.3, "Creating a Desktop Application Distribution," on page 296
- Section 6.4, "Rebuilding Desktop Application Distributions," on page 303
- Section 6.5, "Cleaning Up Desktop Application Distribution Files," on page 305
- Section 6.6, "Sending Desktop Application Distributions Tree-To-Tree," on page 305

6.1 Understanding Desktop Application Distributions

Server Management allows you to solve geographic, workload, and redundancy issues for applications distributed by Novell Application Launcher™ that might otherwise require much of your time in manual configuration work in Desktop Management. Review the following sections to see how Server Management can help you to automate much of your desktop application work.

- Section 6.1.1, "The Purpose of Desktop Application Distributions," on page 279
- Section 6.1.2, "Distributed Application Issues," on page 281
- Section 6.1.3, "Miscellaneous Issues," on page 293

6.1.1 The Purpose of Desktop Application Distributions

- "Applications in Desktop Management" on page 279
- "Distributed Applications in Server Management" on page 280

Applications in Desktop Management

In Desktop Management, you can create Application objects so that users or workstations can receive their applications through Novell Application Launcher. An Application object contains pointers to the files belonging to the application, and also contains configuration parameters for how the application is to installed and configured on the desktop.

In Desktop Management, the files belonging to an application can exist on any server, and the related Application object can exist anywhere in the tree. Therefore, for a workstation to receive an application through Novell Application Launcher, the application's files are copied from a server and installed on the workstation.

However, problems can arise for the Desktop Management administrator, such as:

- **Network traffic:** Many users or workstations can create heavy network traffic (especially across slower WAN links) to obtain their applications

  To address the geographic issue of heavy network traffic, if you use only Desktop Management, you would need to do a lot of manual work. You would have to re-create and custom-configure the Application objects multiple times and copy their files to the various servers that would locally service their workstations.

- **Local application access:** Users need local access to their applications no matter where they connect to their network

  You must manually create duplicate Application objects and create a site list in each copy of the Application object.

  For more information on site lists, see "Setting Up Site Lists" in the *Novell ZENworks 7 Desktop Management Administration Guide*.

- **Server overload:** A server loaded with various application files can be over-worked to service all of its workstations

  If you use only Desktop Management, you can configure Load Balancing (sharing the distribution workload between servers) in an Application object to address a server overload condition by having multiple servers being able to perform the same service. However, you would need to do a lot of manual work to use this feature.

- **Server redundancy:** If a server loaded with various application files goes down, its workstations cannot receive those applications

  If you use only Desktop Management, you can configure Fault Tolerance (server redundancy) in an Application object to address the situation where a server goes down by having multiple backup servers listed in the Application object. However, you would need to do a lot of manual work to use this feature.

Server Management provides solutions to resolve these geographic and manual work issues. To see how, continue with "Distributed Applications in Server Management" on page 280.

Distributed Applications in Server Management

Server Management provides a Desktop Application Distribution that allows you to minimize your network traffic, local application access, server bandwidth, and redundancy issues with less effort on your part.

For example:

- **Network traffic:** Create a Desktop Application Distribution that contains your applications, then the Subscribers in each of your geographic areas create local copies of these applications. There, Novell Application Launcher can use these local applications to service the Subscriber server's users and workstations.

- **Local application access:** After creating and sending a Desktop Application Distribution, link up site lists so that users who travel between geographic locations can have local access to their applications.

  For information on how Server Management sets up site lists, see Step 13 on page 302.

- **Server overload:** Through the Load Balancing feature, you can utilize multiple servers to service a large number of users or workstations via Novell Application Launcher. Simply use a

common working context for each of the servers receiving the Desktop Application Distribution. Then, you have multiple servers available for load balancing.

> **IMPORTANT:** Load balancing is concerned with access to the source paths on Subscriber servers, not with access to the distributed Application objects.

- **Server redundancy:** Through the Fault Tolerance feature, you can have redundancy when servers go down by having other servers equally able to service your users and workstations via Novell Application Launcher. Simply use a common working context for each of the servers receiving the Desktop Application Distribution. Then, you have multiple servers available for fault tolerance.

> **IMPORTANT:** Fault tolerance is concerned with access to the source paths on Subscriber servers, not with access to the distributed Application objects.

To do these things, you simply need to:

1. Create one Desktop Application Distribution for an application, or group of applications.
2. Send the Distribution to multiple servers.

   Server Management automatically configures the application according to each server's environment.
3. Manually assign the necessary users or workstations to the groups that are associated with the new Application objects.
4. Click one button to link up the site lists.

Each server then has:

- Its own copy of an application's files on its file system
- Access to the Application object pointing to those files

  The Application object is used to install the application on the workstations through Novell Application Launcher.

The Distribution process automatically does the multiple Application object creation, custom configuration, and file-copying work.

To further understand how Server Management can resolve these issues, continue with Section 6.1.2, "Distributed Application Issues," on page 281.

6.1.2 Distributed Application Issues

When sending a Desktop Application Distribution, some content in an Application object is kept, some is not kept, and some is modified. The following sections explain this:

- "Understanding Golden and Distributed Application Objects" on page 282
- "Maintaining a Golden Application's Attributes" on page 282
- "Maintaining Associations When Distributing Objects" on page 284
- "Maintaining Application File Rights" on page 285
- "Subscriber Working Context Conflicts" on page 286
- "Maintaining Source Paths" on page 287

Understanding Golden and Distributed Application Objects

When you create a Desktop Application Distribution, you select an application object to be distributed. In Server Management, this is known as the "golden" Application object. All of the Application objects that are created by the Distribution are referred to as the "distributed" Application objects.

- "Uniqueness of Golden Applications" on page 282
- "Synchronizing Golden and Distributed Applications" on page 282

Uniqueness of Golden Applications

As an administrator, you should keep track of which objects are golden Application objects for the Distributions, because Application objects themselves do not have any visual designation in ConsoleOne® to identify them as such.

Because normal Desktop Management activity associated with Application objects can cause the object's internal revision number to change, unnecessary deltas of a Distribution could be triggered and sent. For example, a Distribution rebuild could be triggered by a simple change in a User Group object that is associated with an application contained within the Distribution, which information is not even transferred to the distributed applications. Therefore, your golden Applications should not be used by users or workstations.

We recommend that you keep your golden Application objects in a unique Novell eDirectory™ context and associate users and workstations to only the distributed Application objects. For more information, see Section 6.4, "Rebuilding Desktop Application Distributions," on page 303.

Synchronizing Golden and Distributed Applications

When a Distribution is rebuilt and resent, all distributed Application objects are synchronized with the golden Application object. In other words, if you make important changes in a distributed Application object, but not the golden Application object, then you rebuild and send the Distribution again, you could lose your changes, because the Distribution only uses the content in the golden Application object to update the distributed objects. Therefore, the golden Application objects are the only objects that you should modify when you want to re-send the Distribution.

However, you can make changes to distributed Application objects that will not be overwritten, if those changes are in the attributes that are not normally overwritten by a re-sent Distribution. This is explained in the next section.

Continue with "Maintaining a Golden Application's Attributes" on page 282.

Maintaining a Golden Application's Attributes

Server Management distributes most attributes that exist in a golden Application object, but not all of them. Therefore, various outcomes can occur for the attributes contained in distributed Application objects any time a Distribution is rebuilt.

The following sections provide information on when attributes are or are not distributed:

- "Attributes Distributed" on page 283
- "Attributes Not Distributed" on page 283
- "Attributes Sent Only Once" on page 283
- "Attributes Modified" on page 284

Attributes Distributed

If they can be modified in ConsoleOne, all attributes not listed in the following three sections are distributed as they exist in the golden Application object. These attributes are read from the golden Application object when building the Distribution and are sent every time Server Management creates or updates the distributed Application object.

All attributes contained in a golden Application object, not just the updated attributes, are updated in the distributed Application objects when a Distribution is rebuilt, sent, and extracted. This means that all distributed Application objects are kept in sync with their golden applications, except as noted in the next three sections.

Attributes Not Distributed

The attributes (listed by eDirectory attribute name in Table 6-1) are never read by the Distribution building process, and are not populated by Server Management in the distributed Application object:

Table 6-1   *Location of Properties for Attributes Not Distributed*

Attribute Name	Location in the Application Object's Properties
App:FS Rights Path	Common tab > File Rights subtab > Path column.
App:FS Rights Volume	Common tab > File Rights subtab > Volume column.
App:Printer Ports	Common tab > Drives/Ports subtab > Ports to be Captured list box.

This list includes only those attributes that you can modify in ConsoleOne.

Attributes Sent Only Once

The attribute (listed by eDirectory attribute name in Table 6-2) is sent only once to provide an initial contact list:

Table 6-2   *Location of Properties for Attributes Sent Only Once*

Attribute Name	Location in the Application Object's Properties
App:Contacts	Identification tab > Contacts subtab.

This attribute is not updated by any subsequent Distribution updates. This prevents changes to this attribute in the distributed Application object from being overwritten by an original or updated contacts list in the golden Application object.

This attribute can be modified in ConsoleOne.

Attributes Modified

The attributes (listed by eDirectory attribute name in Table 6-3) are read from the golden Application object when the Distribution is built, but are modified to fit the Application object's new environment when the distributed Application object is created in the target server's working context:

Table 6-3 *Location of Properties for Attributes Modified*

Attribute Name	Location in the Application Object's Properties
ACL	NDS Rights tab > Trustees of This Object subtab.
App:Alt Back Link	Fault Tolerance tab > Remote Alternate App subtab.
App:Associations	Associations tab.
App:Back Link	Run Options tab > Application Dependencies subtab > Show Chain button.
App:Fault Tolerance	Fault Tolerance tab > Fault Tolerance subtab.
App:Load Balancing	Fault Tolerance tab > Load Balancing subtab.
App:Site List	Distribution tab > Link Up Site List button (which only displays if the Server Management snap-ins are installed in ConsoleOne). For how and why to use this button, see Step 13 on page 302.
Application GUID	Distribution Options tab > Options subtab > GUID field.
creatorsName	Listed on the Other tab (you must click Show Read Only to view).
modifiersName	Other tab (you must click Show Read Only to view).
Object Class	Listed on the Other tab.
Revision	Listed on the Other tab (you must click Show Read Only to view).
Used By	Listed on the Other tab (you must click Show Read Only to view).

This list includes only those attributes that you can modify in ConsoleOne, and they are only displayed in an Application object when needed to define the application.

Maintaining Associations When Distributing Objects

When configuring a Desktop Application Distribution, you can specify to maintain associations. This means that you want attribute associations set in the golden Application object to be maintained in the distributed Application object that is created by the Distribution.

The Desktop Application Distribution requires some manual processes, such as adding the applicable users or workstations to the distributed Application object, which is empty of this information in Desktop Application Distribution object. This is because users and workstations can be different for each server receiving a distributed application.

If you select the Maintain Associations option, then attribute associations are handled in the following way:

- ◆ **Maintained:** User Group, Workstation Group, Organization, and Organizational Unit objects.

These are trusted groups and containers (within the source root container). They are maintained in the following manner:

- **Created new:** Group and container objects, if they do not exist.

  You need to manually populate them with the users and workstations who need the distributed applications.

- **Not overwritten:** Group and container objects, if they already exist.

  If group and container objects already exist and have been assigned to the distributed Application object, those settings are not overwritten, because they could already be populated with the users and workstations that need to use the distributed applications. If you want to add other users or workstations to existing groups or containers, you must add them manually.

- **Not created:** User and Workstation objects.

  You can add the applicable users and workstations to the distributed Application objects after the Distribution has been extracted.

The Maintain Associations option is required when you distribute chained application information and folders. This is explained under "Chained Applications in Distributions" on page 294.

Continue with "Maintaining Application File Rights" on page 285.

Maintaining Application File Rights

File rights that you set in a golden Application object are not passed to the distributed Application objects, because file locations vary from server to server and cannot be anticipated.

The Desktop Application Distribution requires some manual processes, such as adding additional rights for file access. These processes are in addition to the minimums set by ZENworks when creating a distributed Application object. (The minimum rights might be enough for most applications.)

Review the following sections to understand how file rights are handled in Desktop Application Distributions:

- "File Rights Are Not Distributed" on page 285
- "File Rights and Groups" on page 286
- "Chained Applications and File Rights" on page 286
- "Setting File Rights" on page 286
- "Setting Trustees and Shares Instead of File Rights" on page 286

File Rights Are Not Distributed

File rights that are explicitly assigned in the Application object using the Rights to Files and Folders tab are not transferred, but are reset to the minimum necessary (Read and File Scan) for users to use the distributed applications. They are set when the distributed Application object is both created and then associated to a container or group.

File rights assigned in the Common > File Rights tab in the Application object are also not distributed.

You can later grant additional rights on these tabs in the distributed Application object and ZENworks does not remove or replace them.

File Rights and Groups

If a user or workstation is a member of a group that is distributed in the Desktop Application Distribution, then individual file rights for the user or workstation do not need to be set. The user or workstation obtains its rights to the application by virtue of its membership in the group.

Chained Applications and File Rights

If a chained application is used, all applications in the chain that require rights to a directory must be associated to a user or workstation group or a container in the golden Application's tree structure, because individual user or workstation objects' rights are not maintained in distributed Application objects.

Setting File Rights

To provide the Read and Write access rights to the files belonging to the chained application, in the Rights to Files and Folders tab in the User or Workstation Group object, assign the file rights.

Setting Trustees and Shares Instead of File Rights

Server Management does not set individual rights on files for NetWare-only trustees are set on the directories that contain the files, and rights are always Read and File Scan. Therefore, on NetWare servers you should grant users Read rights to the directory where the application's files are distributed. For example, if you have the files copied to the \apps directory, users would need Read rights to the \apps directory in order to use the application whose files were copied there.

Server Management also does not set file rights in Windows. Therefore, you should set up individual shares for users to have access to the application's distributed files.

Subscriber Working Context Conflicts

Whether your Subscribers all use the same working context or a unique working context depends on your application distribution design needs. You might have all of the Subscribers who receive a Desktop Application Distribution use the same working context if you want load balancing or fault tolerance to be used. For more information, see Section 6.1.1, "The Purpose of Desktop Application Distributions," on page 279.

Where there are multiple Subscribers using the same working context, an eDirectory collision is possible. In other words, multiple Subscribers cannot extract their copies of the same Desktop Application Distribution at the same time to the same working context in the tree.

For example, if two Subscribers extract an application at the same moment and create an Application object in two different eDirectory replicas, this causes a problem in eDirectory synchronization. When eDirectory finds the two different objects, but with the same name and the same timestamp in the two different replicas, eDirectory resolves this by renaming one of the objects by appending a number to the collision object's name (for details, see TID 10062001 in the Novell Support Knowledgebase (http://support.novell.com/search/kb_index.jsp)).

If you use the same working context for a group of Subscribers, then you must make sure that each Subscriber's Extract schedule fires at a different time, allowing enough time between these schedules for extraction to be completed by a Subscriber before the next Subscriber begins extracting.

If you have each Subscriber use a unique working context, all Subscribers can then extract their copies of the same Desktop Application Distribution at the same time, and no eDirectory collisions occur.

If a Distribution is set to extract immediately, the same scenario can exist.

Continue with

Maintaining Source Paths

Many applications require supporting files, and the paths to those source files must be established in the Application objects in Desktop Management. This is known as the "source path."

This section applies to Desktop Application Distributions containing Application objects that use source paths. For applications that require only an executable file (such as `notepad.exe`), source paths are not required in their Application objects.

Review the following sections to understand how source paths are used in Desktop Application Distributions:

- "Source Path Usage in Server Management" on page 288
- "Purpose of the SOURCE_PATH Macro" on page 290
- "How the SOURCE_PATH Macro's Values Are Interpreted" on page 291
- "MSI Applications and Source Paths" on page 291
- "Defining a Variable in Server Management" on page 291
- "Using the Source Path Option in the Distribution" on page 292

Source Path Usage in Server Management

To show what happens with the attribute between the golden and distributed Application objects during the Desktop Application Distribution process, Table 6-4 lists where you can find and configure source paths in ConsoleOne, their purposes, how these locations are populated, and their distribution status.

Table 6-4 *Source Path Usage Information*

Source Path Name (Type)	Location in the Application Object	Purpose	How Populated	Distribution Information
SOURCE_PATH (macro[1])	Distribution Options > Application Files > Source column	Provides path resolution from the SOURCE_PATH macro.	From the SOURCE_PATH macro.	This is distributed for each application file listed under the Name column that uses it.
SOURCE_PATH (macro)	Common > Macros > Name column	Defines a source path (in the Value column) to be used by the Application object.	From entries that you make on this page when the distributed Application object is created.	This is distributed with modifications to fit the Subscriber's environment. If it is changed in the golden Application object, it is updated in the distributed Application object with the necessary modifications. This source path on the golden Application object should be kept stable, in order to avoid Novell Application Launcher distribution problems.

Source Path Name (Type)	Location in the Application Object	Purpose	How Populated	Distribution Information
Package Source List (box)	Common > Sources	Provides a list of source paths for the Load Balancing and Fault Tolerance properties to use.	From the SOURCE_PA TH macro, from each Subscriber using the same working context that receives and extracts the Distribution, and from any entries you make using the Add button.	Only the SOURCE_PATH macro's entry is duplicated by Server Management using the long (DNS) version of the path. The listed source paths must be either valid UNC paths or variables that resolve to valid UNC paths. Each listed source path points to a complete set of the application's files that are located on the Distributor server's file system. (The Distributor cannot gather its Desktop Application Distribution's files from other servers.) These actual source files pointed to by the source paths are overwritten every time the Distribution is rebuilt and sent again. This field on the distributed Application object is cumulative, and is not overwritten when the Distribution is re-sent. Its entries come from selecting Load Balancing or Fault Tolerance for the Subscribers receiving the Distribution that use the same working context, or your use of the Add button on the distributed Application object. However, if you make a change to the SOURCE_PATH macro in the golden Application object, that source path is updated in this list box and is inserted first in the list. The previous source path is not replaced, but is left in the list. It is no longer valid, and you can delete it.

Source Path Name (Type)	Location in the Application Object	Purpose	How Populated	Distribution Information
Source List (box)	Fault Tolerance > Fault Tolerance	Provides a list of servers that can provide redundancy in case a server being used for Novell Application Launcher work goes down. All source paths listed must point to identical application file sets; otherwise, the distributed applications can fail to be created correctly.	From each Subscriber using the same working context that receives and extracts the Distribution.	This information is not distributed from the golden Application object. You must populate this field by sending the Distribution to multiple Subscribers that are using the same working context.
Source List (box)	Fault Tolerance > Load Balancing	Provides a list of servers that can provide load balancing among them for doing Novell Application Launcher work. All source paths listed must point to identical application file sets; otherwise, the distributed applications can fail to be created correctly.	From each Subscriber using the same working context that receives and extracts the Distribution.	This information is not distributed from the golden Application object. You must populate this field by sending the Distribution to multiple Subscribers that are using the same working context.

[1] A "macro" in Desktop Management has the same functionality as a "variable" in Server Management.

Purpose of the SOURCE_PATH Macro

The SOURCE_PATH macro defines the source path in an Application object to where its application's files reside on the Distributor server's file system. This is where the Distributor accesses the application files for building the Desktop Application Distribution.

The SOURCE_PATH macro's value is used to create the location on the Subscriber server's file system where those application files are to be placed by the Subscriber when it creates the distributed Application object.

The information in the value of the SOURCE_PATH macro includes:

- Server identification (the Distributor server) in either the server name, IP address, or full DNS name
- Volume or drive on the Distributor server
- User-defined path information (if provided in the wizard)

◆ Application path information (selected in the wizard)

Some examples:

```
server1.novell.com\sys\apps\acrobat
server1.novell.com\n\apps\acrobat
```

This is resolved to a valid UNC path, such as:

```
\\server1\sys\apps\acrobat
```

If you include a macro (or variable) within the value of the SOURCE_PATH macro, Server Management does not resolve that embedded information. Server Management only resolves the SOURCE_PATH macro's value to a valid UNC path.

Mapped drive letters can also be used if a global policy variable is defined.

How the SOURCE_PATH Macro's Values Are Interpreted

Tiered Electronic Distribution searches variables to find a match with the golden Application object's source path. For example, if the source path in the golden Application object is `n:\apps\acrobat`, the following order is searched to find a match:

```
n:\
N:\
n:
N:
n
N
```

However, if the golden Application object's source path is `N:\apps\acrobat`, the following order is searched to find a match:

```
N:\
n:\
N:
n:
N
n
```

MSI Applications and Source Paths

The `.mst` files can be entered (on the MSI > Transforms tab) without specifying the file's full path, because Server Management uses the source path defined for the Application object to find these files when building the Desktop Application Distribution. However, this is only true when the `.mst` files are in the same directory as the MSI file.

Defining a Variable in Server Management

Historically, mapped drives have been embedded into an Application object as a means of launching that application from a mapped drive on the desktop. Server Management uses variables to distribute applications that use drive mappings.

Because volume names and mapped drives for the Distributor and all of the Subscribers receiving the Distribution can be different, variables allow you to identify these locations with a value that is interpreted by the Distributor and each Subscriber.

You can define variables globally and individually:

- **Globally using the Tiered Electronic Distribution policy**

  Variables defined in the Tiered Electronic Distribution policy (Service Location Package) are available to all Subscriber objects associated with the policy, such as associating the policy package to the parent containers of the Subscriber objects. For the policy to be in effect for each Subscriber, make sure on the Variables property page that the Include Policy check box is selected.

  The policy package must also be associated with the parent container of the Distributor object. The variable definition in the policy ensures that the Distributor knows where to gather the application files from.

  If both the Distributor and Subscribers use the same variable value, then only one Tiered Electronic Distribution policy is needed, and you can associate its Service Location Package to the parent containers of both the Distributor and the Subscribers.

  For example, the mapped drive source path for a golden Application object is `n:\applications\acrobat` and you want n: to represent the `sys:\public` directory on the Distributor server. To create the variable, in the Tiered Electronic Distribution policy select the Variables tab, then enter the n: for the variable and `sys:\public` for its value. Then, the Distributor can find the `\applications\acrobat` directory on its `sys:` volume when it needs to build the Distribution.

  For more information on this policy, see "Tiered Electronic Distribution" on page 213.

- **Individually in each target Subscriber's properties**

  You can define a variable for any Subscriber object. This definition overrides the same variable if it is defined in a Tiered Electronic Distribution policy that the Subscriber is associated with.

  This is useful for when the Subscriber server's volume name or mapped drive is different than the Distributor server's (so they can't use the same Tiered Electronic Distribution policy), or you have a variety in volume names or mapped drives among the Subscribers receiving the Distribution.

  For information on how to define variables on Subscribers, see Section 9.3, "Defining a Variable," on page 355.

Using the Source Path Option in the Distribution

If a golden Application object uses mapped drives, enable the Keep the Same Source Paths for the Replicated Objects option when running the Desktop Application Distribution Wizard. Enabling this option causes the Distribution to retain golden application source paths for when a mapped drive designation must be used by the application that is distributed. The value of the mapped drive determines where the application files are copied.

If the golden application source paths are mapped drives, but you want the distributed applications to use a UNC path according to the extraction directory, then you do not need to select this option, but you must define the Tiered Electronic Distribution policy in the Service Location Package with variables defined for the mapped drives. This package must be associated to the Distributor with the variable defined in order for the Distribution build to work. It should also be associated with containers for the Subscriber objects, or any container above them, if they have the same drive mappings.

Key points about this option:

- If a golden Application object's Package Source List box contains a mapped drive, you must enable the Keep the Same Source Paths for the Replicated Objects option. The mapped drive letter is treated like a variable that needs to be resolved on both the Distributor and Subscriber to complete the valid UNC path.

- If a golden Application object's Package Source List box contains a drive mapping that is local to a server other than the Distributor server, no application files can be gathered or distributed, because all files to be included in the Distribution must be contained on the Distributor server's file system.

- Enabling this option affects all Application objects in the Distribution, including chained applications. Therefore, all mapped drive properties for each of the Application objects included in the Distribution are distributed to keep their golden application source paths, and each application and chained application must have mapped drives for the source path.

- If you select this option, only the Application object's Default Directory Path is used, because the Application Destination Directory Path field in the next wizard page is disabled. Therefore, you cannot change the path.

- When you select this option, or if you leave it unselected, that choice becomes the permanent use of this option for the Distribution. This is done to prevent problems that can occur from alternating between using and not using an Application object's mapped drives.

- For chained applications, source paths are treated the same for all chained applications as they are for the first application that the others are chained to.

6.1.3 Miscellaneous Issues

- "Application Dependencies and Requirements" on page 293
- "Chained Applications in Distributions" on page 294
- "Extended Characters in Directory Paths" on page 294
- "Tree to Tree Distributions" on page 295
- "Site Distribution Objects" on page 295

Application Dependencies and Requirements

Dependencies and requirements can be confusing with regard to the distribution of attributes:

- **Dependency:** An application dependency, such as a chained application, is updated in a distributed Application object when the Distribution is rebuilt, sent, and extracted.

  To view application dependencies: in the properties of an Application object, click Run Options > Application Dependencies.

- **System requirement:** A system requirement, such as an operating system for the application to run on, is updated in a distributed Application object when the Distribution is rebuilt, sent, and extracted.

  To view system requirements: in the properties of an Application object, click Availability > Distribution Rules.

  One exception is that an application requirement is not updated in a distributed Application object when the Distribution is rebuilt, sent, and extracted. Instead, we recommend using application dependencies.

Chained Applications in Distributions

If multiple applications contain the same chained application, the application's files are only contained once in the Distribution. This reduces the Distribution's file size.

For example, if you distribute several icons (each its own Application object) that each require an office software suite, that suite software is only included once in the Distribution.

If your Desktop Application Distribution has chained applications, you must enable the Maintain Associations option when configuring the Distribution.

Chained applications in Distributions are only available in ZENworks for Desktops 4.*x* and later.

For more information on understanding and setting up chained applications, see "Advanced Distribution: Configuring Application Dependencies and Chains" in the *Novell ZENworks 7 Desktop Management Administration Guide*.

Extended Characters in Directory Paths

If extended characters (such as ê, ë, ì, or í) exist in the path to the `.fil` files for an AOT Application object, you must define a code page variable for both the Distributor and its Subscribers.

The code page variable is necessary for the Distributor so that it can gather the applications files from its file system when it builds the Distribution, and it is necessary for the Subscribers so that they can successfully copy the application files from the Distribution to their file systems. In other words, the code pages used by the Distributor and Subscribers must contain the extended characters used in the paths contained in the Distribution.

To create the code page variable:

1 Determine the code page used by ConsoleOne for the international characters to be used in the Distribution.

  The code page must come from the workstation used to create the golden Application objects that have extended characters in the paths to their AOT files.

  You can use the `encoding.cmd` utility included in the `\codepageutility` directory on the *ZENworks 7 with Support Pack 1 Companion 2* CD to determine the necessary code page. Instructions on how to use this utility are contained in the `readme.doc` file included in the `\codepageutility` directory.

2 In ConsoleOne, create a Service Location Package and access its properties.

3 Click the *Tiered Electronic Distribution* policy to enable it, then click *Properties*.

4 Add the following variable in the policy: CODE_PAGE.

5 Enter the desired code page as the variable's value.

  For example, Cp1252.

  Code page values are case sensitive.

6 Click *OK* to save the policy.

7 Associate the policy package to the container of the Distributor object.

8 To provide Subscribers access to the code page, do one of the following:

   ◆ Associate the policy package to the container of the Subscriber objects receiving the Distribution.

      This causes the variable to be available for use by the Subscribers, providing access to the code page.

   ◆ Define the same code page variable (as in Step 4 and Step 5) in each Subscriber object's properties.

      This provides Subscribers access to the code page.

9 Rebuild the Distribution, if it has already been created.

Tree to Tree Distributions

You can send Desktop Application Distributions to other trees. However, ZENworks 7 Distributions cannot be sent to ZENworks for Servers 3.0.2 Subscribers because of new schema extensions for ZENworks 7 and later.

Site Distribution Objects

ZENworks 7 Server Management does not use a Site Distribution object. Previous versions of ZENworks might have used Site Distribution objects with this Distribution type.

6.2 Requirements

The following requirements must be met before creating and sending Desktop Application Distributions using Tiered Electronic Distribution:

❑ In order to use Novell Application Management with ZENworks 7 Server Management, you must have ZENworks for Desktops 4.01 or ZENworks 7 Desktop Management or later installed. Chained applications in Desktop Application Distributions are only supported in ZENworks for Desktops 4.01 and later.

❑ Desktop Management and Server Management must both be installed to the same tree, including their respective schema extensions.

❑ For golden Application objects to be functional in a Desktop Application Distribution, the snap-ins for both Server Management and Desktop Management must be installed in ConsoleOne.

❑ Make sure all of the associations in the golden Application object are in the source root context or below.

   IMPORTANT: If even one of your associations is outside the source root context, the Distributor fails to build the Distribution.

❑ For Windows and Linux servers, you must have a shared location established for extracting the Distribution's files, where all users can have access to those files.

❑ The source path must point to application files that are located on the Distributor server's file system, because the Distributor cannot gather files from other servers' file systems.

   If the Package Source List box contains a local drive mapping, no application files are gathered or distributed.

If the Package Source List box contains a mapped drive, Keep the Same Source Paths for Replicated Objects must be selected. The drive letter is treated like a variable that needs to be resolved on both the Distributor and Subscriber to complete a valid UNC path.

Use a policy to define the variables on a Distributor. On the Subscriber you can use either the variable list in the Subscriber object, or a policy that is associated to the container where the Subscriber object resides.

❑ The Subscriber object must have the Working Context attribute defined. This is the eDirectory context where the Subscriber creates the objects related to the Desktop Application Distributions that it receives.

Multiple Subscribers can use the same working context if you intend to use them for load balancing or fault tolerance.

❑ If extended characters (such as ê, ë, ì, or í) exist in the path to the `.fil` files for an AOT Application object, you must define a code page variable for the Distributor and Subscribers. For more information, see "Extended Characters in Directory Paths" on page 294.

❑ Determine whether you are using Samba or NCP shares for client access to files on Linux or OES Linux servers, then set up that access. For more information, see Appendix G, "Client Access in Linux," on page 435.

6.3 Creating a Desktop Application Distribution

- Section 6.3.1, "Understanding the Desktop Application Distribution Wizard," on page 296
- Section 6.3.2, "Creating the Distribution," on page 297

6.3.1 Understanding the Desktop Application Distribution Wizard

ZENworks uses Tiered Electronic Distribution to distribute Application objects to other locations in the same tree or other trees. Using a Desktop Application Distribution, the original files associated with the applications are copied to the appropriate server locations where they can be used to locally service user groups and workstation groups associated with the distributed Application objects.

To distribute applications, you use the Server Management Desktop Application Distribution Wizard to configure the Distribution. This includes:

- Determining the destination's tree context
- Determining whether to maintain the associations between user/workstation groups or containers and the applications
- Determining whether to have automated load balancing or fault tolerance
- Determining how to trigger rebuilds of the Distribution
- Selecting the applications
- Determining the file copying paths

To create a Desktop Application Distribution using the wizard, continue with Section 6.3.2, "Creating the Distribution," on page 297.

6.3.2 Creating the Distribution

IMPORTANT: You can also perform these step in iManager instead of running this wizard in ConsoleOne. Instructions are contained in the context-sensitive help in iManager.

1 Fulfill all of the requirements listed under Section 6.2, "Requirements," on page 295, including creation of a code page variable if necessary.

2 In ConsoleOne, right-click the container where you want the Distribution object located, click *New*, click *Object*, select *TED Distribution*, then click *OK*.

3 Provide a name for the Distribution.

IMPORTANT: Periods (.) are not allowed in Distribution names. Instead, use dashes (-) or underscores (_) as word separators. If you use a period in the Distribution name, the Distribution is not sent, and the Distributor is not reloaded after it has been exited.

4 To give a Distributor ownership of the Distribution, browse and select the Distributor object, click *Define Additional Properties*, then click *OK*.

The Distribution object's properties are displayed.

5 Select the *General* tab and fill in the *Settings* fields:

Active: Required. In order to make a Distribution available to Subscribers, it needs to be active.

Use digests: Digests are used by Distributors and Subscribers to verify that Distributions have not been tampered with while in transit. The digest provides an MD5 checksum for the Subscriber to compare.

Digests also detect corruption in a Distribution's package. If corruption is present, the Subscriber renames the `distfile.ted` Distribution file to `distfile.corrupt` and the Distribution is rebuilt and sent the next time the Channel's schedule fires.

Encrypt: You can have the Distribution encrypted if you are sending it across non-secured connections. Encryption provides security for the Distribution during transit between the Distributor and Subscriber when they are not within the same firewall. Select either Strong or Weak encryption.

You must also have NICI 2.6.4 or later installed to each of these servers for encryption to work (see "Installing NICI 2.6.4" on page 54). Older versions of NICI are not compatible with version 2.6.4 or later.

Maximum revisions: This number helps you to control disk space usage by determining how many versions of a particular Distribution are kept in the Distributors' and Subscribers' working directories. The default is 10. Select *Limited* and enter a number.

Increase the number if data is changing often and the changes are minimal (smaller delta files). Decrease the number if data is not changing very often, or if a significant amount of data is changing (larger delta files).

The following e-mail options are available if you set a maximum number. If you select *Unlimited*, these options are dimmed.

- **Approaching maximum revision email notification list:** Contains the e-mail addresses of anyone who is to be notified when a Distribution is approaching the maximum revisions set in the *Maximum revisions* field. Here, you can either remove a single or all displayed addresses.

- **Email address (maximum revision notification):** You can add e-mail addresses to the list in *Approaching maximum revision email notification list*. Just enter an e-mail address and click *Add* and it is displayed in the listing.

- **Send notifications when Distribution revision is ___ or less of reaching maximum revisions:** Enter a number to indicate how close "approaching" is. When the current revision number of Distribution plus this number equal the maximum revisions, an SMTP notification is sent to the listed addressees.

  SMTP must be configured and its e-mail server address listed in the next field.

- **Email server address:** The SMTP server used to send the e-mail notifications. For example, mail.novell.com.

  For information on configuring SMTP e-mail notifications, see "SMTP Host" on page 212.

Priority: You can give the Distribution a priority that determines how it is sent in relation to other Distributions. A High priority means it is sent before Medium or Low priority Distributions.

Distributor: Displays the DN of the Distributor object that builds and sends this Distribution. You selected the Distributor when you created the Distribution object.

Description: Provide useful details about the Distribution, such as the name of the desktop application, the files and directories it contains, intended user groups, and so on.

6 Click *General > Restrictions*.

You can select whether to have platform restrictions for the Distribution itself. This is not a restriction for the distributed Application object.

No restrictions: This option is selected by default. To determine platform restrictions, select this option to disable it, then select the check boxes corresponding to the platforms you want to receive this Distribution.

Platforms with check boxes not selected cannot receive the Distribution. In other words, you restrict sending to a platform by deselecting the No Restrictions option and not selecting the platform.

The available options are:

    No Restrictions
    NetWare All
    NetWare 4.*x* (earlier versions of ZfS supported these platforms)
    NetWare 5.0 (earlier versions of ZfS supported this platform)
    NetWare 5.1
    NetWare 5.*x* (earlier versions of ZfS supported these platforms)
    NetWare 6.*x*
    Windows Server

No Restrictions means that the Distribution can be sent to any of these platforms.

If you select NetWare All, you do not need to select any of the individual NetWare platforms.

7 Select the *Type* tab, select Desktop Application in the Select Type drop-down box, then click *Setup*.

The Desktop Application Distribution Wizard is started. iManager provides its own interface in place of this wizard.

You can also start this wizard from the Desktop Application Agent properties page by clicking *Modify*.

7a Click *Next* after reading the introductory information.

7b Fill in the fields, then click *Next*.

Maintain source tree structure: Duplicates the source tree's structure at the destination's location (the target Subscriber's working context) for placing the distributed Application objects. If you are selecting chained applications, you must select this option.

For more information, see "Maintaining Source Paths" on page 287.

Source root context: Select a container to be used as the root container for the golden Application objects to be distributed. You should select golden Application objects only from this root container and its subordinate containers.

Maintain associations: Distributes the associated groups or containers at the target location if they do not exist. However, users or workstations contained in the groups or containers in the source location are not distributed.

For more information, see "Maintaining Associations When Distributing Objects" on page 284.

You must enable this option if you have chained applications in the Distribution. For more information, see "Chained Applications in Distributions" on page 294.

IMPORTANT: Rights previously set in the associated user/workstation groups or containers that are maintained are not distributed, but set to the minimum necessary in the distributed groups or containers so that users can use the applications.

Overwrite Existing Target Folder Object Attributes: When selected, this check box causes existing target folder objects to be overwritten with the relevant content of the source folder objects, meaning all previous folder associations for the application are replaced by the new folder associations.

To retain previous folder associations while adding new folder associations, deselect this option. (This is the default function previous to ZENworks 7 for how folder objects are handled. The option to overwrite is added in ZENworks 7.)

Always replicate association flags: Causes the launch configuration flags for each group or container associated with the golden Application object to be replicated with each distributed application.

Load balance and fault tolerance support: Choose whether to use automated load balancing, fault tolerance, or neither:

  ◆ **Load balance:** Automates spreading server workloads over the servers being used for the Desktop Application Distributions. The functionality of fault tolerance (redundancy) is automatically accomplished through load balancing.

  ◆ **Fault tolerance:** Allows a server being used for Desktop Application Distributions to assume the distribution duties of another server that goes down. Fault Tolerance does not provide load balancing.

- **None:** Neither option is applied. You must individually configure each distributed Application object for load balancing or fault tolerance, if you want that support on an individual basis.

For these two features to work:

- Multiple Subscribers receiving the Distribution must be using the same working context

- The User Source List button must be selected on the Fault Tolerance > Fault Tolerance or the Fault Tolerance > Load Balancing properties pages of the Application object

Depending on the selected options, the Load Balancing or Fault Tolerance pages are populated with the file locations on all servers that share this working context.

For more information, see "Distributed Applications in Server Management" on page 280.

Rebuild only if any application version number changes: Allows you to control Distribution rebuilding based on the Build schedule. Select this check box to withhold modifications to a golden Application object until you are ready to release them.

Regardless of the status of this check box, if applications are added to or removed from the Distribution, it is rebuilt according to its Build schedule.

For more information, see Section 6.4.2, "Triggering a Rebuild," on page 303.

7c Click *Add* to browse for and select golden Application objects.

Do not browse above the root directory that you established in the previous wizard page, especially if you have selected the Maintain Source Tree Structure option.

IMPORTANT: The Desktop Application source files must reside in the Distributor server's file system. The Distribution cannot be gathered from another server's file system.

7d Select the *Keep the same source paths for the replicated objects* option if you want to retain the golden Application object's source path when the mapped drive feature is used in the distributed application.

For more information, see "Maintaining Source Paths" on page 287.

7e Provide the destination volume or shared folder.

The application files distributed are those that are associated with the golden Application objects you selected in the previous wizard page.

You can provide a variable instead. If you use a variable, it must be defined in the destination Subscriber server's properties to point to the target server's volume or shared folder.

This volume (NetWare) or shared folder (Windows and Linux) becomes the root location for placing subordinate directories where the application files are copied.

7f To use only an application's default path, click *Application's default directory path*, which is placed beginning with the root location you specified in Step 7e.

or

To provide a user-defined directory path to the application's files, click User-Defined Directory Path, then provide your path information.

The path you specify is used in the following manner:

◆ The volume or shared folder name remains unchanged (as specified in Step 7e).

◆ Your path information is inserted after the volume or shared folder name.

◆ Part of the application's default path is appended to your path information, beginning with the default path's immediate parent directory to the application's files. Any default path information that was above the immediate parent directory is replaced by your path entry.

The result is a customized directory path that begins with the volume or shared folder, has your user-defined path information next, and ends with the application's immediate directory. For example, suppose the default path to the application's executable file (`application.exe`) might be:

`\application_root_directory\application_subdirectory`

and you enter `\mypath` for your user-defined path; the new full path to the executable is now:

`c:\mypath\application_subdirectory\application.exe`

If you entered C: as the shared folder and `\mypath` as your user-defined path, `\application_root_directory` is replaced by `\mypath`, and `\application_subdirectory` is the immediate parent directory to `application.exe`.

7g Click *Next* to continue.

The Summary page is displayed.

7h To make changes, click *Back*.

7i When you have finished configuring the Distribution object, click *Finish* to exit the wizard.

You can edit the Distribution at any time on the *Type* tab of the Distribution object by clicking *Modify*.

8 Select the *Channels* tab, click *Add*, then browse for and select the Channel for this Distribution.

Each Distribution must be associated with at least one Channel if it is going to be used to push data to a Subscriber. A Distribution is sent to all Subscribers that are subscribed to the selected Channel.

9 Select the *Schedule* tab, then select a Build schedule:

Section B.1, "Daily," on page 410
Section B.3, "Interval," on page 410
Section B.4, "Monthly," on page 411
Section B.5, "Never," on page 411
Section B.8, "Run Immediately," on page 412
Section B.9, "Time," on page 413
Section B.11, "Yearly," on page 413

10 Click *Apply* to create the Distribution.

You are prompted to copy additional security certificates.

11 Select *Yes* to resolve the certificates.

This copies the security certificates from the Distributor to Subscriber subscribed to the Channel.

For information, see Section 7.1.6, "Resolving Certificates," on page 311.

12 Click *OK* to close the Distribution object.

The next time the Distributor reads eDirectory (this schedule is set in the Distributor object's properties), it retrieves all of the information about the new Desktop Application Distribution, such as Distribution details, the Build schedule, and so on.

The Distribution is built according to the Build schedule, sent according to the schedule set in the Channel object, and extracted according to schedule set in the Subscriber object.

If the Distributor throws an exception during the file gathering process, the Distribution is not built. The Distributor logs the failure in the reporting database.

If the Subscriber throws an exception during extraction, the process is not completed. The Distributor receives this information from the Subscriber and logs the failure in the reporting database.

After extraction, Desktop Management users whose objects are located in the associated containers, or are members of a distributed group, will have access to the desktop applications that were distributed.

IMPORTANT: For Desktop Application Distributions, a built-in delay exists to accommodate directory synchronization when you have multiple applications being distributed at the same time (whether by one or multiple Distributions).

Subscribers can receive Desktop Application Distributions all at the same time, but extract them one at a time. And, when there are multiple applications contained in one Distribution, the Subscriber creates the distributed Application object and copies the files one application at a time. The built-in delay helps directory synchronization for the newly-created Application objects to occur smoothly.

To determine how much additional time this built-in delay might add to the distribution process, multiply each application contained in a Desktop Application Distribution by 30 seconds.

As a rule of thumb, if an application being distributed includes multiple versions, such as one baseline and two deltas, each of these three versions receives the same 30-second delay. For example, if you are sending 10 desktop applications, and each has three versions, the completion of the Distribution extractions could take at least 15 minutes.

---

13 For Desktop Management users and workstations to have automatic access to their applications from any geographic location, you must link up the site lists:

13a Wait for the Desktop Application Distribution to be distributed and extracted by each Subscriber server that received it, because the distributed Application objects must be created and the application's files installed before you can link up the site lists.

13b In ConsoleOne, right-click the golden Application object that was used to build the Distribution, then click *Properties*.

13c Click the *Distributions* tab, then click the *Link Up Site Lists* button.

All distributed Application objects that were created from the golden Application object are displayed in the *Replicated Applications* list box, and all Distributions containing the distributed Application objects are listed in the *Distributions Currently In* list box.

The *Link Up Site List* button does the following:

13c1 For the golden Application object, it searches for each distributed Application object that was created from the Distribution and lists the full DN of those objects in the golden Application object's properties (in the Replicated Applications list box).

13c2 For each distributed Application object, it searches for the other distributed Application objects that were created from the Distribution; for the golden Application object, it lists the full DN of all of these objects in the distributed Application object's properties (in the Replicated Applications list box).

13c3 Step 13c1 and Step 13c2 are repeated for each distributed Application object.

13c4 For each Application object listed in the Replicated Applications list box, any Distributions associated with those objects are listed in the Distributions Currently In list box in each of these Application objects.

Thus, the golden Application object and all distributed Application objects have each other listed in their Replicated Applications list box, which allows users to have local access to the same application no matter where they connect to their network.

13d Click *OK* to close the golden Application object's properties.

13e Repeat Step 13b through Step 13d for each golden Application object that was used to build the Desktop Application Distribution.

You need to perform the site list link-up only on the golden Application objects.

6.4 Rebuilding Desktop Application Distributions

The following sections explain the different issues with rebuilding Desktop Application Distributions, including how to trigger a rebuild:

- Section 6.4.1, "All Attributes Are Updated," on page 303
- Section 6.4.2, "Triggering a Rebuild," on page 303

6.4.1 All Attributes Are Updated

All attributes contained in a golden Application object, not just the modified attributes, are updated in the distributed Application objects when a Distribution is rebuilt, sent, and extracted. This means that if you make a change to an attribute in a distributed Application object, such as a source path, that source path is overwritten by the source path data in the golden Application object. In other words, all distributed Application objects are kept in sync with their golden Application object. Exceptions to this are described in "Maintaining a Golden Application's Attributes" on page 282.

A rebuilt Desktop Application Distribution includes all file changes made after the last time the Distribution was built.

6.4.2 Triggering a Rebuild

You can control when a Distribution is rebuilt by whether you select the Rebuild Only If Any Application Number Changes check box in the Desktop Application Distribution Wizard:

- "Selecting the Check Box" on page 304

- "Leaving the Check Box Disabled" on page 304

Selecting the Check Box

This feature is useful for withholding modifications to a golden Application object until you are ready to release them.

The Distribution is rebuilt according to its established Build schedule, but only after you have manually incremented the Version Number field in the golden Application object, or its dependent application, and the Distributor has read eDirectory to discover the Version Number field change.

If there are multiple applications in a Distribution, a version number change in only one of them triggers a rebuild of the Distribution for all of them.

The Version Number field is on the Distribution Options > Options tab of the Application object's properties.

Regardless of the status of this check box, it is rebuilt according to its Build schedule if applications are added to or removed from the Distribution.

Leaving the Check Box Disabled

If you do not select this option (it is unchecked by default), the Distribution is rebuilt according to its established Build schedule. In this case, there can be two scenarios:

- "Modifying an Object" on page 304
- "Removing a Distributed Application Object" on page 304

Modifying an Object

When you modify a Distribution object or one of its golden Application objects, its internal revision number is automatically changed, which triggers a rebuild of the Distribution according to its established Build schedule.

Modifications include adding or removing applications from the Distribution. However, if you simply update, add, or remove application files in the Distributor server's file system, this does not alter the internal revision number of the Desktop Application Distribution object. The ZENworks file synchronization feature does not apply to the files in Application objects. Therefore, no rebuild is triggered.

If you add, remove, or update any files belonging to a golden Application object, those changes are included when the next rebuild is triggered.

Removing a Distributed Application Object

Removing a distributed Application object causes a backlink to the golden Application object to change without any other changes being made to the object. This causes the internal revision number to change on the golden Application object, which triggers a rebuild of its Distribution according to the established Build schedule.

6.5 Cleaning Up Desktop Application Distribution Files

Tiered Electronic Distribution is not designed to clean up Desktop Management files, for example, when you might delete a golden Application object in Desktop Management. The distributed Application object created by the Desktop Application Distribution is not automatically deleted. You must manually remove this eDirectory object and the related application files on the server's file system.

To do this, search in ConsoleOne for the Application object and note its filename before deleting the golden Application object. This can make manually cleaning up easier after deleting a golden application.

You can verify that the distributed application exists after you've deleted the golden Application object and its files, and then remove the distributed version:

1 In ConsoleOne, delete the golden Application object.

2 In a file browser, delete the files related to the application.

3 In ConsoleOne, right-click the Distributor object associated with the golden Application object's Distribution.

4 Click *Refresh Distributor* from the drop-down menu, then click *Yes > OK*.

5 Check the destination location for the distributed Application object.

   It should still be present, even though it no longer exists in the Distribution.

6 In ConsoleOne, delete the distributed Application object.

7 In a file browser, delete the files related to the distributed application.

6.6 Sending Desktop Application Distributions Tree-To-Tree

Desktop Application Distributions can be sent between trees. However, you must do the following for this to work:

1 Create an External Subscriber object in the Distributor's tree that points to the target server in the other tree where you want to send the Desktop Application Distribution.

   This enables the Distributor server to send the Distribution directly to the target server using the IP address listed in the External Subscriber object.

2 Make sure the target server that is to receive the Desktop Application Distribution has a Subscriber object in its own tree, so that it has the rights to eDirectory for creating the distributed Application object in that tree.

   This Subscriber must be within a working Tiered Electronic Distribution system.

3 Set the working context in the Subscriber object for the target server, if this was not done during installation.

   If the working context is not set for the target server, authentication fails during the extraction process.

4 Create the Desktop Application Distribution (see).

Defining the Desktop Application Distribution is the same process, whether it is being sent within a tree or across trees.

5 Add the External Subscriber object to the Channel where the Desktop Application Distribution is listed.

6 Manually copy the certificates to the Subscriber in the target tree.

or

If you have a mapped drive, browse to the correct path when prompted.

7 Send the Distribution.

Security in Policy and Distribution Services

7

Novell® ZENworks® Server Management provides the following types of security for Policy and Distribution Services:

7.1 Distribution Security Using Signed Certificates and Digests

There are two features of Tiered Electronic Distribution that deal with security:

- **Certificates (required):** Security certificates are issued by each Distributor to all Subscribers receiving its Distributions to validate whether Distributions are from a trusted source, or have been tampered with. This security is automatically used by Policy and Distribution Services for all Distributions. However, there are actions you might need to take to get Policy and Distribution Services to create and process the certificates.

  In order for a Subscriber to accept its first Distribution from a Distributor, it must have a certificate from that Distributor in its \security directory. After receiving its first Distribution from the Distributor, the certificate is first stored in the .keystore file, then the certificate is deleted from the \security directory.

  The .keystore file is a repository of signed certificates from the various Distributors who send Distributions to the Subscriber. In other words, it provides the Subscriber with an accumulation of trusted sources for its Distributions.

  You can view the content of the .keystore file in Novell iManager.

  For information on security certificates for encrypted Distributions, see Section 7.2, "Distribution Security Using Encryption," on page 317.

- **Digests (optional):** You can have digests created for each Distribution at the time it is built. The digest provides an MD5 checksum for the Subscriber to compare against to determine whether a Distribution has been tampered with after it left the Distributor.

  Digests also detect corruption in a Distribution's package. In the case of corruption, the Subscriber renames the distfile.ted Distribution file to distfile.corrupt and the Distribution is rebuilt and sent the next time the Channel's schedule fires.

The following sections provide more information on understanding, creating, and using certificates and digests:

7.1.1 Understanding Digests

Important points about digests:

- Digests can be created for each Distribution at the time it is built. The digest is used by the Subscriber to determine whether a Distribution has been tampered with after it left the Distributor.

- Digests detects corruption in a Distribution's package. In the case of corruption, the Subscriber renames the `distfile.ted` Distribution file to `distfile.corrupt` and the Distribution is rebuilt and sent the next time the Channel's schedule fires.

- The Digest option is available for all Distribution types. The Digest check box is displayed on the General tab of the Distribution object's properties.

- A digest adds to the build time. Factors that can affect build time using digests are CPU and hard drive speeds, amount of RAM, server workload, and so on.

7.1.2 Understanding Certificate Usage in Policy and Distribution Services

A certificate is a security mechanism used by Policy and Distribution Services to ensure that the Distribution received by a Subscriber was actually sent by the Distributor owning that Distribution. Because configuration information can also be sent to the Subscriber, it ensures that the configuration information has been sent from a known Distributor and that the data has not changed.

All Subscribers must receive a valid security certificate from each Distributor that sends Distributions to them. Without a matching certificate, a Subscriber cannot receive Distributions from the Distributor.

Figure 7-1 illustrates the process of using certificates with Distributions:

Figure 7-1 *Resolving Certificates*

Sending Regular Distributions
Resolving Certificates

Distributor

Subscriber

| Distributor's Certificate | → | Resolve Certificates | → | Subscriber's Certificate |

Copy of Distributor's
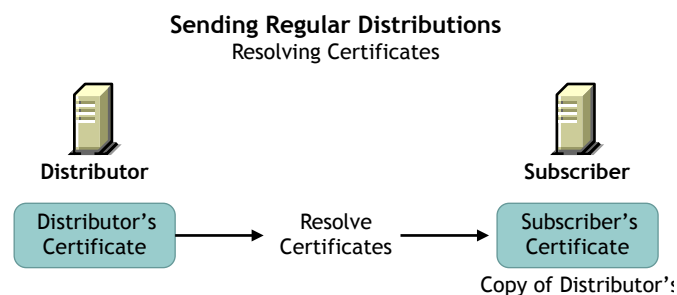
Before a Distribution is sent, certificates must be resolved. This ensures that the Distribution received by a Subscriber was actually sent by the Distributor owning that Distribution.

For information on resolving certificates, see Section 7.1.6, "Resolving Certificates," on page 311.

After certificates have been resolved, the following illustrates how the Subscriber uses the certificate to ensure it is receiving a valid Distribution, as illustrated in Figure 7-2:

Figure 7-2 *Sending the Distribution*

Sending Regular Distributions
Sending the Distribution

Distributor

Subscriber

Distribution Built → Send Distribution

Distributor's Certificate ← Subscriber verifies Signature with Distributor's Certificate ← Subscriber's Certificate

Copy of Distributor's

Signature Matches → Distribution Received And Extracted
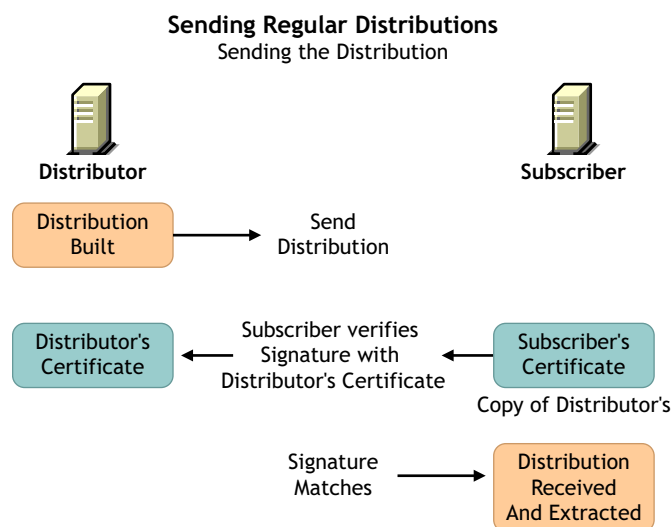
7.1.3 Important Points about Certificates

- Certificates are issued by each Distributor to all Subscribers receiving Distributions from that Distributor. In order for a Subscriber to accept Distributions from a Distributor, it must have received a certificate from that Distributor.

- For security, certificate key pairs are created by the Distributor.

- The public key is written to the Distributor server's file system, which self-signs a certificate and stores it in Novell eDirectory™.

- The private key is stored in the Distributor object's properties and is used for encryption.

- The Subscriber software does not need to be running on the Subscriber server to have certificates copied to the server.

- The association of Distributions (owned by a Distributor) and Subscribers to a Channel determines which Subscribers should receive certificates from which Distributors.

  - A Distributor sends certificates to all Subscribers that subscribe to Channels where the Distributor has Distributions.

  - A Subscriber requests certificates from all Distributors that have Distributions in Channels to which it subscribes.

- A certificate can be passed from a Distributor to a Subscriber under the following circumstances:

  - When a Subscriber is initially subscribed to a Channel and you click OK to apply the changes.

  - When you right-click a Subscriber Object and select Resolve Certificates. The Subscriber then requests certificates from all Distributors that it receives Distributions from.

  - When a Distribution is listed in a Channel and you click OK to apply the changes.

  - When you right-click a Distributor Object and select Resolve Certificates. The Distributor sends certificates to all Subscribers that it sends Distributions to.

    For information on resolving certificates, see .

  - When you add a Distribution or a Subscriber to a Channel. When you click OK, the Resolve Certificates? dialog box is displayed. If you answer Yes, certificates are sent by all Distributors who have Distributions associated with that Channel to all Subscribers subscribed to that channel.

  - Manually copying a certificate file to a transfer medium (such as a diskette or local drive), then to the `\zenworks\pds\ted\security` directory on a server.

  Basically, any time the relationship changes between the Subscribers, Channels, or Distributions, a certificate can be passed.

- If a Distributor object is deleted and re-created to point to the same server, all certificates on the subordinate Subscribers become invalid. Certificates must be deleted from the Subscriber's `\security` directory, then the Distributor must send the new certificates to those Subscribers.

- ConsoleOne copies the certificate files to Subscriber servers. Therefore, the client software on the workstation running ConsoleOne must have access to the Subscriber servers' file systems. For Windows Subscriber servers, the Domain and Workgroup rights on the workstation must be set up to facilitate automatic certificate copying. Otherwise, a 1204a error is given.

7.1.4 ConsoleOne User Rights and Certificate Copying

The administrator using ConsoleOne® must have sufficient rights to the Subscriber server in order for a certificate to be copied to that server when the administrator resolves certificates in ConsoleOne. This is because when you use ConsoleOne to configure a Subscriber object to receive the Distributions from a particular Channel, the Distributors owning the Distributions in that Channel must send certificates to the Subscriber's server.

For NetWare® Subscribers, the ConsoleOne user automatically has sufficient rights by virtue of being able to configure the Subscriber object.

For Windows Subscribers, administrator rights for the ConsoleOne user must be set up in Windows by selecting Active Directory Users and Computers, or selecting Local Users and Groups.

7.1.5 Certificate File Locations

Certificates are stored in the `\zenworks\pds\ted\security` directory on NetWare and Windows Subscriber servers, or in the `/var/opt/novell/zenworks/zfs/pds/ted/security` directory on Linux and Solaris servers.

WARNING: Make sure the `\security` directory is a non-public directory. This directory should not be read by anyone other than an administrator. The `.keystore` file is in the `\security\private` directory and is by default hidden from non-administrative users.

Certificates are usually named after the fully qualified DNS name of the Distributor server, such as `Distributor_Server001.Distributions.ZENworks.Novell.com.cer` or `Distributor_Server001.Distributions.ZENworks.Novell.com.csr`. The TCP/IP address of the server would be used for `.csr` files if a DNS name could not be resolved. The certificate would then be named using its IP address, such as `155.55.155.55.csr`.

7.1.6 Resolving Certificates

IMPORTANT: ConsoleOne copies the certificate files to Subscriber servers. Therefore, the client software on the workstation running ConsoleOne must have access to the Subscriber servers' file systems. For Windows Subscriber servers, the Domain and Workgroup rights on the workstation must be set up to facilitate automatic certificate copying. Otherwise, a 1204a error is given.

When you are automatically presented with the option in ConsoleOne to resolve certificates, determine the following to know whether to select Yes or No:

- If the Distributor currently has Distributions associated with this Channel, and all Subscribers currently subscribed to the Channel have previously received a certificate from this Distributor, select No.

- If this is the first Distribution added to this Channel by the Distributor, or a Subscriber has been newly added to the Channel, select Yes (to resolve certificates).

  This copies the security certificates from the Distributor to the Subscribers subscribed to the Channel.

- If the server is a Linux or Solaris Subscriber that does not have a drive mapped to it (such as through using Samba) from the workstation you are using to resolve certificates, see .

A prompt to copy a certificate is usually displayed when you have added:

- A Channel to a Distribution
- A Distribution to a Channel
- A Subscriber to a Channel
- A Channel to a Subscriber

To initiate resolving certificates:

1 In ConsoleOne, right-click the Distributor object, then click *Resolve Certificates*.

2 Make sure the *Copy Certificates Automatically to Subscribers* option is selected, then click *OK*.

This copies the new certificate to each Subscriber so that it can receive Distributions from this Distributor, as long as the workstation where you are running ConsoleOne can contact all of the Subscriber servers. If you are prompted for a location to copy the certificates, you must have a drive mapped to the destination server.

For information specific to resolving certificates for External Subscribers, see Section 7.1.10, "Manually Copying Certificates for Non-Encrypted Distributions," on page 317.

7.1.7 Handling Invalid Certificates

A Subscriber cannot receive Distributions from a Distributor when the Distributor's certificate has become invalid. A Subscriber cannot receive encrypted Distributions when the Subscriber's encryption certificate has become invalid. For information on encryption certificates, see Section 7.2, "Distribution Security Using Encryption," on page 317.

A Distributor's certificate can become invalid when the DNS name or IP address of the Distributor has been changed. However, if your Distributor is configured to use DNS (the recommended addressing method), IP address changes on the Distributor do not invalidate its certificate. Also, if DNS addressing is being used, changes in a Subscriber's DNS name or IP address do not prevent the Subscriber from receiving Distributions.

However, a Subscriber's encryption certificate can become invalid when the DNS name or IP address of the Subscriber is changed, in which case a new encryption certificate needs to be created.

The following applies for DNS name changes where DNS is your installed addressing method, or for IP address changes where IP address is your installed addressing method:

- "Distributor DNS Name or IP Address Is Changed" on page 312
- "Subscriber DNS Name or IP Address Is Changed" on page 314

Distributor DNS Name or IP Address Is Changed

Because the Distributor identifies itself to Subscribers by its server's DNS name or IP address, if you change the identifier being used on the Distributor server, Subscribers do not recognize the Distributor as a valid source for Distributions.

Changing the DNS name or IP address of a Distributor causes the certificate created by the Distributor to be invalid for all Subscribers that have received the certificate from this Distributor. Therefore, the Distributor must send new certificates to all Subscribers receiving Distributions from that Distributor.

To re-create and resolve the Distributor's certificate, do the following in order:

1. "Modify the Distributor Server's Identification Attributes" on page 313
2. "Create and Send New Certificates" on page 313

Modify the Distributor Server's Identification Attributes

You must first modify the Network Address attribute on the Other tab in the Distributor and Subscriber objects' properties.

If the server is using the DNS Name attribute to identify itself, do the following:

1 In ConsoleOne, right-click the Distributor object, click *Properties*, then select the *Other* tab.

2 Click the + symbol to the left of *NetWork Address*.

3 Select the icon to the left of the field you want to modify.

A *Browse* button is displayed to the right.

4 Click the *Browse* button.

5 If you are modifying the *DNS Name* field, click the drop-down list at the top of the box where Type 13 is displayed.

6 Change the value from Type 13 to IP, then change IP back to Type 13.

This resets the value to now recognize the new DNS name.

7 Click the *Browse* button to the right of the *NetAddress* field in the lower portion of the box.

8 Select *Servers DNS Name* (on the right side of the box), then change it to the new name.

9 Click *OK* to return to the *Other* tab.

10 Click *OK* to finish.

If the server is using the IP Address attribute to identify itself, do the following:

1 In ConsoleOne, right-click the Distributor object, click *Properties*, then select the *Other* tab.

2 Click the + symbol to the left of *NetWork Address*.

3 Select the icon to the left of the field you want to modify.

A *Browse* button is displayed to the right.

4 Click the *Browse* button.

The IP address is displayed in the lower portion of the dialog box.

5 Change the IP address to the new one.

6 Click *OK* to return to the *Other* tab.

7 Click *OK* to finish.

Create and Send New Certificates

1 On the Distributor server, shut down the Distributor Agent:

NetWare: At the `ZENworks Server Management` console prompt, enter `exit`.

Windows: In the Services dialog box, stop the Novell ZENworks Service Manager service.

For information on stopping and starting agents, see "Starting and Stopping Server Management Services" in the *Novell ZENworks 7 Server Management Installation Guide*.

2 In the `\zenworks\pds\ted\security\private` directory on the Distributor server, delete the `.keystore` file.

This file contains the Distributor's certificate.

3 In the `\zenworks\pds\ted\security\csr` directory on the Distributor server, delete the `.csr` file that has a name that matches either the old DNS name or the old IP address.

4 Restart the Distributor Agent.

A new certificate and `.keystore` file are automatically created for the Distributor.

5 To send new certificates to all Subscribers that receive Distributions from the Distributor selected in :

  5a To resolve certificates, in ConsoleOne, right-click the Distributor object, then click Resolve Certificates.

> **IMPORTANT:** ConsoleOne copies the certificate files to Subscriber servers. Therefore, the client software on the workstation running ConsoleOne must have access to the Subscriber servers' file systems. For Windows Subscriber servers, the Domain and Workgroup rights on the workstation must be set up to facilitate automatic certificate copying. Otherwise, a 1204a error is given.

  5b Make sure the Copy Certificates Automatically to Subscribers option is selected, then click OK.

This copies the new certificate to each Subscriber so that it can receive Distributions from this Distributor, as long as the workstation where you are running ConsoleOne can contact all of the Subscriber servers. If you are prompted for a location to copy the certificates, you must have a drive mapped to the destination server.

Subscriber DNS Name or IP Address Is Changed

Because the Distributor obtains the address of a Subscribers from the Subscriber's object in eDirectory, this information must be updated in the Subscriber object so that it can receive its Distributions.

Changing the DNS name or IP address of a Subscriber causes all encryption certificates contained on the Subscriber to be invalid. Subscribers can have one encryption certificate from each Distributor that sends it encrypted Distributions.

Subscribers can continue to receive non-encrypted Distributions, even if the DNS name or IP address is changed.

The following sections outline the steps to resolve DNS name or IP address changes:

Modify the Subscriber Server's Identification Attributes

You must first modify the Network Address attribute on the Other page in the Distributor and Subscriber objects' properties. To accomplish this, do the following as applicable.

If the server is using the DNS Name attribute to identify itself, do the following:

1 In ConsoleOne, right-click the Subscriber object, click *Properties*, then select the *Other* tab.

2 Click the + symbol to the left of *NetWork Address*.

3 Select the icon to the left of the field you want to modify.

A *Browse* button is displayed to the right.

4 Click the *Browse* button.

5 If you are modifying the *DNS Name* field, click the drop-down list at the top of the box where Type 13 is displayed.

6 Change the value from Type 13 to IP, then change IP back to Type 13.

This resets the value to now recognize the new DNS name.

7 Click the *Browse* button to the right of the *NetAddress* field in the lower portion of the box.

8 Click *Servers DNS Name* (on the right side of the box), then change it to the new name.

9 Click *OK* to return to the *Other* tab.

10 Click *OK* to finish.

If the server is using the IP Address attribute to identify itself, do the following:

1 In ConsoleOne, right-click the Subscriber object, click *Properties*, then select the *Other* tab.

2 Click the + symbol to the left of *NetWork Address*.

3 Select the icon to the left of the field you want to modify.

A *Browse* button is displayed to the right.

4 Click the *Browse* button.

The IP address is displayed in the lower portion of the dialog box.

5 Change the IP address to the new one.

6 Click *OK* to return to the *Other* tab.

7 Click *OK* to finish.

Resolve the New Certificates

To reproduce valid encryption certificates for the Subscriber, follow the instructions under Section 7.2, "Distribution Security Using Encryption," on page 317.

7.1.8 Certificate and Private Key Directories

Certificates and private keys for Policy and Distribution Services are stored in the following locations in the `.keystore` file:

- For the Distributor's private key on a NetWare Distributor server:

  `sys:\zenworks\pds\ted\security\private`

- For the Distributor's private key on a Windows Subscriber server:

  `c:\zenworks\pds\ted\security\private`

- For certificates received from Distributors on a NetWare Subscriber server:

  `sys:\zenworks\pds\ted\security`

  After the Distribution has been sent, the certificate is moved into the `.keystore` file.

7.1.9 Creating Security Certificates for Non-Encrypted Distributions

To create a certificate on a Distributor and copy it to its associated Subscribers:

1 On the server where a Distributor is installed, make sure its Distributor Agent is running (use `zfs.ncf` on a NetWare server, restart the Novell ZENworks Service Manager service on a Windows server, or enter `/etc/init.d/novell-zfs start` on a Linux or Solaris server).

This Java process creates the certificate and writes it to eDirectory.

2 Copy the certificate to each Subscriber using one of the following methods:

- If your Channels and Distributions are set up, right-click the Distributor object in ConsoleOne, click Resolve Certificates, then click OK. Make sure the Copy Certificates Automatically to Subscribers option is selected before clicking OK. This copies the new certificate to each Subscriber so that it can receive Distributions from this Distributor.

  For information on resolving certificates, see .

- If necessary, associate Subscribers with a Channel, create a Distribution for the Distributor, then associate the Distribution with a Channel. When you click OK, you are prompted to resolve the certificate. Respond to the query with Yes to resolve certificates for all Subscribers. The certificates are copied to all of the associated Subscribers. The Subscriber Java process does not need to be running on the Subscriber server; the server only needs to be up.

- Manually copy the Distributor's certificate to each Subscriber server's *installation_path*`\zenworks\pds\ted\security` directory (on Linux or Solaris, `/var/opt/novell/zenworks/zfs/pds/ted/security`). This method is necessary if you do not have a drive mapped to the Linux or Solaris server to the workstation you are using to resolve certificates.

- Right-click a Subscriber object, then click Resolve Certificates (repeat for each Subscriber object). This option might only be available if you answered No when prompted to copy security certificates.

The first two options are the easiest when there are many Subscribers receiving Distributions from one Distributor.

3 Because each Distributor creates its own security certificate, repeat Step 1 and Step 2 for each Distributor object in the tree.

7.1.10 Manually Copying Certificates for Non-Encrypted Distributions

To manually copy certificates to Subscribers using ConsoleOne:

1 Right-click a Distributor, Subscriber, or External Subscriber object, then click *Resolve Certificates*.

or

Click *File*, then click *Resolve Certificates*.

2 Select the *Save Certificates to Disk* option.

3 Provide a path for where to copy the certificate file, then click *OK*.

The certificate file that is copied to this path is named using the following syntax:

DNS_Name`.cer`

4 Copy the *DNS_name*`.cer` file from the path you gave to the Subscriber server's `\zenworks\pds\ted\security` directory (on Linux or Solaris, `/var/opt/novell/zenworks/zfs/pds/ted/security`).

7.2 Distribution Security Using Encryption

Policy and Distribution Services provides the option to encrypt a Distribution to prevent unauthorized access to its contents when the Distribution is sent outside your secured network. There is usually no need to encrypt Distributions that are sent within your secured network.

Encrypting Distributions is a two-step process:

1. Select the Encrypt check box in the Distribution's properties in ConsoleOne and select the level of encryption (strong or weak).

2. Manually create and copy the encryption security certificate files between the Distributor and Subscriber servers.

IMPORTANT: For security, you should use a physical medium, such as a diskette, to transfer the certificate between network servers.

Thereafter, the Distribution is sent as an encrypted Distribution.

To understand Distribution encryption, review the following:

- Section 7.2.1, "Creating and Copying Encryption Certificates," on page 317
- Section 7.2.2, "Sending an Encrypted Distribution," on page 320
- Section 7.2.3, "Extracting an Encrypted Distribution," on page 321

7.2.1 Creating and Copying Encryption Certificates

RSA PKIs provide the security process used for encrypted Distributions.

Encryption certificates are created from Certificate Signing Request (`.csr`) files. Every Subscriber server contains a `.csr` file that can be used as a template for creating an encryption certificate for a particular Distributor.

The encryption certificates (`.cer`) are used by the Subscribers to ensure secure transmission of an encrypted Distribution. If you pass the `.cer` file over the wire, the Distribution's encryption key could be compromised. Therefore, you must manually copy the encryption security certificates to ensure that the encryption key contained in the certificate files is kept secure.

IMPORTANT: Do not manually copy a certificate by using a file browser, because that uses transmission lines and can be compromised. Instead, copy the certificate to an external media, such as a floppy diskette, and transport it physically between the Distributor and Subscriber servers.

To use encryption certificates with Subscribers, you must have previously resolved certificates and sent an non-encrypted Distribution to each Subscriber.

For information on resolving certificates, see Section 7.1.6, "Resolving Certificates," on page 311.

Figure 7-3 illustrates the process of manually copying the encryption certificates:

Figure 7-3   *Manually Copying Entryption Certificates*
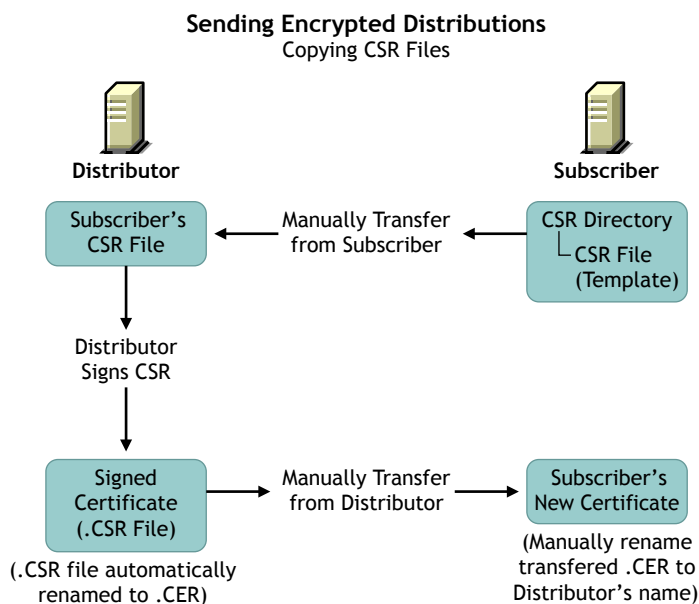
The Distributor signs the `.csr` to create the encryption `.cer` file, which is manually copied from the Distributor to the Subscriber to replace the current non-encryption `.cer` file on the Subscriber server.

The encryption certificate is required for extracting a Distribution. If a Subscriber is only acting as a parent Subscriber to pass the encrypted Distribution on to Subscribers who have subscribed to the Distribution's Channel, the parent Subscriber does not need to have the encryption certificate on its server.

To create certificates for an encrypted Distribution:

1 Determine the Distribution you want encrypted.

2 Determine the Distributor that owns this Distribution.

3 Determine which Subscribers should receive the encrypted Distribution.

4 Resolve certificates for the selected Distributor to the selected Subscribers, then send a non-encrypted Distribution from that Distributor to the Subscribers.

For information on resolving certificates, see Section 7.1.6, "Resolving Certificates," on page 311.

5 Access the file systems of this Distributor and these Subscribers.

6 Copy every `.csr` certificate file contained in the following directory from each Subscriber to the same path on the Distributor:

`\zenworks\pds\ted\security\csr`

This path begins with whatever you used for installing ZENworks Server Management.

The Certificate Signing Request (`.csr`) is used to create the encryption certificate file.

7 In ConsoleOne, right-click the Distributor object, click *Sign CSR Files*, select the `.csr` files to be signed, click *Sign*, click *OK* on the Success dialog box, then click *Close*.

You can select multiple `.csr` files to be signed at the same time.

This creates the Certificate (`.cer`) files in the same Distributor's directory as the `.csr` files you copied from the Subscribers. You will have one `.cer` file for each `.csr` file.

You can also perform this step using iManager:

7a Select *Remote Web Console*.

7b Select or provide the Distributor's IP address.

7c In the *Available Services* drop-down box, select *Tiered Electronic Distribution*.

7d Select the *Security* tab, then click the *Sign CSR* link.

8 For each target Subscriber, do the following:

8a Copy the Subscriber server's corresponding `.cer` files from the following location on the Distributor's file system:

`\zenworks\pds\ted\security\csr`

to the following path on the Subscriber's own server's file system:

`\zenworks\pds\ted\security`

Each `.cer` file contains its Subscriber server's name.

8b Rename the `.cer` files that you just copied to the Subscriber server to have the Distributor's DNS name instead of the Subscriber's.

9 Send the encrypted Distribution.

WARNING: Under the following scenario, the encryption certificates you just created can be overwritten before they are used:

1. Changes are made to the Channel, Subscribers, or Distribution involved with the encrypted Distribution.

2. This causes the prompt for copying certificates to be displayed.

3. If you reply with Yes before the encrypted Distribution has been sent and received by the Subscribers:

   a. The encryption `.cer` file is overwritten on each Subscriber with a non-encryption `.cer` file.

   b. The Subscribers cannot decrypt the Distribution when it is received, because the `.cer` file was overwritten with a `.cer` file that does not contain the encryption keys.

After the encrypted Distribution has been sent once to each Subscriber, the encryption `.cer` file is moved into the `.keystore` file on the Subscriber server's file system so that it cannot be overwritten. Thereafter, you can reply with Yes to copy certificates when this scenario occurs.

7.2.2 Sending an Encrypted Distribution

After an encryption certificate has been established on a Subscriber server, Figure 7-4 illustrates the process for sending encrypted Distributions:

Figure 7-4 *Sending Encrypted Distributions*
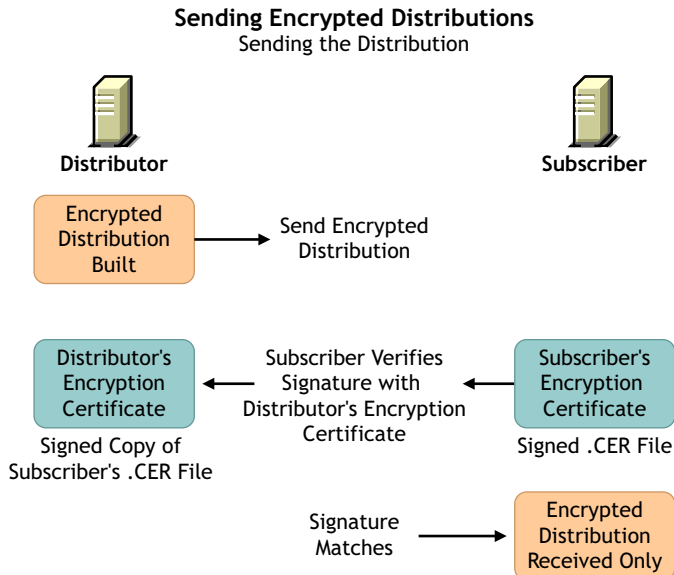
The only Subscribers that need to receive the encryption key are those that are extracting the Distribution. Therefore, parent Subscribers and Subscribers in the Distributor's routing hierarchy do not need to receive the encryption key if they are not extracting the Distribution.

7.2.3 Extracting an Encrypted Distribution

Before an encrypted Distribution can be extracted on a Subscriber server, the Subscriber must receive the encryption key. Figure 7-5 illustrates how the key is sent:

Figure 7-5 *Sending Encryption Keys*

Sending Encrypted Distributions
Sending the Encryption Key
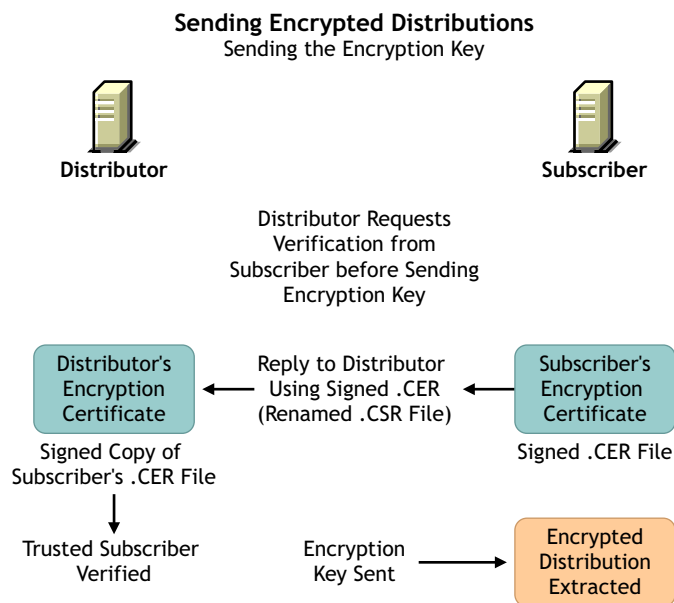
Each Distribution has its own encryption key sent.

7.3 Security for Inter-Server Communication Across Non-Secured Connections

Policy and Distribution Services uses XMLRPC (Extensible Markup Language Remote Procedure Call) for its normal inter-server communications. XMLRPC optionally provides security for inter-server communication across non-secured connections. Policy and Distribution Services can use this security for inter-server communications between servers across non-secured connections, or between a management workstation and servers across non-secured connections. For example, firewalls, intranets, or NAT configurations.

This inter-server communications security ensures that data received across a non-secured connection is from a trusted source, that it has not been tampered with en route, and that the data received can be trusted by other machines. This is accomplished through the use of signed security certificates and digital signatures.

This security requires modifications to certain text files, and is installed using a Server Management wizard.

The following are instances when you could want inter-server communication security:

- **ConsoleOne administration:** When you use a workstation to manage a Distributor server across a non-secured connection.
- **SET parameters:** When you create a SET Parameter policy or a software package for SET parameters, inter-server communication takes place to provide the target server's SET parameter information. This communication could cross a non-secured connection.
- **Server Down policy:** When you use this policy to down a server, the communication between the downed server and another server watching for it to come back up could cross a non-secured connection.

For instructions on installing XMLRPC security, see "Installing Additional Security for Non-Secured Connections" in the *Novell ZENworks 7 Server Management Installation Guide*.

Review the following sections to understand inter-server communications security using XMLRPC:

7.3.1 Terms Used in This Section

The terms and acronyms listed in Table 7-1 are used in this security documentation:

Table 7-1 *Inter-Server Communications Security Terms*

Term	Explanation
CA	Certificate Authority The trusted certificate source responsible for digitally signing other server's x.509 certificates.
CS	Certificate Signer The trusted certificate source responsible for digitally signing other server's XMLRPC certificates.
certificate or security certificate	An electronic document that contains an electronic signature for validating anything associated with the certificate, such as a Distribution.
CSR	Certificate Signing Request Request by a server to have an XMLRPC certificate signed by the trusted CS. This is not an X.509 certificate that would be signed by a root CA, such as VeriSign* or Thawte Consulting.
self-signed certificate	A valid certificate signed by its creator.
signed certificate	A certificate signed by a CS, which makes it valid for acceptance by the receiving server.

Term	Explanation
SSL	Secure Socket Layer
XMLRPC	Extensible Markup Language Remote Procedure Call
	Software used by Server Management and Tiered Electronic Distribution for inter-server communications.

7.3.2 Security Certificates

Inter-server communications security uses signed certificates issued by the Certificate Signer (CS), which are valid only within the context of the Novell ZENworks family of products.

The certificates used are not X.509 compliant and cannot be used for any e-commerce or SSL applications.

7.3.3 Using SSL

When a CS servlet signs a Certificate Signing Request (CSR), the requesting client must authenticate with a username and password via HTTP Basic Authentication. You can secure the username and password by using SSL. For information on how to enable SSL for a commercial Web server, see your SSL documentation.

7.3.4 Format of the Password File

Inter-server communications security uses a password file for the username and password that are authenticated for CSR signing. You can create the password file in a text editor and place it in any secure location. You should also restrict access to the file to only the users who are listed in the file.

Usernames and passwords are both case sensitive. The syntax for the password file is:
`username=password`

For example:

admin=adminpassword
CSsigner=cspassword
JohnDoe=jdpassword

You should limit the access to the password file to those users included within the file.

7.3.5 TCP/IP Addresses and DNS Names

In setting up inter-server communications security, the installation program relies on addresses or names of the servers where you want this security enabled. You can use either TCP/IP addresses or fully distinguished DNS server names.

For the various methods you can use to obtain these addresses or server names, see "Gather Information for Installation" in the *Novell ZENworks 7 Server Management Installation Guide*.

Scheduling

8

The following information on scheduling applies to Policy and Distribution Services in Novell® ZENworks® Server Management:

- Section 8.1, "Understanding Scheduling in Policy and Distribution Services," on page 325
- Section 8.2, "Scheduling and Tiered Electronic Distribution Objects," on page 327
- Section 8.3, "Scheduling and Server Policies," on page 346

8.1 Understanding Scheduling in Policy and Distribution Services

Review the following:

- Section 8.1.1, "Why Scheduling is Necessary for Distributions," on page 325
- Section 8.1.2, "Scheduling Is Required for Some Server Policies," on page 326
- Section 8.1.3, "Scheduling Differences Between Server Policies and Tiered Electronic Distribution," on page 326
- Section 8.1.4, "Precedence of the Tiered Electronic Distribution Policy," on page 327

8.1.1 Why Scheduling is Necessary for Distributions

When you create a Distribution (by creating and configuring its object), you want it to reach certain Subscriber servers to be used by them, and you want that to happen in a useful time frame. The distribution process requires scheduling in order to do this.

Basically, the distribution process is:

1. You create and configure a Distribution object.
2. The Distributor object that you assigned the Distribution object to reads eDirectory and discovers the new Distribution.
3. The Distributor server builds the Distribution file according to your configuration.
4. You associate the Distribution object with a Channel object.
5. The Distributor server sends the Distribution file to the Subscriber servers that are subscribed to that Channel.
6. The Subscriber servers extract and install the Distribution's content.

This distribution sequence needs to be scheduled for the following reasons:

- **Ordering the distribution process:** The flow of a Distribution from one process to another needs to be ordered so that the Distribution gets distributed and used in a timely manner.

  Conflicting scheduling might cause a Distribution to never get through the process, or to arrive and get used much later than you anticipated.

- **Minimizing network traffic:** Scheduling can provide flexibility in controlling network bandwidth usage. For example, you can schedule large Distributions to be sent when your network's traffic is at its lightest.
- **Minimizing impact on servers:** Scheduling helps to minimize the impact of building, sending, and extracting Distributions for the servers involved. For example, you can schedule large Distributions to be built and extracted during off-peak hours or on weekends.

Scheduling does not affect the total network resources used by a Distribution. It only affects when those resources are used.

8.1.2 Scheduling Is Required for Some Server Policies

Some policies must be scheduled before they can be enforced.

If you enable a policy, but do not schedule it, it is activated according to the schedule currently specified in the Default Package Schedule, which provides a default for scheduled policies. The default schedule is to run at System Startup.

The order of enforcement of different server policies is not guaranteed if the policies use exactly the same schedule. In other words, you should stagger the policies' schedules if you want to ensure the order in which they are enforced.

For information on scheduling policies, see Section 8.3, "Scheduling and Server Policies," on page 346.

For information on policies, see Chapter 4, "Server Policies," on page 197.

8.1.3 Scheduling Differences Between Server Policies and Tiered Electronic Distribution

Policies are scheduled according to local times. Tiered Electronic Distribution objects are scheduled according to an offset from Greenwich Mean Time (GMT).

Server Policies example: If you are residing in Utah and set a policy to be executed at 5 p.m. Utah time, it would be executed at 5 p.m. local time in Utah for servers residing in Utah. In California, it would execute at 5 p.m. local time in California. In other words, setting a time of 5 p.m. for a policy makes it execute at 5 p.m. local time wherever the servers reside.

Tiered Electronic Distribution example: If you are residing in Utah during Daylight Saving Time and set a Tiered Electronic Distribution object's schedule for 5 p.m., it would be executed at 5 p.m. local time in Utah. In California, it would execute at 4 p.m. local time (5 p.m. in Utah) for servers residing in California. In other words, Tiered Electronic Distribution schedules are relative to a GMT offset that makes the Tiered Electronic Distribution schedule execute at the exact same moment worldwide.

For Distributions, you can define a window of opportunity during the day for when a schedule's action is to begin and end. Distributions are anticipated to occur during off-peak hours. For some networks, it is possible that the scheduling window can be very short. Other systems on the network also use off-peak hours for processing, such as backups.

You can have instances where the limiting factor is available time; therefore, the critical condition is how fast the distributions can take place, regardless of the resources consumed. You might need to experiment to determine the best relationship between time and resources.

8.1.4 Precedence of the Tiered Electronic Distribution Policy

If you set a schedule in the Schedule tab for the Tiered Electronic Distribution policy (in the Service Location Package), this schedule is the default for all Distributors and Subscribers for which the policy applies, unless in ConsoleOne you set a schedule for a specific Tiered Electronic Distribution object. In other words, modified schedules for Distributors and Subscribers automatically override the Tiered Electronic Distribution policy schedule.

The Distributor and Subscriber schedules are different. There are separate Schedule tabs for the Distributor's Refresh and Subscriber's Extract schedules.

By default, when a schedule is set in the Tiered Electronic Distribution policy, the Use Policy check boxes are displayed on both the General and Schedule tabs for all Distributors and Subscribers. And, the box is automatically selected for the Distributor and Subscriber objects that have not yet had their schedules modified. It is deselected for the objects that have a schedule defined.

You can disable the Tiered Electronic Distribution policy's default schedule for a specific Distributor or Subscriber by deselecting the Use Policy check box in the object's properties. Then you must define a schedule in the object's properties for it to have a usable schedule.

You can override a specific Distributor or Subscriber schedule by selecting the Use Policy check box in that object's properties. The Tiered Electronic Distribution policy's schedule is then applied to that Distributor or Subscriber.

For information on how to create, configure, and schedule the Tiered Electronic Distribution policy, see "Tiered Electronic Distribution" on page 213.

8.2 Scheduling and Tiered Electronic Distribution Objects

Scheduling can be a complex undertaking if you do not understand the fundamental scheduling principles. Review the following sections for guidelines that will help you to set up effective schedules for your Distributions:

8.2.1 Understanding the Tiered Electronic Distribution Objects and Their Schedules

When sending Distributions between Distributor and Subscriber servers, several Tiered Electronic Distribution objects are involved in the distribution process. Because of this, you must set schedules in some of the objects so that the process flows efficiently, yielding the intended distribution results.

The following sections explain the schedules:

- "The Tiered Electronic Distribution Schedules" on page 328
- "Distributor Object's Refresh Schedule" on page 329
- "Distribution Object's Build Schedule" on page 330
- "Channel Object's Send Schedule" on page 331
- "Subscriber Object's Extract Schedule" on page 332

The Tiered Electronic Distribution Schedules

The Tiered Electronic Distribution objects listed in Table 8-1 can be scheduled:

Table 8-1 *Tiered Electronic Distribution Schedules*

Tiered Electronic Distribution Object	Schedule Name	Scheduling Purpose
Distributor	Refresh	Tells the Distributor when it should re-read eDirectory to discover any changes to its Distributions. If it finds changes, it rebuilds the Distributions according to the Distribution objects' Build schedules.
Distribution	Build	Tells the Distributor when it can build a particular Distribution.
Channel	Send	Tells the Distributor when it can send the Distributions it owns in the Channel.
Subscriber	Extract	Tells the Subscriber when it can extract and install any Distributions it has received and hasn't yet extracted.

The above Tiered Electronic Distribution objects must be scheduled or they cannot perform their Distribution-related actions, such as determining when Distributions are discovered, built, distributed, and extracted.

The following Tiered Electronic Distribution objects do not have schedules:

External Subscriber
Subscriber Group
Policy Package [1]

[1] Only the Container Package and Service Location Package. The Distributed Policy Package can be scheduled using the Schedule tab in the Distribution object.

The following sections explain scheduling issues:

- "Server CPU Usage by the Schedules" on page 329
- "Schedule Types" on page 329
- "Resolving Certificates when Changing Schedules" on page 329

Server CPU Usage by the Schedules

Schedules do not directly affect the total resources used by a Distribution (such as CPU cycles, bandwidth, and disk space), but rather when the resources are used. Therefore, Tiered Electronic Distribution's schedules control when Distributions are built, sent, and extracted.

However, CPU usage is affected by which servers are being used to perform a schedule's action. A server's CPU time depends on which Tiered Electronic Distribution function is running on the server, as shown in Table 8-2:

Table 8-2   *CPU Time Usage by Schedule*

Server's CPU Time	Schedules
Distributor	Refresh, Build, Send
Subscriber	Extract
parent Subscriber	Send, Extract

Schedule Types

When you set an object's schedule, you have the following schedule types to choose from:

Never
Daily
Monthly
Yearly
Interval
Time
Run Immediately (except for the Distributor object)

For more information on the schedule types, see "Frequency" on page 334 and Appendix B, "Schedule Types," on page 409.

Resolving Certificates when Changing Schedules

You might need to resolve certificates when making changes to one of the schedules. For more information, see Section 7.1.6, "Resolving Certificates," on page 311.

Distributor Object's Refresh Schedule

A Distributor's schedule determines when the Distributor reads Novell eDirectory™ for configuration changes. This enables the Distributor to respond to a request to build a Distribution. The Distributor rebuilds a Distribution when the Distribution's schedule indicates that is should be built.

When the Channel's Send schedule starts, the Distributor checks with the Subscribers that it sends to directly to see if they have the current Distribution. However:

 • If the Distribution is non-sequential, the Distributor simply checks for the current version.

◆ If the Distribution is sequential (the File or Desktop Application types of Distributions only), it checks to see if the Subscribers have all of the versions of the Distribution, starting with the baseline and every change since the baseline.

If the Subscriber does have the entire Distribution, it checks with its subordinate Subscribers to see if they do, and so on down the routing hierarchy.

The time it takes to verify that all receivers have all of the Distributions in the Channel is minimal.

IMPORTANT: A Distribution might never get sent completely if the Refresh schedule is shorter than the time it takes to build or send the Distribution. In other words, if the Refresh schedule is too short, when the Distributor is refreshed the Distribution in the process of being built or sent could be cancelled before it has completed sending. Therefore, we recommend the Distributor's Refresh schedule be daily, unless changes to Distributions warrant a more frequent refresh, then set it in hours. Do not refresh the Distributor more often than every five minutes.

Scheduling a Distributor

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

2 Select the *Schedule* tab > click the arrow for the drop-down box > click *Interval* > select an interval, such as *Daily*.

3 Set the start and end times, if necessary.

The *Start Time* and the *End Time* fields specify the time window for performing the schedule's action.

You can repeat the action every so often throughout the day.

You can also have the refresh occur randomly in the specified time window. For more information, see "Using the Randomly Dispatch Option in a Distributor's Refresh Schedule" on page 344.

4 Click *OK*.

Distribution Object's Build Schedule

The Distribution's schedule determines when a Distributor is requested to create the Distribution file based on the definition in the Distribution object.

Most Distributions consist of a set of files that change over time and need to be redistributed on a regular basis. Each Distribution has its own Build schedule that tells the Distributor how often to rebuild the Distribution. When the Distributor builds a Distribution, it automatically compares it with the previous version to see if there are any changes.

For the File Distribution, if there are no changes in the current build, no new version is created. If there are changes, a delta is built consisting of only the changes to be distributed.

For the FTP, HTTP, and Software Package Distribution types, a new version is only built if there has been a change since the last version. The Distributor sends the complete new version to all target Subscribers.

The Distribution's End Time is used to determine the end time for randomly dispatching events. In other words, the Distributor does not stop building the Distribution until it is complete.

Deleted files and directory synchronization are handled in the Build schedule.

Scheduling a Distribution

1 In ConsoleOne, right-click a Distribution object > click *Properties*.

2 Select the *Schedule* tab > click the arrow for the drop-down box > select a schedule type, such as *Run Immediately*.

You can repeat the action every so often.

The *Start Time* and the *End Time* fields specify the time window for performing the schedule's action.

You can also have the build occur randomly in the specified time window (if you select the Daily schedule type). For more information, see "Using the Randomly Dispatch Option in a Distribution's Build Schedule" on page 345.

3 Click *OK*.

Channel Object's Send Schedule

A Channel's Send schedule provides a window of time for when a Distributor can start sending its Distributions to the Subscribers associated with that Channel.

The Channel's schedule applies only to the Distributor and its direct receivers (first tier Subscribers). When the Send schedule ends, the Distributor stops distributing to those first tier Subscribers.

Second-tier receivers and beyond do not adhere to the Channel's schedule. The parent Subscribers that are sending Distributions to other Subscribers continues to send a Distribution after the Send schedule ends. Their subordinate Subscribers also ignore the Send schedule.

The Send schedule's End Time forces the Distributor to stop sending a Distribution when the Send schedule ends. The Distributor starts sending the Distribution where it left off when the Send schedule begins again. A Distribution is not totally re-sent. For example, if 50 MB of a 60 MB Distribution had already been sent before the disruption, when the Send schedule starts again for the Channel, the Distributor begins sending the remaining 10 MBs.

For information on how time zones affect a Channel's schedule, see "Scheduling Tiered Electronic Distribution Objects in Different Time Zones" on page 342.

Cache and Forward has no bearing on whether a parent Subscriber continues to send a Distribution when the Channel's Send schedule ends. Parent Subscribers who have completely received a Distribution prior to the Send schedule ending continues to send that Distribution to subordinate Subscribers. There is no mechanism for controlling whether parent Subscribers should continue to send when the Send schedule ends.

IMPORTANT: A Distribution might never get sent if the Send schedule is shorter than the time it takes to send the Distribution. Therefore, we recommend the Channel's Send schedule be daily or in hours. Make the Send schedule at least long enough to allow all of the Channel's Distributions to be sent.

Scheduling a Channel

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

2 Select the *Schedule* tab > click the arrow for the drop-down box > click *Interval* > select an interval (in the *Repeat the Action Every* field), such as 1 hour > click *OK*.

The *Start Time* and the *End Time* fields specify the time window for performing the schedule's action.

For information about randomly starting the Send schedule (if you select the Daily schedule type), see "Using the Randomly Dispatch Option in a Channel's Send Schedule" on page 345.

Subscriber Object's Extract Schedule

The Subscriber's schedule determines when a Subscriber can extract a Distribution that has been received.

The Subscriber's End Time is used to determine the end time for randomly dispatching events. In other words, the Subscriber does not stop extracting the Distribution until it has completed the extraction process.

Scheduling a Subscriber

1 In ConsoleOne, right-click a Subscriber object > click *Properties*.

2 Select the *Channels* tab > click *Add* > browse for the Channel > click *Select* > click *OK*.

Make sure the Channel is listed as Active in the *Channels* list.

3 Select the *Schedule* tab > the arrow for the drop-down box > select a schedule, such as *Run Immediately*, then click *OK*.

This schedule type causes the Subscriber to extract the Distribution as soon as it is received.

The *Start Time* and the *End Time* fields specify the time window for performing the schedule's action.
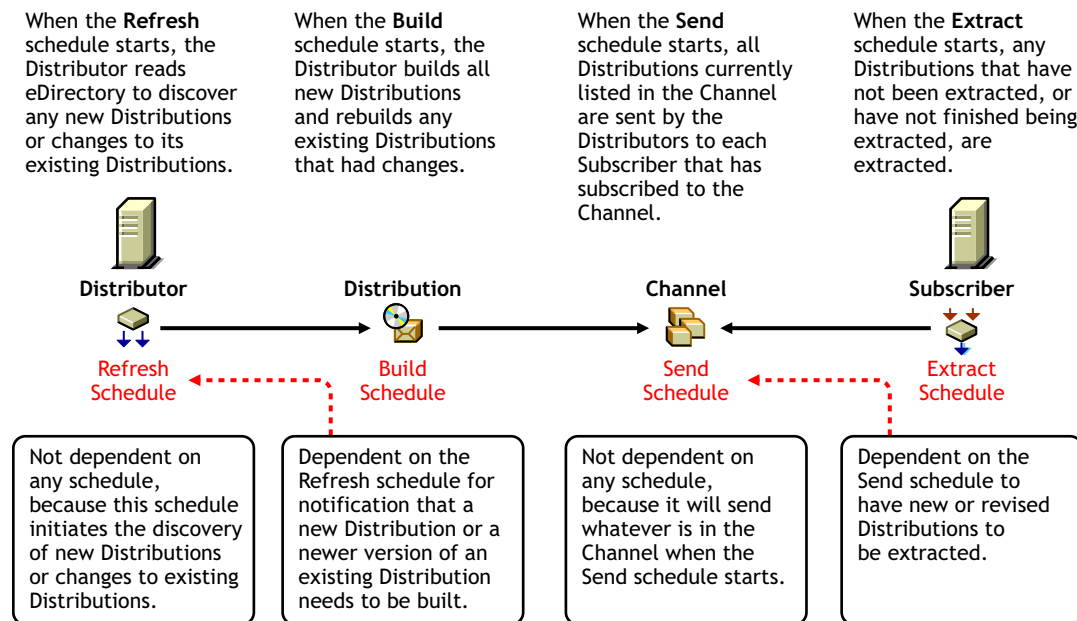
For information about randomly starting the Extract schedule (if you select the Daily schedule type), see "Using the Randomly Dispatch Option in a Subscriber's Extract Schedule" on page 345.

4 Repeat these steps for each Subscriber.

8.2.2 How the Tiered Electronic Distribution Schedules Interrelate

The Tiered Electronic Distribution object's schedules do not all interact directly with each other. There is a flow to how they are sequentially interconnected, as shown in Figure 8-1:

Figure 8-1 *Schedule Interrelationships*

When the **Refresh** schedule starts, the Distributor reads eDirectory to discover any new Distributions or changes to its existing Distributions.

When the **Build** schedule starts, the Distributor builds all new Distributions and rebuilds any existing Distributions that had changes.

When the **Send** schedule starts, all Distributions currently listed in the Channel are sent by the Distributors to each Subscriber that has subscribed to the Channel.

When the **Extract** schedule starts, any Distributions that have not been extracted, or have not finished being extracted, are extracted.

Distributor — **Distribution** — **Channel** — **Subscriber**

Refresh Schedule / Build Schedule / Send Schedule / Extract Schedule

Not dependent on any schedule, because this schedule initiates the discovery of new Distributions or changes to existing Distributions.

Dependent on the Refresh schedule for notification that a new Distribution or a newer version of an existing Distribution needs to be built.

Not dependent on any schedule, because it will send whatever is in the Channel when the Send schedule starts.

Dependent on the Send schedule to have new or revised Distributions to be extracted.

Most importantly, the sequence of Refresh, Build, Send, and Extract must have their schedules configured so that they allow a Distribution to be successfully discovered, built, sent, and extracted within the time frame that you intend.

If even one schedule is out of sync with the other three, it can take longer than intended for a Distribution to be created and eventually used.

For example, if you set the following schedules as indicated, the results are:

- **Set the Refresh schedule to occur hourly:** The Distributor is triggered to read eDirectory each hour to discover new or changed Distributions. You should determine this frequency by how large the Distributions are that this Distributor builds after each refresh. Generally, the shorter the time between refreshes, the better.

- **Set the Build schedule to run immediately:** The Distributor builds the new or changed Distributions immediately after the Distributor's Refresh schedule has caused it to read eDirectory to discovered them. However, very large Distributions might need to be built during off-peak hours.

- **Set the Send schedule to midnight:** The Distributor sends its Distributions during off-peak hours when the network's bandwidth is less busy.

- **Set the Send schedule to run immediately:** The Distributor sends its Distributions as soon as they are built.

- **Set the Extract schedule to 3 a.m.:** The Distributions received are extracted during off-peak hours when the server is likely to be the least busy. This is useful for servers receiving very large Distributions that do not need to be extracted and installed immediately.
- **Set the Extract schedule to run immediately:** The Subscriber extracts the Distribution as soon as it is received. This is useful for servers receiving important Distributions, such as virus patterns.

Ways that you could mess up the distribution scheduling flow:

- Setting the Distributor's Refresh schedule to occur too frequently to allow time to build new Distributions or rebuild changed Distributions.
- Setting the Distributor's Refresh schedule to not occur frequently enough to get important Distributions built and sent on time.
- Setting the Distribution's Build schedule to occur too frequently to allow completion of the Distributions it was building during the previous schedule window.
- Setting the Distribution's Build schedule to not occur frequently enough to get important Distributions built and sent on time.
- Setting the Channel's Send schedule to not coincide with the Distributor's Build schedule, possibly delaying the sending of Distributions.
- Setting the Channel's Send schedule window to be too short for all its Distributions to have time to complete sending.

8.2.3 The Three Timing Aspects of Scheduling

There are three time-related aspects that affect scheduling:

Frequency

When setting schedules, you determine how frequently you want a particular Distribution to be built, sent, and extracted.

The frequency for processing a Distribution can be determined using the schedule types listed in Table 8-3:

Table 8-3 *Schedule Frequencies*

Schedule Type	Functionality
Daily	Repeats the function the same time each day
Monthly	Repeats the function on a specified day of the month
Yearly	Repeats the function on a specified day of the year
Interval	Repeats the function every so often (as determined by you)

Schedule Type	Functionality
Time	The function occurs just once at a specific date and time
Run Immediately	Ignores the schedule's normal settings and starts the function immediately

The frequency you select in scheduling the distribution process should be determined by the purpose of the Distribution. For example:

- Virus protection pattern files should be distributed and installed as soon as possible whenever they become available
- A software update should be sent and installed only once

Duration

Some schedule types have durations that you may need to determine. The duration is defined by start and end times that provide a window for the time wherein the scheduled action can be performed.

Some schedules completely stop their function at the end of the schedule's duration. Therefore, the duration of a schedule must accommodate the size of a Distribution with reference to how long it takes to build it, send it, and extract it.

Duration has different issues for different schedules, as explained in the following:

The Distributor Object's Refresh Schedule

When a Refresh schedule starts, the following happens:

- **Distribution building stops:** The Distributor stops building any Distributions that it is in the middle of building. Temporary build files are not cleaned up, and building of the unfinished Distribution are not resumed where it left off. Unfinished Distributions are rebuilt by starting over when the next Build schedule starts.
- **Distribution sending is interrupted:** The Distributor stops sending any Distributions that it is in the middle of sending. However, when the Send schedule starts again, the Distributor picks up where it left of and finishes sending the Distribution.

Therefore, the Refresh schedule should not overlap the Build or Send schedules. In other words, it should start after the others end, and end when the others have not yet started.

Because the Refresh schedule can stop a Distributor from finishing a Distribution build, you may need to have multiple Distributors in your system to handle the different types of Distributions you'll be creating. For more information, see .

After the Refresh schedule's end time is reached, the Distributor picks up where it left off in sending its Distributions, but restarts building Distributions that it had not completed building when it was interrupted by the start of the Refresh schedule.

The Distribution Object's Build Schedule

When a Build schedule starts, only the Distributions that a Distributor knows about at that time start being built during the duration of the Build schedule. The Distributor learns of changes made to existing Distributions or of newly-created Distributions by reading eDirectory, which is done according to the Distributor's Refresh schedule.

After the Build schedule's end time is reached, building continues on all Distributions that it started building until the Distributions are finished being built or failed to be built.

The Channel Object's Send Schedule

When a Send schedule starts, the Distributor begins sending its Distributions that are listed in the Channel, but only those Distributions that the Distributor knows about that are listed in the Channel at the time the Send schedule starts.

When a Send schedule's end time is reached, the Distributor stops sending its Distributions, even if the Distributions have not completed being sent. However, the next time the Send schedule starts, the Distributor picks up where it left off and completes sending the partially-sent Distributions, plus begins sending any new or revised Distributions that the Distributor discovered during its Refresh schedule time.

The Subscriber Object's Extract Schedule

When an Extract schedule starts, any Distributions it has already received or will receive during its schedule's duration begins to be extracted all at the same time.

When an Extract schedule's end time is reached, the Subscriber continues to extract all Distributions that it started to extract until the Distributions are finished being extracted or failed to be extracted.

Interval

An interval is how often during a schedule's duration that the schedule restarts its function.

An interval is the equivalent to splitting up the schedule into a consecutively run series of mini-schedules. During a schedule's duration, intervals act as a stop/start position within the duration, causing the same actions to take place as for the start and stop times of the schedule itself.

Intervals can have different issues for different schedules, as explained in the following:

The Distributor Object's Refresh Schedule

Use intervals to sync up a Distributor's refresh frequency with how often you want configuration information changes Distributions, Channels, Subscribers, or policies to be recognized.

The interval should not be so short that the Distributor doesn't have time to read eDirectory and build the Distributions that it finds are new or changed.

The Distribution Object's Build Schedule

For Distributions that have changes made often to the Distribution's content that you want distributed in a timely manner, use intervals for the Build schedule to efficiently recognize those changes and provide rebuilt Distributions on time.

The Channel Object's Send Schedule

When you set intervals within a Send schedule's duration, Distributions that are in the process of being sent are stopped each time the interval begins, then pick up where it left off in sending the Distributions, plus start sending any new Distributions that were added to the queue.

If you do not use intervals, any Distributions added to the Send schedule's queue after the Send schedule starts, are not sent until the next time the Send schedule starts. Therefore, setting intervals in the Send schedule allows you to have newly-queued Distributions included in the Send schedule's window of time.

The Subscriber Object's Extract Schedule

Intervals do not make sense for the extraction process. All Distributions received prior to the start of the Extract schedule, or received while the Extract schedule is open, is extracted. Extraction continues after its schedule ends, so intervals would be ignored by the extraction process.

Using Intervals with Distributors

For any schedule type that has an interval, the event does not start until after the Distributor has re-read eDirectory. For example:

- **Daily:** If the Distributor is refreshed before the current day's time window has passed, the event runs on the current day, then every day thereafter; otherwise, it first runs during that time window on the next day, then every day thereafter.

- **Interval:** If you set the interval to be three days, the event runs three days after the day the Distributor re-reads eDirectory, then run every three days thereafter.

- **Weekly, Monthly, or Yearly:** The event runs the first day, month, or specific date (the Yearly option) after the Distributor has re-read eDirectory. For example, on Wednesday you set up a Weekly event to happen each Sunday. The Distributor re-reads eDirectory on Thursday, so the event runs the following Sunday, and every Sunday thereafter.

- **Run immediately:** As soon as the Distributor is refreshed, the event runs, then runs thereafter according to the interval you set.

To cause an event for one of the interval-related schedule types to execute out of sequence (other than Run Immediately), you can use the ZENworks Server Management role in iManager. For more information, see Chapter 2, "Novell iManager," on page 63.

8.2.4 Approaches to Scheduling

The following are approaches that you can use in determining how to set up your schedules:

Determining Whether You Need Other Distributors

Distributor server workload and the ability to complete Distribution building tasks should determine how many Distributors you need.

For example, if you have a very large Distribution that you want built during off-peak hours, which does not need to be sent immediately, and also have virus pattern Distributions that do need to be sent immediately, you might need two different Distributors, one with a daily refresh schedule (because you are only going to be building the Distribution once per day), and another with a frequent refresh schedule for discovering new virus pattern changes, so that their Distributions can be built and sent on time.

Putting Channels In Control

The idea in using Tiered Electronic Distribution is that you have a Distribution that you want to be used by the Subscribers at a certain time. To do so, you would have a Distribution built at a time when you want the Subscribers to use it. The key then is to get the other schedules to cooperate in getting the new Distribution down to the Subscribers on time.

The most useful scheduling configuration to do this places emphasis on the Channel's Send schedule. Review the following scenario:

1. A Distribution's Build schedule depends on how often you expect the Distribution's information to change. For example:

   * If the Distribution consists of forms that change monthly, and it is critical to distribute the updated forms quickly, the Build schedule should be set to Daily. This means the forms would be checked each day for changes, and the change would be found the day, or within a day, of when they are made to the forms. When it is discovered that the forms have changed, a new Distribution is built.

   * If the Distribution consists of a software application that changes once every six months or so, you may want the Distribution to build weekly. When the application is changed for the Distribution, no more than a week would pass before the Distribution was rebuilt.

2. Set all of your Subscriber's Extract schedules to Run Immediately. That way, no matter when a Distribution is built and sent, the Subscriber is ready to use it.

   You can have all of your Subscriber's Extract schedules set to Run Immediately and not worry about impacting the Subscriber server during peak business hours with a large Distribution, because you can use the Channel's Send schedule to control when the Subscribers receives and extracts a particular Distribution.

3. Set the Channel's Send schedule to correlate with when its Distributions are scheduled to be rebuilt, and to occur when you want the Subscribers to extract them.

   * In the case of a Distribution that changes monthly, set the Channel's Send schedule to monthly.

   * In the case of a Distribution that only changes every six months or so, set the Channel's Send schedule to yearly or at an interval of *xxx* number of days.

   * In the case of a large Distribution that needs to be extracted during off-peak hours, set the Channel's Send schedule to run immediately, if all you are concerned with is the Distribution's extraction, which is determined by the Subscriber's Extract schedule, which can be set to control off-peak hour extraction.

Simply, build a Distribution when it is needed, get your Subscribers ready to extract and use the Distribution as soon as they receive it, then set up your Channel to send the Distribution at the optimum time for the Subscribers.

Enabling Load-Balancing for Distributors

To help load balance a Distributor server's distribution duties, do the following:

1. Select the Maximum Number of Concurrent Distributions option on the Distribution object.

2. For the Distribution object, use the Randomly Dispatch option for the Daily, Monthly, or Yearly schedule type.

   For more information on the Randomly Dispatch option, see "Using the Randomly Dispatch During Time Period Option" on page 344.

This spreads the network traffic that is caused by sending many Distributions over the entire scheduling window.

Inactivating Schedules

A Distribution can be set as Active or Inactive:

- **Active:** The Active check box is found on the General tab of the Distribution object.
- **Inactive:** Inactive is used when you are building a Distribution because you want to keep it inactive until it is ready to be sent to a Subscriber.

We recommend that as you are either creating or modifying a Distribution object, its associated Channel be set to Inactive until you are ready to begin distributing the Distribution package. This prevents the Distribution from being inadvertently sent before you have completed its configuration.

Scheduling Conflicts with Other Software

Most distributions are anticipated to occur during off-peak hours. For some networks, it is possible that this scheduling window may need to be very short. Other systems on the network can also use off-peak hours for processing, such as backups.

You might have instances where the limiting factor is available time; therefore, the critical condition is how fast the Distributions can take place, regardless of the resources consumed. You might need to experiment to determine the best relationship between time and resources.

8.2.5 Scheduling Issues

The following explain various scheduling issues:

- "Schedule Interactions" on page 340
- "Time Zones and Scheduling" on page 341
- "Using the Randomly Dispatch During Time Period Option" on page 344
- "Repeating Actions" on page 346

Schedule Interactions

Each of the four schedules (Refresh, Build, Send, and Extract) interact with each other in ways that determine the success or timeliness of the distribution process:

- "Overall Interaction Issues" on page 340
- "Refresh versus Build" on page 340
- "Refresh or Build versus Send" on page 341
- "Build versus Send" on page 341
- "Send versus Extract" on page 341

Overall Interaction Issues

Because the distribution process is dependent on each of the four schedules interacting with each other in a timely manner, the purpose of a Distribution should help you to determine what each of the schedules need to be that are involved in its distribution process.

For example, if you want a virus pattern Distribution to be sent as soon as it is configured or as soon as a change to it has been completed, you need to make sure the schedules involved for all four Tiered Electronic Distribution objects allow the virus patterns to be in use by the target servers as soon as possible.

However, if you want a Distribution to be built, sent, and extracted during off-peak times, because it is very large and requires a lot of bandwidth in sending and server time in building and extracting it, then you would want each schedule to help in determining when to start those processes.

Because your Distributions can vary in both purpose and size, and because you may be using the same Distributor server for building these Distributions, you need to configure the various schedules to compensate for this. For example, the Distributor could have its Refresh schedule set to start every five minutes, and that would work for both sending Distributions immediately or during off-peak hours. This is because the Distribution's Build schedule would trigger when the particular Distribution would get built (immediately or during off-peak hours).

Subscriber Extract schedules are where you could have conflicts between extracting Distributions immediately or during off-peak hours. However, if you set the Subscriber's schedule to immediately extract and install its Distributions, you can use the Build and Send schedules to control when it gets certain Distributions. That way, it can extract virus pattern Distributions immediately (small, so no impact on the server), and extract large Distributions when they are sent during off-peak hours.

The fact that you can control build times individually for each Distribution, and that you usually create Channels unique to the Distributions, you can configure frequent schedules for the Distributor's Refresh and Subscriber's Extract schedules. In other words, most of your scheduling differences can be controlled by the Build and Send schedules.

Refresh versus Build

The Distributor builds a Distribution according to the Distribution object's Build schedule, but not before the Distributor's Refresh schedule has told the Distributor to read eDirectory for changes related to the Distribution, such as whether there is a new one, or that something in an existing Distribution has changed, requiring it to be rebuilt. This means the Build schedule is dependent on the Refresh schedule

Therefore, if you intend that a new Distribution be built right after you have created its object and configured it, the Distributor's Refresh schedule must be frequent enough to cause the Distribution to be built. For example, you would have the Distributor's Refresh schedule set to an interval of every five minutes.

However, if you only want a Distributor server to be building Distributions during off-peak hours, then you'd want its Refresh schedule to start and end during off-peak hours. Therefore, you would select a Refresh schedule type that allows you to specify such a time window. For example, Daily, Monthly, and Yearly each provide the capability to set a time window. Additionally, Daily allows you to specify which days of the week to read eDirectory for changes.

Refresh or Build versus Send

A Distribution can only be listed in a Channel after it has initially been built. Therefore, the Channel's Send schedule is for existing Distributions, whether they be newly built or rebuilt because of changes. The Distributor sends a Distribution according to the Channel object's Send schedule. This means the Send schedule is not dependent on the Refresh or Build schedules.

However, the Refresh and Build schedules are dependent on the Send schedule, because the Subscriber servers do not receive a Distribution until the Send schedule starts.

Build versus Send

A Channel only lists Distributions that can be sent. The Channel doesn't care whether an existing Distribution is in need of being rebuilt. The Send schedule only tells the Distributor when one or more of its Distributions can be sent to the Subscriber servers that have subscribed to the Channel.

Therefore, the Build schedule is dependent on the Send schedule in that a Distribution should be initially built or rebuilt in time to be sent when the Channel's Send schedule starts. Also, after a Distribution has been built, it must wait for the Send schedule to start to be sent.

Send versus Extract

When a Send schedule starts, the Distributor sends the Distributions listed in that Channel to the Subscriber servers subscribed to the Channel. However, the Distributions received are not extracted until the Subscriber's Extract schedule starts.

Therefore, the extraction of a Distribution is only dependent on the Extract schedule. However, the Send schedule determines when the Distribution is available for extraction. Thus, the Extract schedule is dependent on the Send schedule.

The only dependency that the Send schedule has with the Extract schedule is that a sent Distribution is not extracted and installed until the Extract schedule starts, meaning you would not count on the Send schedule alone to get Distributions completely processed.

Time Zones and Scheduling

Multiple time zones can complicate your scheduling efforts. The following sections explain some of the issues:

- "Scheduling Tiered Electronic Distribution Objects in Different Time Zones" on page 342
- "Calculating Time Differences" on page 342
- "Using Geographically-Based Channels" on page 343

Scheduling Tiered Electronic Distribution Objects in Different Time Zones

The following information concerning time zone offsets is from the perspective of the Channel object. However, this information is applicable to all Tiered Electronic Distribution objects that can be scheduled.

Because a Channel is an object in the tree that is not associated with a specific server, the Channel's time is always set to the local time zone of the workstation that is running ConsoleOne® and setting the Channel's schedule.

For example, if you (the administrator) live in New York City, the local time for any Channels you schedule from there is local New York time.

If Distributors in different time zones from the Channel have Distributions in that Channel, the Distributors need to send their Distributions according to the Channel's local time schedule. For example:

1. You set a Channel's schedule to be from 1 a.m. through 5 a.m. local time in Los Angeles.
2. In New York you select to have a Distributor's Distribution listed in that Los Angeles Channel.
3. The Distribution can be sent only between 4 a.m. and 8 a.m. in New York because for New York, being three hours ahead of Los Angeles, its time window of 4–8 a.m. is happening at the same time as the Los Angeles time window of 1–5 a.m.

You should use a time zone offset to determine the true local time when the Distributor can send its Distributions. Also, because a Channel's schedule determines when a Distribution can be sent, you must make sure the build schedules you set for your Distributions occur before a Channel's schedule.

Calculating Time Differences

The World Time Server (http://www.worldtimeserver.com) is a Web site where you can determine the time difference between any two locations in the world.

As you look at the site, note the following:

- The locations in the left frame can be listed by countries or major cities.
- The current GMT time relative to the International Date Line is displayed in the right frame.
- When you select a location in the left frame, the time displayed in the right frame includes the day, date, whether Standard Time or Daylight Saving Time is in effect, and the GMT offset.

To use this site to calculate time differences between Tiered Electronic Distribution locations,

1 Select the location for one of the Tiered Electronic Distribution sites.

2 Note the time, day/date, GMT offset, and whether Daylight Saving Time is in effect (for future reference).

3 Select the location for another Tiered Electronic Distribution site.

4 Note the time, day/date, GMT offset, and whether Daylight Saving Time is in effect.

5 Repeat this process for all of the Tiered Electronic Distribution locations where you want to coordinate schedules.

6 Using the information you have gathered, calculate the time differences between the Tiered Electronic Distribution locations.

7 Taking into consideration when events are taking place locally at the various Tiered Electronic Distribution locations, configure the appropriate schedules using the time differences.

As an example,

- A Distributor in Hawaii lists a Distribution in a Channel in New York.

- Using the World Time Server Web site, you can find that the offset between the two locations is –6 when Daylight Saving Time is in effect. (The negative number means it is later in the time sequence, so you must subtract Hawaii's time from New York's time to arrive at the correct a.m. or p.m.)

- If the Channel's starting time is 1 a.m. in New York, select 7 p.m. for the Distributor's schedule in Hawaii.

- The result is that the Distributor can start to send its Distribution at 7 p.m.

- Because Hawaii is not observing Daylight Saving Time and New York is, when New York moves back to Standard Time, the result would be 8 p.m.

If you wanted the Distributions to be sent later in the evening in Hawaii, the Channel's time window would have to start later than at 1 a.m. in New York. For example:

- You want the Distributions to begin sending at 11 p.m. in Hawaii.

- You need to set the Channel's start time to be 5 a.m. in New York.

When you set up your Channel schedules, you need to consider which object's time window is more important. For example, it might be more important for the Distributor to be sending Distributions during off-peak hours. Therefore, using the New York and Hawaii example, to have the Distributions begin sending after midnight Hawaii time, you would need to have the New York Channel's start time set to 6 a.m. or later.

Using Geographically-Based Channels

If you want a Distribution to be received at the same local time (such as 3 a.m.) when you have Subscribers in different time zones, use geographically-based Channels.

Using a single Channel for the Distribution only allows the Distribution to be received at the same time it is sent, meaning it could arrive at different local times if the Subscribers are not all in the same time zone. For example:

1. Distribution_A is created and listed in Channel_A.

2. The Send schedule for Channel_A is set to cause Distribution_A to be sent at 3 a.m. of the Distributor's local time.

3. Subscriber001 is in the same time zone as the Distributor.

4. Subscriber002 is in a time zone that is 2 hours later than the Distributor's.

5. Subscriber001 and Subscriber002 are subscribed to Channel_A.

6. Channel_A's Send schedule fires at 3 a.m. local time of the Distributor owning Distribution_A.

7. Distribution_A is sent at 3 a.m. local time of the Distributor to Subscriber001 and Subscriber002.

8. Subscriber001 receives the Distribution at 3 a.m. its local time.

9. Subscriber002 receives the Distribution at 1 a.m. its local time.

However, your intention is that Distribution_A be received by both Subscribers at 3 a.m. their local times.

To get Subscriber002 to also receive Distribution_A at 3 a.m. its local time instead of 1 a.m., do the following:

1. Unsubscribe Subscriber002 from Channel_A.

2. Create Channel_A2 for Distribution_A.

3. Subscribe Subscriber002 to Channel_A2.

4. Set the Send schedule for Channel_A2 to cause Distribution_A to be sent at 5 a.m. of the Distributor's local time.

Then:

1. Channel_A2's Send schedule fires at 5 a.m. local time of the Distributor owning Distribution_A.

2. Subscriber002 receives the Distribution at 3 a.m. its local time.

Using Channels that are geographically-based, the Distributor sends the same Distribution at different times of the day.

Using the Randomly Dispatch During Time Period Option

The Randomly Dispatch During Time Period option is available for each of the schedules (Distributor, Subscriber, Channel, and Distribution). It is used in conjunction with a time window (Start and End times) that you can set for a Daily, Monthly, or Yearly schedule type.

Randomly dispatching causes the scheduled action to run at any time during the window for the day. This helps load-balancing on servers. However, random-dispatched schedules can be confusing if you are expecting an action to take place immediately.

The following describe the issues for the Randomly Dispatch option:

- "Using the Randomly Dispatch Option in a Distributor's Refresh Schedule" on page 344
- "Using the Randomly Dispatch Option in a Distribution's Build Schedule" on page 345
- "Using the Randomly Dispatch Option in a Channel's Send Schedule" on page 345
- "Using the Randomly Dispatch Option in a Subscriber's Extract Schedule" on page 345

Using the Randomly Dispatch Option in a Distributor's Refresh Schedule

You can use the Randomly Dispatch option for Distributor Refresh schedules to load balance Distributor refreshes from eDirectory. This is useful to minimize the network traffic that can be caused by many Distributors trying to read eDirectory at the same time.

Be sure to coordinate a Distributor's Refresh schedule with that Distributor's related Distributions' Build and Channels' Send schedules.

The Distributor's Refresh schedule should be determined by how frequently Tiered Electronic Distribution information is updated in eDirectory. For example, how often new Distributions are created, properties of existing Distribution objects changed, new Channels are added, and so on. The Distributor cannot know of changes made to Tiered Electronic Distribution objects without re-reading eDirectory. An eDirectory refresh should finish before the Build and Send schedules begin.

IMPORTANT: Do not refresh the Distributor more often than every five minutes. The following can need up to five minutes to complete their processes: Distribution building, eDirectory replication, and tree walking (when no Search policy is defined).

If you are using the Randomly Dispatch option, you should consider the End time for the Refresh schedule when setting the Start times for the Build and Send schedules.

Using the Randomly Dispatch Option in a Distribution's Build Schedule

You can use the Randomly Dispatch option for a Distribution's Build schedule to load-balance the Distributor's work in building Distributions. This becomes more necessary as the number of Distributions for a Distributor grows.

Be sure to coordinate a Distribution's Build schedule with its Distributor's Refresh schedule and any related Channels' Send schedules. A Distribution build should begin after the Refresh schedule ends and finish before the Send schedules begin.

IMPORTANT: Do not refresh the Distributor more often than every five minutes. The following can need up to five minutes to complete their processes: Distribution building, eDirectory replication, and tree walking (when no Search policy is defined).

If you are using the Randomly Dispatch option, you should consider the End time for its Distributor's Refresh schedule when setting the Build schedule's Start time; and, you should consider the End time for the Build schedule when setting the Start times for the Send schedules.

Using the Randomly Dispatch Option in a Channel's Send Schedule

You can use the Randomly Dispatch option for a Channel's Send schedule to begin sending its Distributions to Subscribers randomly within a scheduling window. Each Distributor that has Distributions in the Channel calculates a random time between the specified Start and End times to begin sending its Distributions. This helps to balance the distribution workload for the network over a period of time.

For example, Distributor A and Distributor B have Distributions in a Channel. Each Distributor would calculate its own random time to begin sending its Distributions.

Another use of the Randomly Dispatch option for the Send schedule is if you have many Channels and you want all Distributions for all Channels to occur between 10 p.m. and 4 a.m. Using the Randomly Dispatch option in each Channel would allow you to disperse Distribution sending times for all Channels over that six-hour period of time.

If you are using the Randomly Dispatch option, you should consider the End time of each associated Distribution's Build schedule when setting the Send schedule's Start time; and, you should consider the End time for the Send schedule when setting the Start times for all associated Subscribers' Extract schedules.

Using the Randomly Dispatch Option in a Subscriber's Extract Schedule

You can use the Randomly Dispatch option for a Subscriber's Extract schedule to balance the Subscriber's work load in extracting Distributions.

If you are using the Randomly Dispatch option, you should consider the End times for the Send schedules of the Channels where the Subscriber is subscribed when setting the Start time for the Extract schedule.

Repeating Actions

For schedule types that have the Repeat the Action Every field, how this option works depends on other factors, such as other schedules and how frequently the Distributor reads eDirectory.

For example:

- You select Daily as the Send schedule for a Channel
- You set 1:00 a.m. to midnight (23 hours) as the sending window
- You set the Repeat the Action Every field with 1 hour as the repeat value

The action (sending the Distribution) repeats as follows:

1. Starting at 1:00 a.m. and repeating every hour, the Distributor queues the Distribution to be sent.
2. If a Distribution is in the process of being sent, it continues to be sent.
3. Once a Distribution is off the queue after being sent, the Distributor queues the next newer version for sending.

   If a previously queued version of this Distribution has not been sent yet (still in the queue), the next newest version is placed in the queue. In other words, only one version of the Distribution (the last built) is queued while another version of the Distribution is being sent.

The Distributor always sends the latest Distribution, even if the Subscriber already has it.

8.3 Scheduling and Server Policies

All policies use the default schedule (Package Schedule) except where you change the schedule in a particular policy. You can also edit the default package schedule.

Review the following sections:

8.3.1 Policy Schedules Versus Distribution Schedules

Policies that must be scheduled have two scheduling methods:

- Individual policy schedules, which are configured in a policy's properties
- The Default Package Schedule that applies to all policies that are enabled in the package that do not have individual schedules set

With reference to policies that must be distributed, policy schedules and Distribution schedules are used for different purposes:

- **Enforcement:** A policy's schedule determines when the policy can be enforced.
- **Distribution:** A Policy Package Distribution's schedule determines when the policy's Distribution is built so that the policy package can be sent, received, and extracted. After extraction, the policy's schedule then determines when the policy can be enforced.

In other words, a Distribution's schedule does not directly determine when a policy is enforced. However, a Distributions's schedule, combined with the Channel's Send and Subscribers' Extract schedules, could delay the enforcement of a policy.

8.3.2 Scheduling a Server Policy

To schedule an individual policy:

1 In ConsoleOne, right-click a Policy Package object, click Properties, then select the Policies tab.

2 Select a policy, click Properties, then select the Policy Schedule tab.

3 Select a schedule in the Schedule Type field, then configure the schedule:

IMPORTANT: The Relative and Run Immediately schedules are not available for the Scheduled Down policy.

8.3.3 Editing the Default Package Schedule

To edit the default package schedule:

1 In ConsoleOne, right-click a Policy Package object, then click Properties.

2 Click Edit.

3 Select a schedule in the Schedule Type field, then configure the schedule:

Variables

9

Review the following sections for information on variables in Novell® ZENworks® Server Management:

9.1 Understanding Variables

Review the following:

9.1.1 Why Variables?

You can use variables in Server Management to save time by more easily managing varying path information. For example, to globally control changes to the same location on all servers, you can use a variable for all volumes or drives in the script:

- Create the variable in each server's Subscriber object where the value of the variable is the server's volume or drive where the location exists.
- When the script runs, it passes the variable to each server, which in turn determines the variable's value from the variable definition in its Subscriber object's properties.
- The value identifies which volume or drive contains the desired location.

Using a variable for this information, you didn't have to individually list each server's name with its volume or drive in the script.

Variables are used to simplify referencing something that is specific to individual servers or software run on servers. For example:

Destination Volumes or Drives	Script contents to be executed
DNS Names	Server Names
IP Addresses	Working Directories
Names of text files to be modified	

Each of these can have different data per server. Variables allow you to account for those differences easily.

9.1.2 Variable Usage

Variable Syntax

Variables can be thought of as having three parts: name, value, and usage. The syntax for each is:

Name syntax: *variable_name*

Example: DEST

Value syntax: *value_of_variable*

Example: sys:

Usage syntax: %*variable_name*%

Example: %DEST%

Thus, the variable DEST would equate to sys: on the particular server where the variable is defined.

When defining a variable, you do not provide the % character for the variable's name or value. However, when using the variable, you use the % character before and after its name.

The software uses the % character to identify the beginning and ending of variable names. For example:

1. %DEST% tells the software that DEST is a variable name.

2. The software looks up DEST in a variable definition table on the receiving server to discover its value.

3. The value is then used to complete the path.

Nested Variables

You can nest variables to any level. For example, you can do the following to automate destinations:

1. Define DEST as the destination volume and directory:
   - **Variable name:** DEST
   - **Value:** %VOL_DRV%%DIR%

     Here, you have nested two other variables inside of DEST to establish its value.
   - **Usage:** %DEST%

2. On a NetWare Subscriber, define the VOL_DRV variable:

 ◆ **Variable name:** VOL_DRV

 ◆ **Value:** *attribute*

  For example, data:.

 ◆ **Usage:** %VOL_DRV%

3. On a Windows Subscriber, define the VOL_DRV variable:

 ◆ **Variable name:** VOL_DRV

 ◆ **Value:** *attribute*

  For example, C:.

 ◆ **Usage:** %VOL_DRV%

4. Define the DIR variable:

 ◆ **Variable name:** DIR

 ◆ **Value:** *attribute*

  A directory, such as `\apps`.

 ◆ **Usage:** %DIR%

The result is that you can define the destination as simply DEST, which resolves to the directory and volume or drive specified at each target server. For example:

NetWare Subscriber: `data:\apps`

Windows Subscriber: `c:\apps`

Literal % Symbols

The % symbol is a valid character for file and directory names. Therefore, you need to identify literal usage of a % character. Otherwise, the software would think a nested variable name was being provided.

Literal % characters are identified by adding an extra % character immediately before a % character in the variable's value. This makes the software recognize the % character as a literal character and not a variable indicator. For example:

Variable name: DEST

Path for the variable: temp%abc%xyz

Variable value: temp%%abc%%xyz

The first % lets the software know that the next % character is literally part of the pathname, and not an indicator that a nested variable name is next. Without the double % characters, "abc" would be interpreted as a nested variable name.

9.1.3 Variable Usage Differences

General variable definitions, such as those in the Tiered Electronic Distribution policy, provide default variable values for Subscribers where they have none defined. Variables defined in a Subscriber object override such default variable values.

For Server Software Packages, variable names are resolved differently:

1. Is the variable defined in the Server Software Package component? If so, use that value.

   IMPORTANT: A variable defined in a software package overrides any value defined in the Subscriber.

2. Is the variable one of the predefined variables? If so, use that value.

3. Is the variable a Java environment variable? If so, use that value.

9.1.4 Precedence for Determining Which Variable to Use

Variables are checked for in a specific order to determine which variable to use. The order is:

1. Server Software Packages [1]
2. Subscriber objects [1]
3. Tiered Electronic Distribution policy [1]
4. Default variables [2]
5. Environment variables [2]

[1] User-defined in Server Management
[2] Predefined

The variable is used from the first place where it is found.

9.1.5 Distribution Variable Example

Variables can also be used to specify where a Distribution is to be extracted, including the full path.

For example, you have a single Distribution with 20 Subscribers. You want to extract the Distribution to a specific volume on each of the Subscriber's servers. However, the volume name varies from server to server: 15 servers are using the data: volume and five are using vol1:.

You can edit the Distribution Volume variable for some of these Subscribers by changing the Resolve To field on the Subscriber from data: to vol1: for the five Subscribers using that volume.

When the Distribution is extracted, it goes to the correct volumes on each of the 20 servers.

9.2 Types of Variables

There are two types of variables:

- Section 9.2.1, "Predefined Variables," on page 353
- Section 9.2.2, "User-Defined Variables," on page 354

9.2.1 Predefined Variables

Predefined variables are created when ZENworks Server Management starts. They are used in Server Software Packages and Tiered Electronic Distribution, and are recognized by policy packages.

Predefined variables are not case sensitive, although they are displayed in all uppercase on the server console and in this documentation.

Syntax:

%predefined_variable_name%

where *predefined_variable_name* is the name defined by Server Management, and the % symbols tell the software that a variable name exists between them. For example:

%WORKING_PATH%

To make a predefined variable useful, its value must be set in the Server Software Package component, or in a Tiered Electronic Distribution object.

The Java environment can use predefined variables, such as SERVER_DN being used in a Java process call in a `.ncf` file.

An example of how a policy package can use a predefined variable is for the Broadcast Message text in the Server Down Process policy. The text can include a variable for the server name (%SERVER_DN%) so that the broadcast message displays the name of the server.

The Server Management predefined variables listed in Table 9-1 are available:

Table 9-1   Predefined Variables

Variable	Description and Value
BASE_PATH	Location of the Policy Manager: `sys:\zenworks\pds\smanager\`
CONF_PATH	Location of configuration files: `sys:\zenworks\pds\ted\`
IP_ADDRESS	IP address of a server, such as: 192.68.1.255
LOAD_DIR	(NetWare® only) Directory where the server was loaded from: `c:\nwserver`

Variable	Description and Value
LOG_PATH	Location of log files: `sys:\zenworks\pds\smanager\`
PLUGINS_PATH	Where the Server Management plug-ins were installed: `sys:\zenworks\pds\smanager\plugins\`
POLICY_PATH	Where the policy files (.pol) are stored: `sys:\zenworks\pds\smanager\policy\`
PROP_PATH	Where Novell eDirectory™ object properties are stored: `sys:\zenworks\pds\smanager\prop\`
SERVER_DN	Distinguished server name in eDirectory, such as: `server01.servers.novell`
SERVER_NAME	Name given the server when NetWare was installed, such as: `server01`
TED_PATH	Path to the …\ted directory: `sys:\zenworks\pds\ted\`
TREE_NAME	Name of the eDirectory tree where Server Management servers reside. This is established during installation.
VOL	Default volume: `sys:`
WORKING_PATH	Working directory for the Server Policies and Server Software Packages components: `sys:\zenworks\pds\smanager\working\`
ZWS_PATH	Where the ZENworks Web Server files are located: `sys:\zenworks\zws\`
ZWS_PROP_FILE_PATH	Where the ZENworks Web Server property files are located: `sys:\zenworks\zws\`
ZWS_SECURITY_PATH	Where the ZENworks Web Server security files are located: `sys:\zenworks\zws\security\`

9.2.2 User-Defined Variables

User-defined variables are created in the Server Software Package component, Subscriber objects, and the Tiered Electronic Distribution policy. Policy packages do not recognize user-defined variables.

User-defined variables are not case sensitive.

Syntax: %*variable_name*%

where *variable_name* is the name you give the variable when you define it. Spaces cannot be used in variable names. Use hyphens (-) or underscores (_) to separate words.

Variables defined in the Subscriber object are simple text substitutions. Text entered for the value of the variable is substituted for the variable name.

9.3 Defining a Variable

You can create variables in three locations:

9.3.1 Defining Default Variables for All Subscribers

You can use the Tiered Electronic Distribution policy to define default variables for all Subscribers. Any variables you set in this policy as defaults for all Subscribers are overridden by any same-named variables defined on the Subscriber (see Section 9.3.2, "Defining Variables for a Specific Subscriber," on page 356).

To define default variables:

1 In ConsoleOne®, right-click a Service Location Package object, click *Properties*, select the check box for the Tiered Electronic Distribution policy to both select and enable it, click *Properties*, then select the *Variables* tab.

2 Click *Add*.

3 Provide the name of the variable.

The name can be user-defined, an environment variable (Java or native), or a predefined variable.

4 Provide the value for the variable.

The value is what the variable resolves to. It can also be another variable for nesting variables.

To ensure that extraction takes place, provide an absolute path to all Subscribers. For example, if the path is only the data: volume, make sure the colon (:) is included, because it is a necessary part of the full path.

5 Provide a description (optional), then click *OK*.

6 Repeat Step 2 through Step 5 to define another variable for the Subscribers.

7 Click *OK* when you have finished defining the default variables.

8 On the Service Location Package properties page, select the *Associations* tab.

9 If there are no associations listed that include all Subscriber objects, click *Add*.

10 Browse for an eDirectory container that includes all Subscriber objects, click *OK*.

This ensures that the policy is enforced for all Subscribers. For more information, see "Tiered Electronic Distribution" on page 213.

11 Click *OK* to exit the policy package's properties.

9.3.2 Defining Variables for a Specific Subscriber

1 In ConsoleOne, right-click a Subscriber object, then select the *Variables* tab.

2 Click *Add*.

3 Provide the name of the variable.

The name can be user-defined, an environment variable (Java or native), or a predefined variable.

Enter only the variable's name. Do not include the % symbols that would accompany the variable when you use it.

4 Provide the value for the variable.

The value is what the variable resolves to. It can also be another variable for nesting variables.

To ensure that extraction takes place, provide an absolute path to the Subscriber. For example, if the path is only the data: volume, make sure the colon (:) is included, because it is a necessary part of the full path.

5 Provide a description (optional), then click *OK*.

6 Repeat Step 2 through Step 5 to define another variable for this Subscriber.

7 Click *OK* to exit the Subscriber's properties.

9.3.3 Defining Variables for a Server Software Package

1 In ConsoleOne, right-click a software package, then select the *Variables* tab.

2 Click *Add*.

New Variable #1 is defaulted in the *Variables* column.

Enter only the variable's name. Do not include the % symbols that would accompany the variable when you use it.

3 To provide a different name for the variable, use the Backspace key to delete the default name, type a new variable name, then press the Tab key.

The name can be user-defined, an environment variable (Java or native), or a predefined variable.

4 Provide the value for the variable.

The value is what the variable resolves to. It can also be another variable for nesting variables.

5 Repeat Step 2 through Step 4 to define another variable.

6 Click *OK* to exit the Server Software Package's properties.

9.4 Viewing All Variables in iManager

In Novell iManager, you can view all of the variables that are being used by a Server Management object:

1 In iManager, under the *ZENworks Server Management* role, click *Remote Web Console*.

2 Select a Distributor or Subscriber object, then click *OK*.

3 In the *Display* list box, select *Policy/Package Agent*.

This automatically displays the *Configurations* tab.

4 Scroll down to view the variables listed to the right of *Variables*.

All variables that can be used by the object are listed.

9.5 Using a Variable to Change a Subscriber's Console Prompt

The Subscriber can use the value of the PROMPTvariable as its server console prompt.

1 In ConsoleOne, right-click a Subscriber object, then click *Properties*.

2 Select the *Variables* tab, then click *Add*.

3 In the Variables dialog box, provide information for the following fields:

Variable: Enter PROMPT as the variable name.

Value: Type the prompt text to be displayed. For example, %SERVER NAME% Subscriber could display as:

```
Provo_01 Subscriber >
```

Description: Provide a meaningful note (optional).

4 Click *OK* twice.

9.6 Using Variables to Control File Extraction

You can use variables to control the location that files are extracted on the Subscriber. Any destination can be used as a variable defined in a Subscriber object by encapsulating it with the percent (%) symbol.

IMPORTANT: Any variable value specified in the Tiered Electronic Distribution policy is a default value and is overridden by variable values set in a Subscriber object.

For the location where files are extracted, the destination root is identified in the File Grouping dialog box as a directory named destroot. This is the top-level directory used by a Subscriber to determine where to extract the file. The dialog box lets you build groups of directories under the `\destroot` directory.

You can specify the destination root as a known location (for example, %APP_DIR%). You can then go to the Variables tab on the Subscriber object and specify a value for this variable.

For example, variable APP_DIR would have the value:

```
sys:\apps
```

To use a variable to set the location that files are extracted to:

1 In ConsoleOne, right-click the Subscriber object, then click *Properties*.

2 Select the *Variables* tab, then click *Add*.

3 Provide the name of the variable.

The name can be user-defined, an environment variable (Java or native), or a predefined variable.

4 Provide the value for the variable.

The value is what the variable resolves to. It can also be another variable for nesting variables.

To ensure that extraction takes place, provide an absolute path to the Subscriber. For example, if the path is only the data: volume, make sure the colon (:) is included, because it is a necessary part of the full path.

5 Provide a description (optional), then click *OK* twice to exit the properties.

6 Create a new Distribution object.

For information, see .

7 In the Distribution object's properties, select the *Type* tab, select *File in the Select Type* drop-down box, then click *New Target*.

8 Replace the default %DEST_VOLUME% with the variable name, then click *OK* as necessary to exit the properties.

A directory named `\dest_volume` is created by default in the *Destination* column. You should select this directory to change the destination root. To select it, select the actual directory name (destroot). You can then specify a known location or use a variable with surrounding percent symbols.

ZENworks Database

<div align="right">

10

</div>

The following sections provide information for understanding and using the Novell® ZENworks® Server Management database in Policy and Distribution Services:

10.1 Understanding the ZENworks Database

The following sections provide an understanding of the ZENworks database:

10.1.1 The Database Engine

ZENworks Server Management is shipped with the Sybase database engine. This can only be installed once on a server. However, you can install Sybase to multiple servers.

Oracle* and SQL are not supported.

10.1.2 The Database File

Policy and Distribution Services uses a Sybase database file named `zfslog.db`. Server Management can function normally without the database, because it uses `zfslog.db` only to log information for Policy and Distribution Services reporting.

`Zfslog.db` is normally located in the `\zenworks\pds\db` directory on a server. Its location is determined when using the installation program. It can reside on both NetWare® and Windows servers.

10.1.3 The Database Object

A Novell eDirectory™ database object is created during installation, named Server Management Database_*server_name*. In its properties, the location of the database file (`zfslog.db`) is listed, if established during installation; otherwise, you can configure the ZENworks Database policy

(Service Location Package) to specify the database object. The location and policy are necessary for the database file to be found for logging information.

10.1.4 Running the Database

On NetWare servers, the database is run by using the `mgmtdbs.ncf` file (located in the `sys:\system` directory), which is executed from `autoexec.ncf`.

On Windows servers, the database is run by using the Novell Database - Sybase service.

10.1.5 Database Caching

Database files can become very large, which is why a 32 MB cache is recommended on the server where you are running the database. Caching improves server performance because of how frequently information is logged to `zfslog.db`.

10.1.6 Database Information

`Zfslog.db` is used by Policy and Distribution Services to log successes and failures for the Server Policies or Tiered Electronic Distribution components. You can purge policy information automatically according to a policy setting. You can purge Tiered Electronic Distribution information manually from the database object. For information on purging, see Section 10.5, "Purging the Database," on page 368.

`Zfslog.db` does not contain any configuration information.

The information listed in Table 10-1 is written to `zfslog.db` by the agents:

Table 10-1   *Agents that Write to the Database*

Agent	Information
Policy/Package	Failed and successful policies
	Discovered and unenforceable policies
	Down Server policy status
	Server Software Packages and components
Distributor	Distribution status:
	◆ When built, sent, and extracted
	◆ Successes (plus reasons) of builds and extractions
	◆ Failures (plus reasons) of a build, send, receive, and extraction
	Subscriber status
	Revision histories

For information on obtaining reports on the database information, see Chapter 11, "Reporting," on page 371.

The following provides information on gathering data for the database:

- A Distributor keeps track of each Subscriber in its routing hierarchy, so it knows which parent Subscribers have received a Distribution.

- The Distributor knows which Subscribers are at the end of a particular route, so it can know if Subscribers have not received a Distribution because a Subscriber higher up in the hierarchy failed to receive the Distribution.

- Subscribers send messages directly to the Distributor indicating that they have received a Distribution. The Distributor does not return a confirmation that it received the Subscriber's message.

- If a Distributor is not running when a "Successfully Received" message is sent from a Subscriber, this information is not written to the database. Because a message receipt confirmation is not received by the Subscriber, it does not re-send the message.

10.1.7 Coexisting Databases

You can have multiple Server Management databases in the tree. The number you have depends on whether you want consolidated reporting and can live with the additional network traffic in a WAN environment.

If you do not require consolidated reports, you can install one database file and object on different servers for each of your WAN segments. This eliminates writing to the database file over a WAN link by the Distributor.

For the server selected for a database file, you should not install a ZENworks Desktop Management database when a ZENworks Server Management database exists for Policy and Distribution Services. The Desktop Management database file replaces the ZENworks Server Management database file, causing all ZENworks Server Management database information to be lost. However, you can install a ZENworks Server Management database where a Desktop Management database exists and not lose any Desktop Management database information.

However, the databases for Management and Monitoring Services, Server Inventory, and Policy and Distribution Services can coexist on a server, because their database files use different filenames. You only need to name the database objects differently from each other, because they all have the same default object name of ZENworks Database.

10.2 Determining How Many Databases You Need

You can install the database to both NetWare and Windows servers.

The installation program checks the version of the Sybase engine before updating it. If it doesn't exist, or is an older version, Sybase software is installed.

IMPORTANT: Make sure you select a server for the database where you are installing the Subscriber/Policies option. The Purge Database option in the ZENworks Server Management policy (Distributed Server Package) works only if the Policy/Package Agent software and the `zfslog.db` file are located on the same server.

The installation program automatically creates a database object for each instance of the database that is installed, and you can select a database for the object during installation. You can install only

one instance of the database per run of the installation program. The database object is installed to the same eDirectory container as the Server object for the server where the database file, zfslog.db, is also installed.

Review the following to understand whether to have multiple database files:

10.2.1 Database Logging and Tiered Electronic Distribution Reporting

Policy and Distribution Services can function normally without using a Server Management database, because it uses the zfslog.db file to only log information for reports. Zfslog.db for Policy and Distribution Services does not contain any configuration information. To obtain Distribution status information, use the Tiered Distribution View and the Remote Web Console options under the *ZENworks Server Management* role in Novell iManager. Policy information (written to the database file) can be obtained through the canned reports available from the Tools menu in Novell ConsoleOne®.

The Distributor Agent writes its distribution status information (built, sent, received, extracted) and Server Software Package installation information to the database file (zfslog.db). The Policy/Package Agent writes policy enforcement successes and failures to the database file. This database information is used for the Policy and Distribution Services reports.

For the agents to know which database file they should write to, policies must be created for them. If not established during installation, the ZENworks Database policy can be used to associate (using the Service Location Package) so that the Distributor Agent can know where to write. The ZENworks Database policy is distributed (using the Distributed Server Package) to the Subscriber for the Policy/Package Agent to know where to write.

Policy and Distribution Services provides six predefined reports for the Server Policies component and four for the Tiered Electronic Distribution component. The report information is obtained from information logged to its database file. The reports listed in Table 10-2 are available:

Table 10-2 *Policy and Distribution Services Reports*

Server Policies Reports	Tiered Electronic Distribution Reports
Discovered Policies	Distribution Detail
Down Server Policy	Revision History
Packages	Revision History Failure
Failed Policies	Subscriber Detail
Successful Policies	
Unenforceable Policies	

A selected report displays all of the applicable Server Policies or Tiered Electronic Distribution information currently logged in the database. The criteria you can specify for a report include date ranges, specific Distributions, Distribution versions, and so on.

You might want multiple databases for specialized reporting. For more information, see "Advantages" on page 363.

For information on reporting, see Chapter 11, "Reporting," on page 371.

10.2.2 Multiple Databases

Policy and Distribution Services supports multiple instances of the Server Management database per tree. However, we recommend that you install only one instance of the database per tree. Review the following:

- "Advantages" on page 363
- "Distributor Object Contexts and Multiple Databases" on page 363
- "Determining Whether You Need Multiple Databases" on page 364

Advantages

The advantage in having only one database is that the Distribution information provided by all of the Distributor Agents and Policy/Package Agents can be displayed in a single report.

For example, with a single database, your software package information can be contained in one report:

- The Distributor Agent's information on building and sending the Software Package Distribution
- The Policy/Package Agent's information on extracting and installing the software package

The advantages in having multiple databases are:

- Minimizing traffic over slow WAN links

  For example, having a separate database for Policy/Package Agent logging on its server's side of a WAN link.

- Providing individual databases for specialized reporting

  For example, if you have one database for the Distributor Agent (distributions) and one for the Policy/Package Agent (policies), the build and send information for the Software Package and Policy Package types of Distributions is written to the distributions instance of the database, and the software package installation and policy enforcement information is written to the policies instance of the database.

Distributor Object Contexts and Multiple Databases

One `zfslog.db` file can receive log entries from multiple Distributors, and a Distributor can only log to one `zfslog.db` file. The following explains why:

- For a Distributor Agent to locate a database file, it must have a ZENworks Database policy (Service Location Package) associated with a context above the Distributor's object that points to the Database object, which contains the file's location in its properties. (Distributors receive their policies through association.)
- If you have separate databases installed on two or more of your Distributor servers, each database requires its own ZENworks Database policy for locating it (the policy points to the database's object, which contains its file's location).

- Only one Service Location Package (which contains the ZENworks Database policy) can be associated with a given context, such as the container holding your Distributor objects.

- Because only one Service Location Package can be associated with a given context, you must install your Distributor objects to different contexts to have multiple Distributors writing to their individual database files. Each Distributor would need its own database location policy that is associated with its own parent container.

  For ease of management, you can keep your Distributor objects near each other by creating individual containers for each of them under the container where you usually place all of them. Then you can associate the different Service Location Packages with their appropriate Distributor's unique parent containers.

- To have all of your Distributors write to the same database file, place each of their Distributor objects somewhere under the container where you associate the Service Location Package. They would all use the same database location policy.

Determining Whether You Need Multiple Databases

Consider the following to determine how many databases to have in the tree:

- **WAN traffic:** Tiered Electronic Distribution does not perform a large number of database updates, so the actual impact on system resources should be minimal. The greatest impact could be the time it takes to perform the transaction. However, if you have slow WAN connections, you might not want database logging to occur over the WAN.

- **Multiple Distributors:** If you have multiple Distributors in the tree, you can have one database for each, or have them share one or more databases. The type of Distributor reporting you want should determine whether to have a separate database for each. For example, are your Distributors specialized in the types of Distributions they'll send?

- **Consolidated reporting:** To have only one report for all of your Tiered Electronic Distribution information, install only one database object and file and have all Distributors log to that one file, regardless of WAN traffic considerations. Use the ZENworks Database policy (Service Location Package) to direct all Distributors to that database file.

- **Specialized reporting:** You might want reports that are specific to a region or group of servers. You can install a database object and file for each such region and have the Distributors in those regions or server groups log to that database. Use a separate ZENworks Database policy (Service Location Package) to direct each Distributor to its desired database file.

10.3 Installing and Connecting to the Server Management Database

You should install the Server Management database on a server where policies are enforced. This is required so that you can use the ZENworks Database policy to locate the database file, `zfslog.db`.

The Server Management Database object is automatically created in the tree when you run the installation program and select a server for the database.

The installation program can install only one database at a time. To install additional databases to the tree, you need to perform the steps in the following sections for each database to be installed.

Perform the steps in the following sections to install and set up the database:

- Section 10.3.1, "Installing the Database," on page 365
- Section 10.3.2, "Connecting to the Database," on page 366

10.3.1 Installing the Database

To install a Policy and Distribution Services database:

1 On a workstation, insert the *ZENworks 7 Server Management with Support Pack 1 Program* CD.

The startup screen is displayed. If the startup screen is not automatically displayed after inserting the CD, you can start it by running `winsetup.exe` at the root of the CD.

> **IMPORTANT:** Installation from a remote CD is not supported unless there is a drive mapped on the workstation to that CD. For example, if you place the CD in a Windows server CD drive, then run the installation from a workstation, you must have a drive mapped to the CD drive of that Windows server.

2 Select the *Server Management* option.

3 Click *Policy-Enabled Server Management* to start the installation program.

4 If you agree with the Software License Agreement, click *Accept*, then click *Next* to display the Installation Type page; otherwise, select *Decline* and click *Cancel* to exit.

5 On the Installation Type page, click *Next* to perform a new installation and display the Installation Options page.

6 On the Installation Options page, click *Next* to accept the defaults and display the eDirectory Tree for Creating Objects page.

7 Browse and select the tree to install to (you can only select one tree), then click *Next*.

The tree name is not case sensitive.

8 On the Server Selection page, click *Add*, then browse for the server where you want to install the database.

You can select only one server per run of the installation program.

You might want a database for each Distributor to write its own information to. However, Distributors can share a database. Because the Distributor writes information to the database for all Tiered Electronic Distribution objects, you should install the database on the same server as the Distributor to minimize network traffic.

> **IMPORTANT:** Make sure you select a server for the database where you are installing policies. The Purge Database option works only if the `zfs.ncf` and `zfslog.db` files are on the same server.

9 Under *Additional Options*, select the *Server Management Database* check box to enable it, then click *Next* to display the File Installation Paths and Options page.

The installation program checks all mounted volumes on the server to see if `zfslog.db` exists. If not, both the file and the database object are installed. If the file exists, the database object is still installed.

10 Click *Next* to accept the defaults and display the Database Settings page.

11 To change the default path to the database file, edit the *Database Path* field.

> **IMPORTANT:** Because the database file can become very large, we recommend that you change the default NetWare volume from sys: to another volume on that server.

12 Accept the other defaults on the Database Settings page by clicking *Next* to display the Policy and Distribution Services Database Logging page.

13 To determine logging for the Server Management database that you configured in a previous installation page, select one of the following:

Log to an existing Server Management database: Select an existing database file for logging by browsing for and selecting the database object to associate it with.

Log to a Server Management database that will be installed: The database object name that you configured in a previous installation page is the default. However, you can browse for and select an existing database object.

Do not log to a Server Management database: You can elect to not log to a database at this time, even though you have configured a database in the previous installation page.

14 On the Summary page, review your selections, then click *Finish*.

The installation program now copies files and installs the database objects.

> **WARNING:** If you click Cancel, none of the work you did in the installation program is saved.

After the installation has finished, you can check the installation log file (see Step 10) to see if any components failed to install.

The ZENworks Database policy is automatically created and configured during installation of this new database.

15 Continue with Section 10.3.2, "Connecting to the Database," on page 366.

10.3.2 Connecting to the Database

To make sure that the database can be written to by the Policy/Package Agent:

1 On a server, load the agent by doing the following:

Server Platform	Agent Startup Method
Windows	1. Open the Control Panel.
	2. Click Admin Tools, then click Services.
	3. Click Novell ZENworks Service Manager, then click Start.
NetWare	`sys:\zenworks\pds\smanager\zfs.ncf`
Solaris or Linux	`/etc/init.d/Novell-ZfS Start`

Note whether a message is displayed indicating that the agent has connected to the database.

2 To determine whether the agent is writing to the database, do the following:

  2a At a NetWare server's console prompt, view the monitor while the agent is loading.

     A message should display that states whether the agent connected with the database.

  2b If the message indicates that the agent did not connect to the database, you should check the following:

     ◆ Is the database is running on the server?

     ◆ Is there a database object that has its Policy/Distribution Management tab set up with the server where the database file is installed?

     ◆ Is there an effective ZENworks Database policy pointing to the database object?

10.4 Creating a ZENworks Database Object

The ZENworks Database object might not exist if you had inadvertently deleted the object.

If the database object does not exist in the tree because you didn't originally install it, you must use the GUI installation program to create it. For more information, see "Starting the Installation Program" in the *Novell ZENworks 7 Server Management Installation Guide*. Follow only the steps and select only the options that are necessary to create the database.

To re-create a database object that was inadvertently deleted:

1 In ConsoleOne, right-click a location in the tree for the database object, click *New > Object*, then click *ZENworks Database*.

2 Provide a database name.

3 Select the *Define Additional Properties* check box, then click *OK*.

4 On the *ZENworks Database* tab, select either the *Server DN* or *Server IP Address* option.

One of these location IDs could already be the default. If not, provide the information for the server where zfslog.db resides.

5 Select the *eDirectory Rights* tab, click *Trustees of This Object*, click *Add Trustee*, then select *[Public]*.

The database object must be assigned a trustee of Public, or the Policy/Package Agent displays messages that it cannot connect with the database or read the ZENworks Server Management policy.

6 Click *OK*.

If you click *Cancel*, none of the information you added or changed on any of the tabs is saved. However, the database object remains on the tree.

7 Set up the ZENworks Database policy.

For steps to specify the location of a database, see "ZENworks Database" on page 235.

8 Associate the Service Location Package with a container above where the Distributor object resides.

10.5 Purging the Database

Because Policy and Distribution Services logs all successes and failures for the Server Policies or Tiered Electronic Distribution components, `zfslog.db` can quickly grow in size. Therefore, you should periodically purge this database file.

The following database information types are purged using different methods:

10.5.1 Tiered Electronic Distribution Information

To manually purge a selected database of all Tiered Electronic Distribution information older than a specific date and time:

1 In ConsoleOne, right-click the database object, then click *Purge*.

2 In the Purge Database dialog box, select a date and time, then click *OK*.

   Records older than the date entered are purged from the database.

When the purge has been completed, a dialog box is displayed indicating that the purge was successful.

10.5.2 Server Policies Information

Purging of policy information is done automatically according to how you configure the ZENworks Server Management policy and which of the following events occurs:

- A server is restarted where the Policy/Package Agent is running that writes policy information to the database.

- Server Management is restarted on a server where the Policy/Package Agent is running that writes policy information to the database.

- On a server where The Policy/Package Agent is running that writes policy information to the database, the Policy/Package Agent is manually refreshed by typing the REFRESH command on the `ZENworks Server Management` console prompt.

   The REFRESH command or Refresh option only causes database purging if given on a server where the database file resides.

In each of these events, the database file that is purged is the one written to by the associated Policy/Package Agent.

To set up policy information purging:

1 In ConsoleOne, do one of the following:

   - If you want to use a different policy package schedule for purging information than an existing Distributed Server Package (see Step 5) is using, in ConsoleOne click *File > New > Policy Package*, select *Distributed Server Package*, and provide a name that identifies its purpose. For example, Purge_Policy_*server_name*, where *server_name* is the server where the `zfslog.db` file resides.

- To use the same policy package that has other policies enabled, in ConsoleOne right-click the existing Distributed Server Package, then click *Properties*.

  In this case, we recommend that in Step 5 you select *Run Immediately* for the package schedule. That way, any change you make to the number of days in Step 3 is immediately available the next time a purge is triggered.

2 In the Distributed Server Package, select the *ZENworks Server Management policy* check box, then click *Properties*.

3 Select the *ZENworks Server Management Configuration* tab and select a number of days.

The default is 100 days. Records older than the number of days that you determine are purged.

Select a number that maintains the desired database file's size. The amount of policy-related information accrued in the database is determined by how often you have policies being run by servers writing to this database. Depending on how frequently you purge the database, you may need to experiment over time to determine the optimum number of days.

4 Click *OK* to close the policy's properties.

5 To set the package schedule, do one of the following:

- To accept the default package schedule, which is Run Event: System Startup, click *OK* to close the package's properties.

- To change the default schedule, click Edit, select a schedule, then click *OK* twice to close the package's properties.

  For more information on the schedules, see Section 4.7, "Scheduling Policies," on page 239.

The package schedule determines when any configuration changes that you make are available. For example, if you previously selected Event: System Startup for the package schedule and then later changed the 100 days to 60, that change is not recognized if the Policy/Package Agent is refreshed to trigger purging. It is only recognized after system startup occurs.

6 Create a Policy Package Distribution for this policy.

For more information, see "Creating and Configuring the Distribution" on page 57.

7 Send the Distribution to the Subscriber server where the `zfslog.db` file resides.

For more information, see Section 1.2.6, "Sending the Distributions," on page 60.

If this policy package is dedicated the ZENworks Server Management policy for purging, you need to send this Distribution only to each server where a database file resides, because you need just one instance of this policy per database file.

Reporting 11

Novell® ZENworks® Server Management provides predefined reports for the Policy and Distribution Services components:

- Section 11.1, "Understanding Policy and Distribution Services Reporting," on page 371
- Section 11.2, "Report Descriptions," on page 373
- Section 11.3, "Generating Reports," on page 377
- Section 11.4, "Creating Customized Reports," on page 379

11.1 Understanding Policy and Distribution Services Reporting

Review the following:

- Section 11.1.1, "Reporting Categories," on page 371
- Section 11.1.2, "Reporting Scope," on page 372
- Section 11.1.3, "Accessing Reports," on page 372
- Section 11.1.4, "Creating and Storing Report Information," on page 372

11.1.1 Reporting Categories

Server Policies has six predefined reports, and Tiered Electronic Distribution has four. For details on each predefined report, see Section 11.2, "Report Descriptions," on page 373.

The following sections describe the purposes of the Policy and Distribution Services reports:

- "Purposes for the Server Policies Reports" on page 371
- "Purposes for the Tiered Electronic Distribution Reports" on page 371

Purposes for the Server Policies Reports

Server Policies reports show which servers have processed which policies, when they were processed, and if their enforcement was successful.

Purposes for the Tiered Electronic Distribution Reports

The Distribution-level reports show the view from the Distributor side and are very useful for checking which Subscribers succeeded or failed to receive and extract a particular Distribution. The Subscriber reports are used to determine which Distributions a single Subscriber has received.

Reporting gives very detailed information regarding which nodes succeeded. All known error conditions are caught and error conditions are reported to the database. However, when a process status is in progress, errors can occur or failures can occur on the node that are not caught (for example, the machine went down or the process was killed).

Subscribers that did not attempt to receive the Distribution (because they were not set up correctly or were not running) do not have information displayed on the report. You can compare the number expected against the actual numbers and look for missing Subscribers on the report. After Subscribers are set up and have been functioning, this should not be a common problem.

11.1.2 Reporting Scope

A selected report displays all of the applicable Server Policies or Tiered Electronic Distribution information currently logged in the database. There are options for defining the selection criteria for the data that will appear on some reports, such as date ranges, or for selecting Policy Package objects or Tiered Electronic Distribution objects.

11.1.3 Accessing Reports

There are two access points for Policy and Distribution Services reports:

Via the Object

1 In ConsoleOne, right-click a ZENworks Database object.

2 Click *Reporting*.

The report dialog box for the Policy and Distribution Services canned reports is displayed.

Via the Menus

1 In ConsoleOne, click *Tools > ZENworks Reports*.

2 Click *Reporting*.

The report dialog box for the Policy and Distribution Services canned reports is displayed.

11.1.4 Creating and Storing Report Information

A Policy and Distribution Services database file (`zfslog.db`) is used to store the report information. After you have installed and run the database and data has been placed in `zfslog.db`, Policy and Distribution Services reporting is enabled.

The Policy/Package Agent running on each Subscriber server writes Server Policies information to the database. The Distributor Agent writes the Tiered Electronic Distribution information and the Server Software Package information to the database.

Each Distributor may normally have its own ZENworks Database object and database file (`zfslog.db`), so report information could be given only for the particular Distributor associated with the ZENworks Database object selected.

Information is logged to the `Zfslog.db` file when any of the following actions have occurred:

- The ZENworks Database policy (Service Location Package) has been configured and enabled (see "ZENworks Database" on page 218)
- The ZENworks Database policy (Distributed Server Package) has been configured and enabled (see "ZENworks Database" on page 218)

- The Policy/Package Agent has either been refreshed from the server console or Server Management has been restarted

  The ZENworks Database policy (contained in the Distributed Server Package) must already have been received and extracted on the Subscriber server before the Policy/Package Agent can log to the database file.

- The Distributor Agent has been restarted (not refreshed) after the ZENworks Database policy has been enabled (which includes associating it with the Distributor object's container).

11.2 Report Descriptions

The following sections describe the Policy and Distribution Services reports:

- Section 11.2.1, "Tiered Electronic Distribution Reports," on page 373
- Section 11.2.2, "Server Policy Reports," on page 375

11.2.1 Tiered Electronic Distribution Reports

There are four predefined Tiered Electronic Distribution reports:

- "Distribution Detail Report" on page 373
- "Revision History Report" on page 374
- "Revision History Failure Report" on page 374
- "Subscriber Detail Report" on page 374

Distribution Detail Report

Displays a detailed, time-line history of Distributions for the selected Subscribers, including:

- Distributions Sent
- Distributions Received
- Distributions Extracted (including start time, end time, and completion code)

Sorting is by time; grouping is by Distribution name and version.

The report criteria include:

- **Subscriber:** If you selected a Subscriber object, it appears in the Subscriber field and the report only displays information for the receive and extract actions performed by this Subscriber. Information for parent Subscribers also displays a Received Stage heading.

  If you selected the Database or Distribution object, the report includes all actions that have occurred with a Distribution. In other words, information for all Subscribers involved is displayed.

- **Latest version only:** Deselect to include versions that are within the specified date range.
- **Select the date range criteria for the report:** Specify the range.

Revision History Report

Displays a history of a Distribution package's versions, including:

- Distribution (DN of package)
- Distributor (DN of object)
- Version Number
- Creation Date/Time
- Distribution Size

Sorting is by version number.

The report criteria include:

- **Distribution:** If you selected a Distribution object, it appears in the Distribution field. If you selected the Database object, you need to browse for the Distribution object.

Revision History Failure Report

Displays the versions of the Distribution that failed during creation, including:

- Distribution (DN of package)
- Distributor (DN of object)
- Creation Date and Time
- Error Description

Sorting is by version.

The report criteria include:

- **Distribution:** If you selected a Distribution object, it appears in the Distribution field. If you selected the Database object, you need to browse for the Distribution object.

Subscriber Detail Report

Displays status information for the Subscribers that received the Distribution, including:

- Distribution and Version
- Subscriber (DN of object) and Subscriber's Address
- Channel Name
- Source (DN of Distributor)
- Stage
- Status
- Date and Time
- Error Description

Sorting is by Subscriber/Parent Subscriber, then Stage.

The report criteria include:

- **Distribution:** If you selected a Distribution object, it appears in the Distribution field. If you selected the Database object, you need to browse for the Distribution object.
- **Version number:** If Distribution versions exist, you can choose one from the drop-down menu. Select All to include all versions.
- **Distribution stage:** You can select All, Extract, or Receive.
- **Distribution status:** You can select All, Success, or Not Success.

11.2.2 Server Policy Reports

For all server policy reports, the default date ranges are for the current date (from midnight to midnight).

There are six predefined server policy reports:

Discovered Policies Report

Displays the servers that have discovered policies within the specified packages, including:

- Package (DN)
- Server DN
- Server Name
- OS Name and OS Version
- Date/Time of Discovery

Sorting is by package, then by context/server name, maintaining the tree's hierarchy. For example, `myserver.servers.novell` is sorted novell, servers, myserver.

The report criteria include:

- **Package:** Select a policy package from the drop-down list or select All.
- **Policy type:** You can select All, Server Down Process, Scheduled Down, SNMP Trap Targets, Community Strings, Set Parameters, Script, Text File, Scheduled Load/Unload, or Database Location.
- **Select the date range criteria for the report:** Specify the range.

Server Down Process Report

For a selected server or all servers in the tree, displays Server Down Process policy information, including:

- Down Action and Code for each policy

Sorting is by server name only.

The report criteria include:

- **Server:** Select a server from the drop-down list or select All.
- **Select the Date Range Criteria for the Report:** Specify the range.

Failed Policies Report

For all servers in the tree, displays all policies that have failed, including:

- Package (DN)
- Server DN
- Server Name
- OS Name
- Date/Time of Failure
- Reason for Failure (Description)

Sorting is by context/server name, maintaining the tree's hierarchy. For example, `myserver.servers.novell` is sorted novell, servers, myserver.

The report criteria include:

- **Package:** Select a policy package from the drop-down list or select All.
- **Failure type:** You can select All, Failed, Unenforceable, or Partial Enforcement.
- **Policy type:** You can select All, Server Down Process, Scheduled Down, SNMP Trap Targets, Community Strings, Set Parameters, Script, Text File, Scheduled Load/Unload, or Database Location.
- **Select the date range criteria for the report:** Specify the range.

Packages Report

Displays information on Server Software Packages and their components, including:

- Success status of each package
- Success status of each component

Sorting is by context/server name, maintaining the tree's hierarchy. For example, `myserver.servers.novell` is sorted novell, servers, myserver.

The report criteria include:

- **Package:** Select a software package from the drop-down list or select All.
- **Server:** Select a server from the drop-down list or select All.
- **Select the date range criteria for the report:** Specify the range.

Successful Policies Report

For all servers in the tree, displays all policies that have been successfully enforced, including:

- Package (DN)
- Server DN
- Server Name
- OS Name
- Date/Time of Run
- Action Code

Sorting is by context/server name, maintaining the tree's hierarchy. For example, `myserver.servers.novell` is sorted novell, servers, myserver.

The report criteria include:

- **Package:** You can specify a single policy package or select All.
- **Success type:** You can select All, Change, or No Change.
- **Policy type:** You can select All, Server Down Process, Scheduled Down, SNMP Trap Targets, Community Strings, Set Parameters, Script, Text File, Scheduled Load/Unload, or Database Location.
- **Select the date range criteria for the report, from/to:** Specify the range.

Unenforceable Policies Report

Displays all unenforceable policies because of the absence of an enforcer on a server for all servers in the tree, including:

- Package (DN)
- Server DN
- Server Name
- OS Name and OS Version

Sorting is by package, then by server name.

The report criteria include:

- **Package:** Select a policy package from the drop-down list or select All.
- **Select the date range criteria for the report:** Specify the range.

11.3 Generating Reports

1 In ConsoleOne, right-click the ZENworks Database object.

   The ZENworks Database object must be one where its Policy/Distribution Management tab (not the Inventory Management tab) is properly configured.

2 Click *Reports*.

3 Select a report:

- Server Policy Reports

  Discovered Policies
  Failed Policies
  Packages
  Server Down Process Policy
  Successful Policies
  Unenforceable Policies

- Tiered Electronic Distribution Reports

  Distribution Detail
  Revision History
  Revision History Failure
  Subscriber Detail

4 Select the reporting criteria.

If you need more detail on reporting criteria or content, see Section 11.2, "Report Descriptions," on page 373.

5 Click *Run Selected Report*.

The View Report dialog box is used to display the generated report. The dialog box has the following features:

- To expand how much of the report you can view on screen, resize the dialog box.

  The report is displayed in landscape orientation for printing purposes.

- Use the following navigation options to move through the report:

  First Page
  Previous Page
  Next Page
  Last Page
  Go To Page

- To print the report, click File > Print.

  Your default printer is selected.

- To export the report, click File > Export Report.

  You can export to the following formats:

  Text
  HTML
  PDF
  SDF

11.4 Creating Customized Reports

Using the following database information, you can create custom reports for the Server Policies and Tiered Electronic Distribution components.

However, for Tiered Electronic Distribution objects such as a Subscriber or the External Subscriber, you should use ZENworks reporting options (see Chapter 11, "Reporting," on page 371) or iManager (Chapter 2, "Novell iManager," on page 63) for determining the status of Distributions or policies.

The database file (zfslog.db) contains the following information:

11.4.1 Default Sybase Database User ID and Password

The Sybase database (zfslog.db) that ships with Server Management has the following default user ID and password:

User ID: dba

Password: sql

11.4.2 Server Policies Database Contents

Following are the database table definitions for server policies:

SERVERS

Contains one record for each server running the Policy/Package Agent.

Table 11-1 *Servers Field Names*

Field Name	Type	Use	
SERVERID	integer	not null	Unique number that is automatically assigned.
SERVERNAME	varchar	not null	The short name of the server as seen on the console prompt.
SERVERDN	varchar		DN of the Server object in eDirectory (dot separated).
REVERSEDN	varchar	not null	SERVERDN in reverse order and backslash (\) delimited.
OSNAME	varchar		Name of the operating system, such as NetWare 5.1.
OSVERSION	char		Version of the operating system, such as 5.1, 6.0, and so on.
TREENAME	varchar		Name of the eDirectory tree containing the server.

Primary key (SERVERID)

SERVERIP

Contains one record for each server running the Policy/Package Agent.

Table 11-2 *ServerIP Field Names*

Field Name	Type	Use	
SERVERIPKEY	integer	not null	Assigned automatically: Default Auto increment.
SERVERID	integer	not null	Links to the SERVERS table.
IPADDRESS	varchar	not null	Server's IP address.

Primary key (SERVERID) REFERENCES SERVERS
Primary key (SERVERIPKEY)

PACKAGES

Contains one record for each version of a software package that the Policy/Package Agent has attempted to process.

Table 11-3 *Packages Field Names*

Field Name	Type	Use	
PACKAGEGUID	char	not null	Assigned automatically.
PACKAGENAME	char		Name of `.cpk` file or policy package.
PACKAGEDESC	char		Description contained in a Server Software Package component.
PACKAGEVERSION	char		Version of the software package.
BUILDDATE	integer		Date the software package was compiled.

Primary key (PACKAGEGUID)

POLICIES

Contains one record for each policy or policy package combination.

Table 11-4 *Policies Field Names*

Field Name	Type	Use	
POLICYID	integer	not null	A globally unique ID.
POLICYDN	varchar		The DN of the eDirectory policy object.
POLICYPACKAGE	varchar		The DN of the policy package the policy belongs to.
POLICYCLASS	varchar		The class or type of policy. For definitions, see "Valid Entries for POLICYCLASS" on page 381.
POLICYTREENAME	varchar		The name of the tree the policy object is in.

Primary key (POLICYID)

Valid Entries for POLICYCLASS

zenZFSServerDowningPolicy
zenZFSScheduleDownPolicy
zenZFSSetServerParamPolicy
zenZFSServerScriptPolicy
zenZFSTextFilePolicy

zenZFSScheduledRunPolicy
zenZFSZFSPolicy
zenZFSCommunityPolicy
zenZFSSNMPTrapTargetPolicy
zenZFSSMTPHostPolicy
zenZFSDatabaseLocationPolicy
zenZFSLicenseLocationPolicy
zenZFSTEDPolicy

POLICYACTION

Contains one record for each action performed.

Table 11-5   *PolicyAction Field Names*

Field Name	Type	Use	
POLICYACTIONKEY	integer	not null	Assigned automatically: Default Auto increment.
POLICYID	integer	not null	Links to the POLICIES table.
SERVERID	integer	not null	Links to the SERVERS table.
CREATIONDATE	timestamp		Time stamp of the action.
DESCRIPTION	varchar		Undefined string describing an error.
CODE	integer		Code representing the result of the action. For definitions, see "Valid Entries for CODE" on page 382.
ACTIONCODE	integer		The action being performed. For definitions, see "Valid Entries for ACTIONCODE" on page 383.

Primary key (POLICYACTIONKEY)

Valid Entries for CODE

RC_POL_SUCCESS	= 0
RC_POL_PARTIAL_SUCCESS	= 1
RC_POL_FAILURE	= -1
RC_POL_EMPTY	= -2

Exception: If the value in the ACTIONCODE field is AC_POL_DOWN_CONNECTIONS or AC_POL_DOWN_DISCONNECTIONS, then the value of CODE is either the current number of active connections, or the number of forced disconnects.

A number 1 in the CODE field can mean one of the following:

- There was a partial success
- There is one active connection
- There was one forced disconnect

This is because the meaning of the entry in the CODE field is determined by the content of the ACTION CODE field.

Valid Entries for ACTIONCODE

AC_POL_DISCOVERED	= 101
AC_POL_SCHEDULED	= 102
AC_POL_APPLIED	= 103
AC_POL_APPLIED_CHANGE	= 104
AC_POL_NO_ENFORCER	= 105
AC_POL_DOWN_CONNECTIONS	= 106
AC_POL_DOWN_DISCONNECTIONS	= 107
AC_POL_DOWN_UNLOAD	= 108
AC_POL_DOWN_EMAIL	= 109
AC_POL_DOWN_NOTIFY	= 110
AC_POL_DOWN_CANCELED	= 111
AC_POL_DOWN_IGNORED	= 112
AC_POL_DOWN_REQUESTED	= 113

PACKAGEACTION

Contains one record for each action taken on a Server Software Package.

Table 11-6 *PackageAction Field Names*

Field Name	Type	Use	
PACKAGEACTIONID	integer	not null	Assigned automatically: Default Auto increment.
PACKAGEGUID	char	not null	Links to the PACKAGES table.
SERVERID	integer	not null	Links to the SERVERS table.
CREATIONDATE	timestamp		Time stamp of the action.
DESCRIPTION	varchar		For definitions, see "Valid Entries for DESCRIPTION" on page 384.

Field Name	Type	Use	
CODE	integer		Code representing the results of the action. For definitions, see "Valid Entries for CODE" on page 384.
ACTIONCODE	integer		Code representing the action being performed. For definitions, see "Valid Entries for ACTIONCODE" on page 384.
STARTEDPACKAGEACTIONID	integer		0 = started running the package, or when the new action is logged then the PACKAGEACTIONID of the new action replaces the 0.

Primary key (PACKAGEACTIONID)

Valid Entries for DESCRIPTION

Started package
Finished rollback
Error description
Or it is empty

Valid Entries for CODE

Success	= 0
Failure	= 1
Partial	= 2

Valid Entries for ACTIONCODE

AC_PACKAGE_INSTALL	= 0
AC_PACKAGE_ROLLBACK	= 1
AC_PACKAGE_INSTALL_STARTED	= 2
AC_PACKAGE_ROLLBACK_STARTED	= 3

SOFTWARECOMPONENTACTION

Contains one record for each server Server Software Package component.

Table 11-7 *SoftwareComponentAction Field Names*

Field Name	Type	Use	
SOFTWARECOMPONENTACTIONKEY	integer	not null	Assigned automatically: Default Auto increment.
PACKAGEACTIONID	integer	not null	Links to the PACKAGEACTION table.
NAME	char	not null	Name of the software component.
CREATIONDATE	timestamp		Time stamp of the action.
DESCRIPTION	varchar		The first record for the component the description is the description provided by the user when the component was created. As the components finish the description is one of those defined under "Valid Entries for DESCRIPTION" on page 385.
CODE	integer		Code representing the results of the action. For definitions, see "Valid Entries for CODE" on page 386.
ACTIONCODE	integer		Code representing the action being performed. For definitions, see "Valid Entries for ACTIONCODE" on page 386.

Primary key (SOFTWARECOMPONENTACTIONKEY)

Valid Entries for DESCRIPTION

Did not meet requirements
Error processing requirements
Pre-install load/unload
Error pre-install load/unload
Pre-install scripts
Error pre-install scripts
Copy file changes
Error processing copy file
Text file changes
Error processing text files
NetWare SET parameters
Error processing NetWare SET parameters
Registry process
Error processing Registry
NetWare products process

Error in NetWare products process
Post-install script process
Error in post-install script process
Post-install load/unload process
Error in post-install load/unload process

Valid Entries for CODE

Success	= 0
Failure	= 1
Partial	= 2

Valid Entries for ACTIONCODE

Started	= 200
Pre-Load	= 201
Pre-Scripts	= 202
Copy File Changes	= 203
Text File Changes	= 204
Set Parameters	= 205
Registry	= 206
Products.dat	= 207
Post Scripts	= 208
Post Load	= 209
Requirements	= 210

Foreign Keys

Foreign keys set up relationships between tables.

POLICYACTION

"add foreign key (POLICYID) references POLICIES (POLICYID)"

POLICYACTION

"add foreign key (SERVERID) references SERVERS (SERVERID)"

PACKAGEACTION

"add foreign key (PACKAGEGUID) references PACKAGES (PACKAGEGUID)"

PACKAGEACTION

"add foreign key (SERVERID) references SERVERS (SERVERID)"

SOFTWARECOMPONENTACTION

"add foreign key (PACKAGEACTIONID) references PACKAGEACTION
(PACKAGEACTIONID)"

11.4.3 Tiered Electronic Distribution Database Contents

Following are the database table definitions for Tiered Electronic Distribution:

TAB_NODE

Contains one record for each Distributor, Subscriber, and External Subscriber in the tree.

Table 11-8 *Tab_Node Field Names*

Field Name	Type	Use		
ID	numeric(8,0)	identity	not null	Unique number automatically assigned.
NAME	varchar(255)		not null	Tiered Electronic Distribution object DN.
TYPE	char		not null	"D"=Distributor "T"=Subscriber (Transceiver)
NETWORK_ADDRESS	varchar(255)			IP address of server.
SERVER_NAME	varchar(255)			Not currently used.

Primary key (ID)
Unique (NAME)

TAB_CHANNEL

Contains one record for each Channel object in the tree.

Table 11-9 *Tab_Channel Field Names*

Field Name	Type	Use		
ID	numeric(8,0)	identity	not null	Unique number automatically assigned.
NAME	varchar(255)		not null	DN of Channel object.

Primary key (ID)
Unique (NAME)

TAB_DISTRIBUTION

Contains one record for each Distribution object in eDirectory.

Table 11-10 *Tab_Distribution Field Names*

Field Name	Type	Use		
ID	numeric(8,0)	identity	not null	Unique number automatically assigned.
NAME	varchar(255)		not null	DN of Distribution object.
DISTRIBUTOR_ID	numeric(8,0)		not null	Links to the TAB_NODE table.

Primary key (ID)
Unique (NAME)

TAB_DIST_VERSION

Contains one record for each version of a Distribution and it is linked to the TAB_DISTRIBUTION table.

Table 11-11 *Tab_Dist_Version Field Names*

Field Name	Type	Use		
ID	numeric(10, 0)	identity	not null	Unique number automatically assigned.
DISTRIBUTION_ID	numeric(8,0)		not null	Links to the TAB_DISTRIBUTION table.
VERSION	bigint		not null	Time stamp of the version.
SIZE	integer		not null	Size of `distfile.ted` (the file containing the Distribution).

Field Name	Type	Use	
TIMESTAMP	datetime	not null	Time stamp when the entry was made to the database.
DIRECT_ROUTING	bit	not null	Not used at the current time.
LATEST_VERSION	bit	not null	Latest version of this Distribution. Used internally to keep track of the latest version.

Primary key (ID)
Unique (DISTRIBUTION_ID, VERSION)

TAB_DIST_ACTION

Contains multiple records for each Distribution version for Send, Received, and Extracted.

Table 11-12 *Tab_Dist_Action Field Names*

Field Name	Type	Use		
ID	numeric(12, 0)	identity	not null	Unique number automatically assigned.
DIST_VERSION_ID	numeric(10, 0)		not null	Links to the TAB_DIST_VERSION table.
NODE_ID	numeric(8,0)		not null	Links to the TAB_NODE table for the node performing the following tasks: Create Send Receive Extract Post process
TIMESTAMP	datetime		not null	Time stamp when the action was logged into the database.
STAGE	char		not null	"C"=Create "S"=Send "R"=Receive "E"=Extract "P"=Post process
STATUS	char		not null	"S"=Success "F"=Failure "P"=In process
STATUS_TIMESTAMP	datetime		not null	Time stamp when the record was updated.
REASON_TEXT	varchar(255)			Reason for success or failure. For definitions, see "Valid Entries for REASON_TEXT" on page 390.

Field Name	Type	Use	
CHANEL_DIST_ID	numeric(8,0)		Links to the TAB_CHANNEL_DISTRIBUTION table.

Primary key (ID)

Valid Entries for REASON_TEXT

The following are valid entries for the REASON_TEXT field name:

- ◆ "The Distribution was not received because this Subscriber does not meet the platform restrictions."

    Self-explanatory.

- ◆ "The Distribution was shut down before it was received."

    This one is received in one of two situations: 1) there is a new configuration on the Subscriber so it needs to be updated before it can receive the Distribution; or, 2) there is a signature exception, such as the Subscriber cannot trust the Distribution came from a Distributor it trusts.

- ◆ "The Distribution was terminated before it was received."

    The Distribution was cancelled for a controlled reason.

- ◆ "There was an error receiving the Distribution."

    Something unexpected failed. For example, a socket exception, transport exception, and so on.

TAB_CHANNEL_DISTRIBUTION

Contains one record for each Channel/Distribution.

Table 11-13   Tab_Channel_Distribution Field Names

Field Name	Type	Use		
ID	numeric(8,0)	identity	not null	Unique number automatically assigned
CHANNEL_ID	numeric(8,0)		not null	Links to the TAB_CHANNEL table.
DISTRIBUTION_ID	numeric(8,0)		not null	Links to the TAB_DISTRIBUTION table.
TIMESTAMP	datetime		not null	Time stamp for when the Distribution was built.

Primary key (ID)
Unique (CHANNEL_ID, DISTRIBUTION_ID)

Foreign Keys

Foreign keys set up relationships between tables.

TAB_DISTRIBUTION

" add foreign key FK_TAB_DIST_REF_591_TAB_NODE (DISTRIBUTOR_ID)" + " references TAB_NODE (ID) on update restrict on delete restrict;";

TAB_DIST_VERSION

" add foreign key FK_TAB_DIST_REF_37_TAB_NODE (DISTRIBUTOR_ID)" + " references TAB_DISTRIBUTION (ID) on update restrict on delete restrict;";

TAB_DIST_ACTION

" add foreign key FK_TAB_DIST_REF_380_TAB_NODE (DIST_VERSION_ID)" + " references TAB_DIST_VERSION (ID) on update restrict on delete restrict;";

TAB_DIST_ACTION

" add foreign key FK_TAB_DIST_REF_1525_TAB_NODE (NODE_ID)" + " references TAB_NODE (ID) on update restrict on delete restrict;";

TAB_CHANNEL_DISTRIBUTION

" add foreign key FK_TAB_DIST_REF_572_TAB_DIST (DISTRIBUTION_ID)" + " references TAB_DISTRIBUTION (ID) on update restrict on delete restrict;";

TAB_CHANNEL_DISTRIBUTION

" add foreign key FK_TAB_DIST_REF_572_TAB_CHAN (CHANNEL_ID)" + " references TAB_CHANNEL (ID) on update restrict on delete restrict;";

Distribution Types

A

The following sections describe how to configure the Distribution types:

A.1 Desktop Application

Use this option when the Distribution consists of an application created in ZENworks Desktop Management.

To create a Desktop Application Distribution:

1 Click the *Setup* button.

The Desktop Application Distribution Wizard is started.

After running the wizard the first time to create the Desktop Application Distribution, the *Setup* button is renamed to *Modify*.

For information on using the wizard, see Step 7 under Section 6.3, "Creating a Desktop Application Distribution," on page 296.

After you exit the wizard, the *Current Configuration* field displays the current configuration of the Desktop Application Distribution. This is same information that is displayed on the Summary page of the Desktop Application Distribution Wizard.

2 To modify the Desktop Application Distribution's configuration, click *Modify*.

This opens the Desktop Application Distribution Wizard again, where you can change the displayed configuration.

A.2 File

This option distributes files from the Distributor's file system. Files cannot be gathered from locations accessed by way of mapped drives or UNC paths. Files from other servers can be distributed using the FTP, HTTP, and RPM types of Distribution.

With this type you can select files and directories for the Distribution and select a destination path for extraction on the Subscriber.

Use the following fields and buttons to configure a File Distribution:

A.2.1 Files to Be Distributed

An expandable tree structure showing target paths to the files to be distributed.

Modify the Distribution's content as needed, then either click *Apply* to save the updated Distribution contents list, or click *OK* to save the changes and exit the Distribution's properties.

A.2.2 New Target

If you do not want to use the default variable %DEST_VOLUME%, replace it with a target location indicating where you want the files to be distributed; for example:

- A volume on a NetWare® server, such as `data:\`
- A drive on a Windows server, such as `D:\`
- A file system on a Linux or Solaris server, such as `/`

Otherwise, press *Enter* to accept the variable. You can also replace it with a different variable.

IMPORTANT: If you use a UNC path, all Distributions are sent to only that one location. Instead, use variables. For more information, see Chapter 9, "Variables," on page 349.

The target can also be a full or partial path, not just the root designation of volume, drive, or root. For example:

- A volume on a NetWare server, such as `data:\file_distribution\files`
- A drive on a Windows server, such as `d:\file_distribution\files`
- A shared folder on a Windows server, such as `\\myserver\files`
- A file system on a Linux or Solaris server, such as `/user/file_distribution/files`

You must press *Enter* to add the target location change to the Distribution contents list.

Any target entries that do not exist in the target location are automatically created by the Distribution.

A.2.3 Add Directory

This option builds the target path. If you did not provide the full path using the *New Target* option, replace *New Directory* with a directory name.

IMPORTANT: If a directory has a % character as part of its name, you must enter two consecutive % characters (%%) so that the second character is recognized as a literal % character and not a variable indicator.

You can use this option to add multiple directories to create the full path. You can also add sibling directories to create additional paths, as determined by the subdirectory you select before clicking *Add Directory*.

You must press *Enter* each time to add the target directory change to the path.

A.2.4 Add Files

Use this option to browse for directories or files on the Distributor's file system that you want copied to the target Subscriber's file system.

Each directory or file you select is displayed with the full path on the source file system. This path identifies where to obtain the directory or file for copying to the target file system. The only path that is created on the target file system is the one you create using the *New Target* and *Add Directory* buttons, including any directories that you select with the *Add Files* button.

If you select a directory, all files and subdirectories under it are also selected for copying. Unlike the Copy File component in the Server Software Package, you cannot prune files and subdirectories from a selected directory. Any directory you browse for and add is not expandable in this view.

You can create multiple paths (sibling directories) at any point in a particular path, but you can only have one root location.

IMPORTANT: The directories that you select for the Distribution and any target directories cannot be Read-Only. File-writing to or from such directories will fail.

When you are finished adding files and target locations, either click *Apply* to save the updated Distribution contents list, or click *OK* to save the changes and exit the Distribution's properties.

A.2.5 Delete

Deletes whatever you have selected:

- ◆ **Root Location:** Removes all directories and files below it from the tree.
- ◆ **Directory:** Removes the directory and any of its files and subdirectories from the tree.
- ◆ **File:** Removes the file from the tree, but not from its hard disk location on the server.

Click *Apply* to save your changes, or click *OK* to save the changes and exit the Distribution's properties.

A.2.6 Synchronize/Desynchronize

This option causes the directories on the target server to be synchronized with the directories contained in the Distribution, or you can desynchronize them. When you specify a directory to be synchronized, it includes all files and subdirectories under the synchronized directory.

Directory synchronization does not affect adding of directories and files through a File Distribution. Synchronization's purpose is to delete files and directories on the Subscriber servers. In other words, if the directories and files on the target Subscriber server are not contained in the File Distribution, they are deleted if synchronization is enabled.

WARNING: If the target server contains directories not contained in the Distribution, those directories, subdirectories, and files are deleted from the target server's file system when the Distribution is extracted.

This can be very destructive, especially if the target directory is a root directory. Enable directory synchronization only where you are certain you want to allow existing directories that are not contained in the Distribution to be deleted.

Also, if the Distributor whose file system you are using for this Distribution is also a Subscriber that is subscribed to the Distribution, the Distributor's file system is treated like the other target Subscribers' file systems and can be deleted.

For more information on synchronization, see Section 3.11.1, "Directory Sync Granularity for File Distributions," on page 178.

Directory synchronization provides granularity, where you can specify synchronization at any directory level in the Distribution to provide synchronization "from here down."

To synchronize directories, click either the target directory or the Distributor's source entry, then click *Synchronize*. A synchronization icon is added before the entry's graphic symbol.

The source location on the Distributor server determines what is kept in the synchronized directory on the Subscriber server, as depicted in the Distribution's *Files To Be Distributed* list. Thus, you are synchronizing the Subscriber's file system with the Distributor's, but only for the synchronized directory, with all its files and subdirectories.

The *Unsynchronize* button turns synchronization off (removing the icon), so that the specified directory is no longer synchronized.

If you enable synchronization, make sure that you refresh the Distributor before you make any changes to the directory being synchronized. This allows the Distributor to recognize that synchronization has been turned on, so that it rebuilds the Distribution with synchronization enabled.

A.2.7 Verify Distributions

Each time a Distribution changes, such as when files are modified or added, a new version is built and subsequently sent to the Subscribers. However, Subscribers might need to verify that the files contained in a Distribution have been extracted and installed to all Subscribers, even when there is no new version to send.

The verification option allows you to specify that if there is no new version of the Distribution to send, when the Send schedule starts the Distributor should send a request for the Subscriber to re-extract the current version to ensure that the files are installed.

A.2.8 Maintain Trustees

This option maintains each file's trustee attributes for the target NetWare file system so that they are the same as the source file system. The trustee information is obtained when the Distribution is built.

This is additive, meaning that it does not remove trustees on the target file system.

If synchronization is enabled for directories in a Distribution, the trustees of those directories are also synchronized.

A.2.9 Extract Error Handling

You have four options:

- **Fail on error:** Extraction of the Distribution stops. This results in a partial distribution. Correct the error and resend the Distribution.

- **Continue on error:** The extraction continues with an error written to the Subscriber's log file concerning the part of the extraction that failed.

  By default, the Subscriber continues past error conditions, so as many files as possible are successfully extracted. You can avoid locked open file errors by selecting *Kill connection on open files*.

- **Retry ___ times:** By default, open files on the Subscriber server are not overwritten when the Distribution is extracted because the open files are locked by the operating system. As result, updated files in the Distribution are not replaced on the Subscriber server if they are open when the Distribution is extracted.

  If you want the Subscriber to try multiple times to overwrite an open, locked file during extraction of the Distribution, specify the number of times the Subscriber should check the open file before failing to replace it because it is locked.

- **Kill connection on open files:** (NetWare only) Kills the connection that is holding the file open so that the file can be overwritten and the extraction can continue. (This applies only to files on the Subscriber server during extraction, not to files being accessed to build the Distribution.)

  Server and NLM™ connections cannot be killed.

Error messages are written to the Subscriber log file.

A.3 FTP

This option distributes files from one or more FTP sources. Each source can contain one or more directories and/or files.

The FTP Distribution type enables the files to pass through your firewall as they are gathered into the Distribution.

If a target file is locked during extraction, the Subscriber throws an exception stating that the file could not be copied. The Distributor receives this information from the Subscriber and log the failure in the reporting database.

If you want to distribute an RPM software package from a Linux or Solaris FTP site, use the RPM type of Distribution rather than the FTP type.

Use the following fields and buttons to configure an FTP Distribution:

- Section A.3.1, "Files To Be Distributed," on page 398
- Section A.3.2, "New FTP Source," on page 398
- Section A.3.3, "New Target," on page 398
- Section A.3.4, "Add Directory," on page 399
- Section A.3.5, "Add Files," on page 399
- Section A.3.6, "Delete," on page 399
- Section A.3.7, "Properties," on page 400
- Section A.3.8, "Binary Transfer," on page 400
- Section A.3.9, "Include Symbolic Link Files," on page 400

A.3.1 Files To Be Distributed

An expandable tree structure showing target paths to the FTP files to be distributed.

Modify the Distribution's content as needed, then either click *Apply* to save the updated Distribution contents list, or click *OK* to save the changes and exit the Distribution's properties.

A.3.2 New FTP Source

Specifies an FTP server from which to gather files for distribution:

- **FTP Server:** Specify the fully qualified host name of the FTP server.
- **Login Name:** Specify the login name that the Distributor Agent should use to access the FTP server. The default is Anonymous, which is often sufficient.
- **Password:** Provide the password for the login name. (You might need to scroll to the right to see the field.) For Anonymous, the password is typically your e-mail address.

A.3.3 New Target

If you do not want to use the default variable %DEST_VOLUME%, replace it with a target location indicating where you want the FTP files to be distributed; for example:

- A volume on a NetWare server, such as `data:\`
- A drive on a Windows server, such as `D:\`
- A file system on a Linux or Solaris server, such as `/`

Otherwise, press *Enter* to accept the variable. You can also replace it with a different variable.

The target can also be a full or partial path, not just the root designation of volume, drive, or root. For example:

- A volume on a NetWare server, such as `data:\ftp_distribution\files`
- A drive on a Windows server, such as `d:\ftp_distribution\files`
- A file system on a Linux or Solaris server, such as `/user/ftp_distribution/files`

You must press *Enter* to add the target location change to the Distribution contents list.

Any target entries that do not exist in the target location are automatically created by the Distribution.

A.3.4 Add Directory

This option builds the target path. If you did not provide the full path using the *New Target* option, replace *New Directory* with a directory name.

You can use this option to add multiple directories to create the full path. You can also add sibling directories to create additional paths, as determined by the subdirectory you select before clicking *Add Directory*.

You must press *Enter* each time to add the target directory change to the path.

A.3.5 Add Files

If you specified the correct FTP server information under *New FTP Source*, then you are provided access to the FTP server. Browse for the files. You can add multiple files.

If you added sibling directories using *Add Directory*, be sure to select those target paths and click *Add Files* to browse for the files to be distributed to those path locations.

When you are finished adding files and target locations, either click *Apply* to save the updated Distribution contents list, or click *OK* to save the changes and exit the Distribution's properties.

A.3.6 Delete

Deletes whatever you have selected:

- **Root Location:** Removes all directories and FTP files below it from the tree.
- **Directory:** Removes the directory and any of its FTP files and subdirectories from the tree.
- **File:** Removes the FTP file from the tree, but not from the FTP location.

Deleting any part of a target location, including a directory, means that the designated FTP files that were in the Distribution are no longer sent to the deleted location.

However, deleting FTP files from the Distribution means that the corresponding files are deleted from each Subscriber server when the Distribution is sent again and processed.

Click *Apply* to save your changes.

A.3.7 Properties

Displays the properties of the selected FTP source.

A.3.8 Binary Transfer

Enables file transfers in binary for when you are distributing executable files. Text files do not require a binary transfer.

A.3.9 Include Symbolic Link Files

If you want all symbolic link files in the added directory (including from all of its subdirectories) to be part of the FTP Distribution, select this box. This applies only when you add a directory.

You do not need to check this box for individually added symbolic link files.

This check box does not have any control over whether symbolic link files are displayed when browsing to add directories and files.

Click *Apply* to save your changes, or click *OK* to save the changes and exit the Distribution's properties.

A.4 HTTP

This option distributes files from one or more HTTP sources. Each source can contain one or more target entries.

The HTTP Distribution type enables the files to pass through your firewall as they are gathered into the Distribution.

If a target file is locked during extraction, the Subscriber throws an exception stating that the file could not be copied. The Distributor receives this information from the Subscriber and logs the failure in the reporting database.

Use the following fields and buttons to configure the Distribution:

- Section A.4.1, "Files To Be Distributed," on page 400
- Section A.4.2, "New Target," on page 401
- Section A.4.3, "Add Directory," on page 401
- Section A.4.4, "Add Files," on page 401
- Section A.4.5, "Delete," on page 402

A.4.1 Files To Be Distributed

An expandable tree structure showing target paths and the URLs to the HTTP files to be distributed.

Modify the Distribution's content as needed, then either click *Apply* to save the updated Distribution contents list, or click *OK* to save the changes and exit the Distribution's properties.

A.4.2 New Target

If you do not want to use the default variable %DEST_VOLUME%, replace it with a target location indicating where you want the HTTP files to be distributed; for example:

- A volume on a NetWare server, such as `data:\`
- A drive on a Windows server, such as `D:\`
- A file system on a Linux or Solaris server, such as `/`

Otherwise, press *Enter* to accept the variable. You can also replace it with a different variable.

The target can also be a full or partial path, not just the root designation of volume, drive, or root. For example:

- A volume on a NetWare server, such as `data:\http_distribution\files`
- A drive on a Windows server, such as `d:\http_distribution\files`
- A file system on a Linux or Solaris server, such as `/user/http_distribution/files`

You must press *Enter* to add the target location change to the Distribution contents list.

Any target entries that do not exist in the target location are automatically created by the Distribution.

A.4.3 Add Directory

This option builds the target path. If you did not provide the full path using the *New Target* option, replace *New Directory* with a directory name.

You can use this option to add multiple directories with this option to create the full path. You can also add sibling directories to create additional paths, as determined by the subdirectory you select before clicking *Add Directory*.

You must press *Enter* each time to add the target directory change to the path.

A.4.4 Add Files

Specify the URL where the files are available. You can specify a URL to a directory where multiple files exist, or include the filename in the URL to distribute a specific file.

To add files, you can select any directory in the path to assign the URL to it, including the root designation (such as `data:\`, `D:\`, or `/`).

If you added sibling directories using *Add Directory*, be sure to select those target paths and click *Add Files* to specify the URL for the files to be distributed to those path locations.

When you are finished adding files and target locations, either click *Apply* to save the updated Distribution contents list, or click *OK* to save the changes and exit the Distribution's properties.

A.4.5 Delete

Deletes whatever you have selected:

- **Root Location:** Removes all directories and files below it from the tree.
- **Directory:** Removes the directory and any of its URLs and subdirectories from the tree.
- **File:** Removes the URL from the tree, but not the files from the HTTP location.

Deleting any part of a target location, including a directory, means that the files designated by the URL that were in the Distribution are no longer sent to the deleted location.

However, deleting URLs from the Distribution means that the corresponding files are deleted from each Subscriber server when the Distribution is sent again and processed.

Click *Apply* to save your changes, or click *OK* to save the changes and exit the Distribution's properties.

A.5 MSI

This option distributes Microsoft Software Installer (MSI) packages to Windows servers for any Windows-based application, where the MSI engine is used to install the Windows software included in an MSI Distribution. You can create and configure MSI Distributions in both ConsoleOne and iManager.

Use the following to configure the Distribution:

- Section A.5.1, "Adding," on page 402
- Section A.5.2, "Removing," on page 402
- Section A.5.3, "Configuring," on page 402
- Section A.5.4, "Rearranging," on page 405

A.5.1 Adding

Add one or more MSI or MSP packages using one of the following:

Add from Distributor: The `.msi` and `.msp` files must reside on the file system of the Distributor server that owns this MSI Distribution.

Add from FTP site: The `.msi` and `.msp` files can be retrieved from an FTP site.

A.5.2 Removing

Select an MSI package from the list and click Remove to delete it from the list.

A.5.3 Configuring

To configure each MSI package, select the package in the *Selected Packages* column, click *Edit Parameter List* to open the Edit Parameters dialog box, then fill in the fields:

- "Distribution Includes Box" on page 403
- "Options Box" on page 403

Distribution Includes Box

Select one of the following options:

Package file only: Include only the MSI package in the Distribution.

Package files and folders: Include the MSI package, all files located in the same folder, and all subfolders and files. This assumes that all of the necessary supporting files for an MSI package are included in its folder and subfolders.

Options Box

Select from the following options:

Install: Causes the MSI package to be installed.

Uninstall: Causes the MSI package to be uninstalled.

Patch: This field is dimmed because it applies only to an MSP package.

Administrative install: Causes the MSI package to be installed without deleting the MSI package (as standard practice), so that it can available for a self-repair. This option is used in conjunction with an administrative image of the package. For more information, see the InstallShield Tip from AdminStudio (http://www.installshield.com/news/newsletter/0302-articles/setupexe.asp).

Repair: If you select this option, select from the following check boxes. They are the common MSI flags that can be passed to the MSI engine to specify the types of repairs to be made:

- **Missing file:** Instructs Windows Installer to reinstall a file only if it is missing.

- **Older file:** Instructs Windows Installer to reinstall a file if it is missing or if the installed file's version is older than the file in the MSI package.

- **Equal or older file:** Instructs Windows Installer to reinstall a file if it is missing or if the installed file's version is the same as or older than the file in the MSI package.

- **Force all:** Instructs Windows Installer to reinstall all files.

- **Use registry keys:** Instructs Windows Installer to rewrite all per-user entries from the MSI package to the Windows system registry. Per-user entries are those entries contained in the HKEY_CURRENT_USER and HKEY_USERS registry hives.

- **Computer registry keys:** Instructs Windows Installer to rewrite all per-machine entries from the MSI package to the Windows system registry. Per-machine entries are those entries contained in the HKEY_LOCAL_MACHINE and HKEY_CLASSES_ROOT registry hives.

- **Failed checksum:** Instructs Windows Installer to perform a checksum on all executable files and to reinstall a file if it is missing or if the checksum verifies that the file is corrupt. Only files that have msidbFileAttributesChecksum in the Attributes column of the MSI package's File Table are repaired.

- **Install and re-cache:** Instructs Windows Installer to install files from the re-cache (local) source rather than the source package.

- **Shortcuts:** Instructs Windows Installer to reinstall the MSI application's shortcuts, overwriting any existing shortcuts and icons.
- **Different file version:** Instructs Windows Installer to reinstall a file if it is missing or if the installed file's version is not exactly the same as the file in the MSI package.

Transforms Box

To include transforms in the Distribution, select an MSI package (not an MSP package) to be transformed, select the Edit Parameters button, then do the following. Repeat these steps as necessary for each transform in an MSI package, and for each MSI package.

Adding

Add one or more transforms using one of the following:

Add from Distributor: The `.mst` file must reside on the file system of the Distributor server that owns this MSI Distribution.

Add from FTP Site: The `.mst` file can be retrieved from an FTP site.

Transform files are used to modify the behavior of the MSI package that you selected in the Selected Packages column of the Type tab.

When two or more transforms are applied to the same MSI package property, it retains the value applied by the transform that was last applied.

For more information about creating and configuring transforms, see the documentation you received with the software application.

Removing

Select a transform from the list and click Remove to delete it from the list.

Rearranging

Use the Up and Down buttons to rearrange the order in which the transforms are applied.

When you rearrange the execution order, remember that an MSP package patches a specific MSI package, so it should be listed after the MSI package.

Custom Parameters Field

You can modify the listed command line parameters for the MSI package.

Some MSI Distributions can fail to extract on Windows 2000 servers. To solve this problem, see

Command Field

This field is display-only.

The parameters listed in this field are for the default options when you first view the Parameters dialog box. These parameters are automatically updated as you modify any options in the Distribution Includes, Options, or Transforms boxes, or add any parameters in the Custom Parameters field.

A.5.4 Rearranging

Use the Up and Down buttons to rearrange the order in which the MSI packages are applied.

A.6 Policy Package

Use this option when the Distribution consists of one or more policy packages containing enabled and configured policies. This is how Subscribers receive policies.

Only the policies contained in the Distributed Policy Package are distributed for enforcement on Subscriber servers. The Container Package and Service Location Package are not distributed; they continue to be associated for enforcement on Distributor servers.

To send a Policy Package Distribution to a Subscriber using an External Subscriber object, you must edit the `agentinfo.properties` file to prevent trusted tree errors. For more information, see "Preventing Trusted Tree Errors for Policy Package Distributions" on page 163.

For information on creating specific policies, see Chapter 4, "Server Policies," on page 197.

Use the following fields and buttons to configure the Distribution:

- Section A.6.1, "Up/Down," on page 405
- Section A.6.2, "Add," on page 405
- Section A.6.3, "Delete," on page 405
- Section A.6.4, "Properties," on page 406
- Section A.6.5, "The Following Policy Packages Will Be Distributed," on page 406

A.6.1 Up/Down

Rearranges the installation order for the policy packages.

A.6.2 Add

Adds a policy package to the Distribution.

A.6.3 Delete

Deletes the policy package from those listed.

A.6.4 Properties

Displays the properties of the selected policy package, which you can then edit.

A.6.5 The Following Policy Packages Will Be Distributed

Lists the policy packages to be distributed and the order of distribution.

A.7 RPM

You can distribute any Red Hat Package Manager (RPM) packages you have created to your Linux or Solaris servers through Tiered Electronic Distribution.

Use the following fields and buttons to configure the Distribution:

A.7.1 Up/Down

Arranges the installation order for the RPM packages.

A.7.2 Add From Distributor

Browse the Distributor's file system and select the RPM packages.

A.7.3 Add From FTP Site

Browse the FTP site and select the RPM packages.

A.7.4 Delete

Deletes the selected RPM package from the list.

A.7.5 Selected Packages

Lists the RPM packages you have added.

A.7.6 Installation Parameters

Lists the RPM installation parameters you have added.

Important points when entering parameters:

- You must press Enter for parameter entries in the text field, or the entries are not saved.
- You cannot remove a single parameter once it has been entered; you must re-enter the entire parameter string without the one you wanted removed.
- You cannot change the case of a parameter and have that change recognized. Instead, change the parameter to a different character, then change it back again to the original character with the desired case.

A.8 Software Package

Use this option when the Distribution consists of one or more software packages created in the Server Software Package namespace in ConsoleOne.

Use the following fields and buttons to configure the Distribution:

A.8.1 Up/Down

Rearranges the installation order for the software packages.

A.8.2 Add

Adds a software package to the Distribution.

A.8.3 Delete

Deletes the software package from the list.

A.8.4 Selected Software Packages

Lists the software packages to be distributed and the order of distribution.

Schedule Types

B

Table B-1 describes each of the Novell® ZENworks® schedule types, with links to the steps for configuring them.

Table B-1 *Schedule Types*

Schedule Type	Description
Daily	Runs the scheduled item daily. Daily includes specifying a run time window, running randomly within the window of time, and running repeatedly every *xxx* hours or minutes. Used by all Policy and Distribution Services components.
Event	Runs the scheduled policy according to the specified event, such as at system startup or shutdown, or a third-party application-defined event. Used only by policies.
Interval	Repeats running the scheduled item every *xxx* days, hours, minutes, and/or seconds. For Distributors only, the interval begins after the Distributor re-reads Novell eDirectory™. Any frequency from a few seconds to many days can be specified. Used by policies, Distributors, Distributions, Channels, and Subscribers.
Monthly	Runs the scheduled item on the selected day of the month. Monthly includes specifying a run time window and running randomly within the window of time. Used by all Policy and Distribution Services components.
Never	Prevents any of the four possible schedules from occurring. Only used by Tiered Electronic Distribution. This is generally used for manual control over a particular schedule. Typically, you do not need to leave an object configured with the Never schedule type for an extended period of time. If an object is no longer used, you can remove it using the Delete TED Object menu option in Novell ConsoleOne®.
Package Schedule	Runs the scheduled item according to the default schedule, which can be changed on the Policies tab. Used only by policies.
Relative	Runs the scheduled policy one time relative to a specified number of days, hours, minutes, and seconds from when the policy package is extracted. For example, if you set the time to one hour and refresh the Distributor, a new policy package is sent to the Subscriber, and it runs one hour after extraction. Used only by policies. Any time range, from a few seconds to many days, can be specified.
Run Immediately	Runs the scheduled item immediately upon refreshing the policy, beginning after the Distributor re-reads eDirectory. Includes repeating the action every *xxx* days, hours, minutes, and seconds. Any frequency from a few seconds to many days can be specified. Used only by policies, Distributions, Channels, and Subscribers.
Time	Runs the scheduled item once at the date and time specified. Used by all Policy and Distribution Services components.
Weekly	Runs the scheduled item on the selected day of the week. Weekly includes specifying a run time window, and running randomly within the window of time. Used only by policies.

Schedule Type	Description
Yearly	Runs the scheduled item on the selected day of the year. Yearly includes specifying a run time window, and running randomly within the window of time. Used by all Policy and Distribution Services components.

B.1 Daily

To schedule an item to run daily:

1 Click the down-arrow on *Schedule Type* > select *Daily*, then select one or more days of the week.

2 In *Start Time*, select the schedule's starting time for the day.

3 In *End Time*, select the latest time in the day for the schedule to run.

4 To have the schedule start randomly during the selected time period, select the *Randomly Dispatch* check box.

5 To have the schedule repeat the action, select the check box for the *Repeat the Action Every* field, then select how often the action should be repeated.

You can leave any of the options zeroed, but you must have a value in at least one of the time increments.

6 Click *Apply* to save the change.

B.2 Event

To schedule a policy to run when an event happens:

1 Click the down-arrow on *Schedule Type*, select *Event*, then select the event to activate the schedule:

Event	Description
System Startup	Runs the action when the system starts up.
System Shutdown	Runs the action before the system shuts down.
Custom Event ID	Third-party application-defined event.

2 Click *Apply* to save the change.

B.3 Interval

To schedule an item to run at an interval of time:

1 Click the down-arrow on *Schedule Type*, select *Interval*, then select the interval of time for repeating the action.

You can leave any of the options zeroed, but you must have a value in at least one of the time increments.

2 Click *Apply* to save the change.

B.4 Monthly

To schedule an item to run monthly:

1 Click the down-arrow on *Schedule Type*, select *Monthly*, select the option, then select the day of the month.

or

Select the option for the last day of the month (whether 28, 29, 30, or 31).

2 In *Start Time*, select the schedule's starting time for the day.

3 In *End Time*, select the latest time in the day for the schedule to run.

4 To have the schedule start randomly during the selected time period, select the *Randomly Dispatch* check box.

5 Click *Apply* to save the change.

B.5 Never

This is generally used for manual control over a particular schedule. Typically, you do not need to leave an object configured with the Never schedule type for an extended period of time. If an object is no longer used, you can remove it using the Delete TED Object menu option in ConsoleOne.

Never has the following effects on the distribution schedules:

- **Refresh (Distributor object):** Prevents the Distributor from reading eDirectory to discover new distribution work.

- **Build (Distribution object):** Prevents the Distributor from building that particular Distribution.

- **Send (Channel object):** Prevents all Distributions listed in the Channel from being sent.

- **Extract (Subscriber object):** Prevents the Subscriber from extracting any of the Distributions it has received but not yet extracted. However, the Subscriber server can still receive Distributions.

    When Distributions are changed to another schedule, all Distributions not yet extracted are extracted by the Subscriber according to the new schedule. When temporarily overridden using the *ZENworks Server Management* role in Novell iManager, all Distributions not yet extracted by the Subscriber are then extracted.

In each of these cases, you can manually override the Never action in iManager (see "Forcing Policy and Distribution Services Agent Actions" on page 80). However, the Never type continues to be set for the schedule after that override action occurs.

To schedule a Tiered Electronic Distribution item to never run automatically:

1 Click the *Type* tab, then select the down-arrow on *Schedule Type*.

2 Select *Never*.

3 Click *Apply* to save the change.

B.6 Package Schedule

Each policy package has a default schedule for all policies in that package.

You do not need to do anything to schedule a policy to run according to the current Default Package Schedule.

To change the Package Schedule:

1 In ConsoleOne, select the OU containing your server policies, right-click the Distributed Server Package (in the right pane), then click *Properties*.

2 Click *Edit*.

3 Change *Package Schedule* to one of the following:

Daily	Yearly	Event
Weekly	Relative	Interval
Monthly	Run Immediate	Time

4 Click *Apply* to save the change.

B.7 Relative

To schedule a policy to run relative to the time the policy package has been extracted:

1 Click the down-arrow on *Schedule Type*, select *Relative*, then select an amount of time.

You can leave any of the options zeroed, but you must have a value in at least one of the time increments.

2 Click *Apply* to save the change.

B.8 Run Immediately

To schedule an item to run immediately:

1 Click the down-arrow on *Schedule Type*, then select *Run Immediately*.

2 If you want to repeat the action, select the *Repeat* check box.

3 Select a length of time.

You can leave any of the options zeroed, but you must have a value in at least one of the time increments.

4 Click *Apply* to save the change.

B.9 Time

To schedule an item to run at a specific time:

1 Click the down-arrow on *Schedule Type*, select *Time*, then select the calendar icon.

2 In the *Select Date and Time* dialog box:

   2a Select the month.

   2b Select the year.

   2c Select the day of the month.

   2d Select the time of day, then click *OK*.

3 Click *Apply* to save the change.

B.10 Weekly

To schedule a policy to run weekly:

1 Click the down-arrow on *Schedule Type*, select *Weekly*, then select one day of the week.

2 In *Start Time*, select the schedule's starting time for the day.

3 In *End Time*, select the latest time in the day the schedule can run.

4 To have the schedule start randomly during the selected time period, select the *Randomly Dispatch* check box.

5 Click *Apply* to save the change.

B.11 Yearly

To schedule an item to run yearly:

1 Click the down-arrow on *Schedule Type*, select *Yearly*, then select the calendar icon.

2 In the Select Date dialog box:

   2a Select the month.

   2b Select the day of the month.

3 In *Start Time*, select the schedule's starting time for the day.

4 In *End Time*, select the latest time in the day the schedule can run.

5 To have the schedule start randomly during the selected time period, select the *Randomly Dispatch* check box.

6 Click *Apply* to save the change.

Server Console Commands

C

You can perform some of the Novell® ZENworks® Server Management functions using command line entries on a NetWare® server console. The server commands documented here are those that are applicable to Server Management's Server Policies and Tiered Electronic Distribution.

For ways to perform the server console commands in a Web browser using the ZENworks Server Management role in Novell iManager, see Chapter 2, "Novell iManager," on page 63.

A ZENworks Server Management console command that is entered on a server console is executed only on that server. For more information, review the following sections:

C.1 ZENworks Server Management Console Commands

Table C-1 lists the ZENworks Server Management server console commands with short descriptions of the commands. The table also indicates at which server console prompt you can give a command.

The column heading M is for the server's main console prompt and Z for the ZENworks Server Management prompt. Under a console prompt column, a Y indicates that you can issue the command at that prompt and a – indicates that you cannot issue the command at that prompt.

Table C-1 *Console Commands*

Command	M	Z	Description
HELP	Y	Y	Displays a list of available commands. Only the commands applicable to a component are displayed.
HELP *command*	Y	Y	Displays help for the specified command.
CLS	Y	Y	Clears the screen. Useful for quickly recognizing which information is new when you enter a command.

Command	M	Z	Description

DOWN *option*	Y	Y	This is similar to the command used on the server's main console prompt. However, if you use DOWN at the `ZENworks Server Management` prompt, server policy settings for downing the server are followed.
			
			For the `ZENworks Server Management` prompt, this command has several options:
			
			  ◆ **DOWN SERVER:** Downs the server only; does not bring it back up.
			  ◆ **DOWN STATUS:** Displays the current down status.
			  ◆ **DOWN RESTART:** Downs the server, then restarts it.
			  ◆ **DOWN RESET:** Downs the server, then resets it.
			  ◆ **DOWN CANCEL:** Allows you to cancel the down, up to when the server is actually taken down. This does not leave the server in an unusable state.
			  ◆ **DOWN !:** Causes the down process to execute immediately, ignoring the Down Server Process policy that might be in effect.

EVENTS *option*	–	Y	The command has three options:
			
			  ◆ **EVENTS LIST:** Lists all registered events, including third-party events.
			  ◆ **EVENTS STATUS:** Gives the status of each event.
			  ◆ **EVENTS FIRE *event_ID*:** Allows you to manually run an event.

EXIT	–	Y	Closes the current command prompt's Java software. For example, if given at the Subscriber prompt, the Subscriber's Java software is closed.
LISTPLUGINS	–	Y	Lists the current Server Management plug-ins.

PACKAGE *option*	–	Y	You can do the following for the software packages installed on the server:
			
			  ◆ **PACKAGE LIST:** Lists the currently installed software packages. This is useful for knowing which packages to roll back and the order that they should be rolled back, which is the reverse order in which they finished installing, not the order that they started installing.
			  ◆ **PACKAGE PROCESS *full_package_path*:** Use this to manually install a software package.
			  ◆ **PACKAGE ROLLBACK:** Automatically rolls back (uninstalls) the most recently installed software package. For example, you installed three software packages on a server (Package1, Package2, and Package3), and Package1 was installed first, Package2 second, and Package3 last. If you want to roll back Package2, you need to first roll back Package3. To do so, enter `package rollback` at the server console once for Package3, then again for Package2.
			
			The software package installation order is not guaranteed, because the order is determined by when a package has finished processing. Therefore, the installation order might be Package2, Package1, Package3 when using the Package Rollback command. This order is shown by the Package List command.

Command	M	Z	Description
POLICY or POLICY LIST	–	Y	Lists the effective server policies. Each policy listed has a corresponding policy number for reference when using the POLICY ENFORCE command.
POLICY ENFORCE *policy_number*	–	Y	Used to manually enforce a specific policy. You can find the *policy_number* using the POLICY LIST command. This is useful for enforcing a policy ahead of its schedule. However, you usually use POLICY REFRESH first to ensure you are enforcing the most recent changes.
POLICY ENFORCE ALL	–	Y	Used to manually enforce all effective policies, such as after doing a POLICY REFRESH.
POLICY EVENTBASED	–	Y	Lists the event-based policies.
POLICY PLUGINS	–	Y	Lists the current policy enforcers and the current event handlers.
POLICY REFRESH	–	Y	Refreshes only the server's policies and schedules, as required (unlike the REFRESH command, which refreshes policies and undoes any changes made to the prompts). After using this command, you should do a POLICY ENFORCE.
POLICY REFRESHONLY	–	Y	Refreshes the server's policies, but does not schedule effective policies.
POLICY RESCHEDULEONLY	–	Y	Reschedules all current policies according to their schedules. Does not refresh the effective policies.
POLICY SCHEDULES	–	Y	Lists all policy schedules that are in effect.
PROMPT	–	Y	Temporarily resets the current prompt. It reverts back to whatever is specified in the Novell eDirectory™ object for the console prompt when the Java process is exited or restarted, or when the REFRESH command is given.
REFRESH	–	Y	Manually forces a refresh of a policy, including pending changes to service locations for the current server and temporary changes to `ZENworks Server Management` prompts. Used alone, it refreshes only the ZENworks Server Management policy. Use POLICY REFRESH to refresh all policies.
SETCONSOLELEVEL *number*	–	–	Sets the console message level: **0:** No messages **1:** Errors **2:** Successes & level 1 messages **3:** Warnings & level 2 messages **4:** Information & level 3 messages **5:** Trace information & level 4 messages **6:** Developer trace information & level 5 messages

Command	M	Z	Description
SETFILELEVEL *number*	–	Y	Sets the file message level:
			0: No messages
			1: Errors
			2: Successes & level 1 messages
			3: Warnings & level 2 messages
			4: Information & level 3 messages
			5: Trace information & level 4 messages
			6: Developer trace information & level 5 messages
SHOWSCHEDULE	–	Y	Lists the current schedules.
SHOWVARS	–	Y	Lists the predefined variables and their values. These variables can be used in Server Software Packages.
STATUS	–	Y	Lists the current status of Policy and Distribution Services, including:
			Base Path
			Plug-ins Loaded
			Events Registered
			Scheduled Items
			Console Level
TIME	Y	Y	Returns the current date and time that the server is set to.
VERSION	Y	Y	Returns the Server Management version for the `ZENworks Server Management` prompt, and the NetWare version for the console's main prompt.

C.2 Java Console Commands

Table C-2 lists some useful Java Virtual Machine (JVM*) commands.

Table C-2 *Java Commands*

Command	Description
java -show	Lists all loaded Java processes.
java -kill*nnn*	Kills the specified Java process. (*nnn* represents the Java process number from the `java -show` listing.)
java -killzfsexit	Kills all Server Management Java processes.
java -killall	Stops all loaded Java processes; however, it leaves Java loaded.
java -version	Displays the JVM version.
java -exit or unload java	This attempts to unload all Java process, including the JVM. `Java -exit` is the preferred command. This command is required for unloading any native NLM™ files that are called from Java, such as `zenfile.nlm`.

Load/Unload Actions

<div style="text-align: right; font-size: large;">D</div>

This information is used in several setup steps for the Server Policies components (see Chapter 4, "Server Policies," on page 197) or Server Software Packages (see Chapter 5, "Server Software Packages," on page 243).

D.1 Load NLM/Process

For all supported platforms.

If you select an NLM™ to be loaded by the software package, and the NLM is already running on the target server, the package installation fails and is rolled back (if rollback is enabled).

You can make sure that an NLM is not already loaded when you are including it in the software package by adding an unload option for that NLM before adding the load option, but only if this NLM does not require user input from the keyboard to unload it.

Filename: This must be the exact name, including the full path to the executable, unless the path to the file is a system path variable. For NLM files, include the `.nlm` extension.

Parameters: Include any command line parameters for the NLM or process being run.

Wait for this process to terminate before continuing: You can select this option for an NLM or process that terminates itself. It must terminate within 10 minutes or the whole loading process fails. By default, this option is dimmed.

D.2 Load Java Class

For NetWare® only.

Filename: This must be the exact class name as listed in the JAR file's source code. For example, `cpkmidtier.CPKMidTierConfigure` is used by the JAR file listed in the example shown below in the *JVM Parameters* field.

The `.class` extension is not necessary.

Parameters: Include any command line parameters required by the Java application being run, such as the Windows variable %computername%.

JVM parameters: Include any parameters for the `java` command. For example, the following parameters are the same as used with the `java` command on a Windows command prompt:

```
-cp -DSystemRoot=%SystemRoot% -DSystemDrive=%SystemDrive% -
Djava.library.path=C:\onenet C:\onenet\CPKMidTierConfigure.jar
```

In this example, %systemdrive% is a Windows system variable and your JAR file is named `CPKMidTierConfigure.jar`. The class entered in the *Filename* field (such as `cpkmidtier.CPKMidTierConfigure`) is used by the `CPKMidTierConfigure.jar` file, and this JAR file needs a Djava library path to be `C:\onenet`. Copy your JAR file into the `C:\onenet\` directory for the CPK to find it.

Wait for this process to terminate before continuing: You can select this option for a Java application that terminates itself. There is no time limit. It waits as long as the application is running. By default, this option is dimmed.

D.3 Unload Process

For all supported platforms.

If the NLM requires intervention to unload, you must remember to unload it manually before trying to install the software package.

Filename: This must be the exact name (the full path is not required). Because many NLM programs require user input to unload, their unloading cannot be automated.

Wait for this process to unload before continuing: You can select this option for a process that unloads itself. By default, this option is dimmed.

D.4 Start Service

For Windows only.

Service name: This must be the exact name.

Wait for this service to finish running before continuing: You can select this option for a service that starts itself. By default, this option is dimmed.

D.5 Stop Service

For Windows only.

Service name: This must be the exact name.

Wait for this service to stop before continuing: You can select this option for a service that stops itself. By default, this option is not selected.

Requirements for Server Software Packages

<div align="right">E</div>

This information is used in several setup steps for software packages. For more information, see Chapter 5, "Server Software Packages," on page 243.

IMPORTANT: By selecting a requirement, you are prescribing that it must be met to allow the software package or package component to be installed.

Table E-1 lists the requirement types:

Table E-1 *Server Software Package Requirements*

Requirement	Description
Operating System	The operating system (OS) requirements for running the files in the software package, including both the OS the files need for running and whether the target server has that OS.
Memory (RAM)	The minimum RAM required for running the files in the software package. If the target server does not meet that minimum, the software package is not distributed to it.
Disk Space	The minimum free disk space required for installing the files on the target server. If the target server does not meet that minimum free space, the software package is not distributed to it.
SET Commands	Which NetWare® SET commands you want specifically configured on the target server for the software package.
Registry	The registry changes that can be required on the target server for the files in the software package. For information on configuring individual registry entries, see Appendix F, "Registry Entries for Server Software Package Components," on page 431.
File	Indicates whether a file on the target server should exist or have a certain date.
Products.dat	Changes to `products.dat` that the software package requires. Usually, the changes are to update the versions of the software on the server from the contents of the software package. The `products.dat` file is used to determine which software and which version exist on the server.

E.1 Operating System

You can require the server to have a certain operating system before installing the software package.

To configure the server operating system requirement:

1 With the operating system requirement selected, select the server's platform.

Available platforms are NetWare, Windows, Linux, and Solaris.

2 Select the version relationship:

Any
Less Than
Less Than or Equal To
Equal To
Greater Than
Greater Than or Equal To

3 If you select an option other than *Any* for the *Version* field, fill in the *Major*, *Minor*, and *Revision* fields according to the information in the following table.

For Windows servers, version information cannot be specified. Therefore, Windows is not included in the table.

The *Major* and *Minor* fields are for the upper version limit. For Netware and Windows, the *Revision* field is for the required service pack revision. For Linux, the *Revision* field is for the Linux distribution update version.

Operating System Version	Subscriber Version [1]	Major	Minor	Revision
NetWare 5.1 + SP5	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5, ZENworks 6.5 SP1, ZENworks 6.5 SP2, or ZfS 3.0.2	5	10	5
NetWare 5.1 + SP6	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5, ZENworks 6.5 SP1, ZENworks 6.5 SP2, or ZfS 3.0.2	5	10	6
NetWare 5.1 + SP7	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5, ZENworks 6.5 SP1, or ZENworks 6.5 SP2	5	10	7
NetWare 5.1 + SP8	ZENworks 7 or 7 w/SP1only	5	10	8
NetWare 6 + SP2	ZfS 3.0.2 only	6	0	2
NetWare 6 + SP3	ZfS 3.0.2 only (with JVM 1.4.1)	6	0	3
NetWare 6 + SP4	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5, ZENworks 6.5 SP1, ZENworks 6.5 SP2, or ZfS 3.0.2	6	0	4
NetWare 6 + SP5	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5, ZENworks 6.5 SP1, or ZENworks 6.5 SP2	6	0	5
NetWare 6.5	ZfS 3.0.2 only	6	5	0
NetWare 6.5 + SP1a	ZENworks 6.5 or ZfS 3.0.2	6	5	1

Operating System Version	Subscriber Version [1]	Major	Minor	Revision
NetWare 6.5 + SP1.1	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5, ZENworks 6.5 SP1, ZENworks 6.5 SP2, or ZfS 3.0.2	6	5	1
NetWare 6.5 + SP2	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5, ZENworks 6.5 SP1, or ZENworks 6.5 SP2	6	5	2
NetWare 6.5 + SP3	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5 SP1, or ZENworks 6.5 SP2	6	5	3
NetWare 6.5 + SP4	ZENworks 7 and 7 w/SP1 only	6	5	4
NetWare 6.5 + SP5	ZENworks 7 w/SP1 only	6	5	5
OES NetWare	ZENworks 7 and 7 w/SP1 only	6	5	4
OES NetWare + SP1	ZENworks 7 and 7 w/SP1 only	6	5	5
OES NetWare + SP2	ZENworks 7 SP1 only	6	5	6
OES Linux	ZENworks 7 and 7 w/SP1 only	2	6	variable [2]
OES Linux + SP1	ZENworks 7 and 7 w/SP1 only	2	6	variable [2]
OES Linux + SP2	ZENworks 7 SP1 only	2	6	variable [2]
Red Hat Linux 7.1, 7.2, 7.3, 8	ZfS 3.0.2 only	2	4	variable [2]
Red Hat Linux 9	ZENworks 6.5 only	2	4	variable [2]
Red Hat Advanced Server 2.1	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5, ZENworks 6.5 SP1, ZENworks 6.5 SP2, or ZfS 3.0.2	2	4	variable [2]
Red Hat Enterprise Server 2.1	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5, ZENworks 6.5 SP1, ZENworks 6.5 SP2, or ZfS 3.0.2	2	4	variable [2]
Red Hat Enterprise Linux AS 3	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5, ZENworks 6.5 SP1, or ZENworks 6.5 SP2	2	4	variable [2]
Red Hat Enterprise Linux AS 4	ZENworks 7 and 7 w/SP1 only	2	4	variable [2]
Red Hat Enterprise Linux ES 3	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5, ZENworks 6.5 SP1, or ZENworks 6.5 SP2	2	4	variable [2]
Red Hat Enterprise Linux ES 4	ZENworks 7 and 7 w/SP1 only	2	4	variable [2]
Solaris 8	ZfS 3.0.2 only	5	8	N/A
Solaris 9	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5, ZENworks 6.5 SP1, or ZENworks 6.5 SP2	5	9	N/A
SUSE® Linux 8.1 [3]	ZfS 3.0.2 only	2	4	variable [2]
SUSE Linux 8.2	ZfS 3.0.2 only	2	4	variable [2]

Operating System Version	Subscriber Version [1]	Major	Minor	Revision
SUSE Linux Enterprise Server (SLES) 8	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5, ZENworks 6.5 SP1, or ZENworks 6.5 SP2	2	4	*variable* [2]
SUSE Linux Standard Server (SLSS) 8	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5, ZENworks 6.5 SP1, or ZENworks 6.5 SP2	2	4	*variable* [2]
SLES 9	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5, ZENworks 6.5 SP1, or ZENworks 6.5 SP2	2	6	*variable* [2]
SLES 9 SP1, SP2	ZENworks 7 and 7 w/SP1 only	2	6	*variable* [2]
SLSS 9	ZENworks 7, ZENworks 7 w/SP1, ZENworks 6.5, ZENworks 6.5 SP1, or ZENworks 6.5 SP2	2	6	*variable* [2]
SLSS 9 SP1, SP2	ZENworks 7 and 7 w/SP1 only	2	6	*variable* [2]

[1] The Subscriber column indicates the Subscriber version that is required for processing the software package on a server platform. You do not need to specify the Subscriber version here; however, the software package cannot be successfully sent and extracted on a server with one of these network operating systems unless the correct product version for the Subscriber software is running on it.

It is possible to have both ZENworks 6.5 (or later) Server Management and ZENworks for Servers 3.0.2 running in your network, such as when you are upgrading incrementally. This table provides platform information for both ZENworks product versions. For information on upgrading incrementally, see "Upgrade Concepts and Issues" in the *Novell ZENworks 7 Server Management Installation Guide*.

[2] Where *variable* is listed for the Linux platforms in this table, the number that you enter in the *Revision* field is the kernel revision that was either first shipped with the installed Linux operating system, or a later version that you might have updated your Linux servers to. You can use the uname -a command to determine the exact revision number.

Also, the revision number you need to enter depends on what you specify in the *Version* field (*Equal To*, *Greater Than*, and so on). For example, you could enter 0 in the *Revision* field and select *Greater Than* in the *Version* field to include all kernel revisions.

[3] Because Linux versions are the same for Red Hat and SUSE, and you can only select Linux in the *Platform* field, to differentiate between those two Linux distributions, elsewhere in the software package you can require a certain file belonging to either Red Hat or SUSE to exist on the server. For example, the /etc/SuSE-release file could be required on the server, so that only servers with the SUSE Linux version would be accepted for receiving the software package.

E.2 Memory (RAM)

To configure the server memory requirement:

1 With the memory requirement selected, select the condition:

> Less Than
> Less Than or Equal To
> Greater Than
> Greater Than or Equal To

2 Specify the size in megabytes of RAM for the condition selected.

E.3 Disk Space

To configure the disk space requirement:

1 With the disk space requirement selected, select the root location.

The two options are *SYS Volume* and *Volume*. To conserve disk space usage on NetWare servers, do not select the `sys:` volume if you have other volumes with available disk space.

IMPORTANT: Do not use literal volume/drive names, such as the forward slash (/) character, when you are sending to multiple platforms. For example if you specify the / character for Linux in a software package, and the Software Package Distribution also gets sent to Windows servers, Windows will recognize the / as meaning its root location and the files intended only for the Linux servers would be installed on the Windows servers. To solve this problem, do one of the following:

a. Use variables for disk space locations, which allows you to send to multiple platforms.

or

b. Use the Operating System requirement in conjunction with the Disk Space requirement to confine the Distribution to only the server platforms where the literal disk space location exists.

Examples of literal locations you can provide:

NetWare:
```
sys:
data:
```

Windows:
```
C:\
\\myserver\data\shared_folder
```

Linux or Solaris:
```
/
/usr
/usr/data
/usr/data
/etc
/mnt/files
```

For Linux and Solaris servers, it is any path that identifies a disk partition.

2 If you selected *Volume*, provide the volume's name.

3 Select the condition:

Less Than
Less Than or Equal To
Greater Than
Greater Than or Equal To

4 Specify the free disk space needed in megabytes for the condition selected.

It is important that the free disk space you specify exists at the location you specified in Step 1.

E.4 SET Commands

When adding SET commands, the SET Commands Wizard is automatically run.

To configure the SET commands requirement:

1 With the SET commands requirement selected, provide the name of the SET command.

2 Provide the SET command's value.

E.5 Registry

You can require that certain entries must exist in the registry before installing the software package.

To configure the registry requirement:

1 With the registry requirement selected, select the *Entry Type*:

Key
Name
Data

2 For both entry types *Key* and *Name*, select if it exists or does not exist.

or

For the entry type *Data*, select if it equals or does not equal.

3 Enter the text for *Key*, *Name*, or *Data* (depending on which you selected in Step 1).

Make sure you add the two backslashes to the beginning of the *Key*. For example,
`\\HKEY_LOCAL_MACHINE\software\....`

IMPORTANT: The % symbol is not valid in NetWare registry names.

HKEY_LOCAL_MACHINE does not convert to "My Server" for this *Registry* field entry as it does for the *Registry Settings* field (see "Registry Settings" on page 271).

E.6 File

To configure the file requirement:

1 With the file requirement selected, provide the name.

   Include the file's full path.

2 Select the required file status:

   File Exists
   File Does Not Exist
   Date Is

E.7 Products.dat

WARNING: Modifying the `products.dat` file could prevent something from running or being installed on the NetWare server. Never modify any entries supplied by Novell®.

To configure the `products.dat` requirement:

1 With the `products.dat` requirement selected, provide the name of item in the `.dat` file.

IMPORTANT: Names are case sensitive.

   The item is the ID of the product in the `.dat` file.

2 Provide the version text that corresponds with the item selected in Step 1.

3 Select whether the version *Contains*, *Begins With*, or *Matches* the version specified in Step 2.

4 Provide the description text that corresponds with the item selected in Step 1.

5 Select whether the description *Contains*, *Begins With*, or *Matches* the description provided in Step 4.

Registry Entries for Server
Software Package Components

<div style="text-align: right">F</div>

The following information is used in several setup steps for software packages. For more information, see "Registry Settings" on page 271.

The NetWare® or Windows registry entries you can change are keys, value names, and value data. You can select keys and value data types for making changes, and you can provide the corresponding value names when you select one of the types.

In all cases, you must enter the exact key name or value name that is expected in the registry, as well as the correct data values.

The registry settings under HKEY_LOCAL_MACHINE are the only ones you can configure using a software package.

You can change the following registry entries when you install a software package:

- Section F.1, "Key," on page 431
- Section F.2, "Binary," on page 432
- Section F.3, "Expand String," on page 432
- Section F.4, "(Default)," on page 432
- Section F.5, "DWord," on page 433
- Section F.6, "Multi-Value String," on page 433
- Section F.7, "String," on page 433

F.1 Key

Keys create the paths to the various registry entries. For example, HKEY_LOCAL_MACHINE is a registry key at the root level, and HARDWARE is a key directly under it. The keys are displayed with folder icons in tree fashion. You can click the plus or minus signs to expand or compress the tree structure.

In the box where the HKEY_LOCAL_MACHINE key is displayed, you need to use the Key registry entry to create the path to where the registry changes are placed.

To configure a Key entry:

1 In the box displaying your key tree, select the location where you want the key inserted.

2 Select *Key* from the drop-down box, then click *Add*.

New Key #1 is displayed.

3 Change the default key name to the key name that you need.

When entering information into this field, you must press Enter for the change to be saved.

4 Select a condition for making the registry change:

Create
Delete

5 To apply the setting to all subordinate keys, click *Apply To All*.

F.2 Binary

A value data type that is a list of hexadecimal numbers, such as:

```
d0 04 72 6e
```

You must first use the Key registry setting option to create the path to the key that holds the Binary information.

To configure a Binary entry:

1 In the box displaying your key tree, select the location where you want the binary data inserted.

2 Select *Binary* from the drop-down box, then click *Add*.

New Value #1 is displayed.

3 Change the default binary name to the name that you need.

4 Select a condition for making the registry change:

Create
Delete

5 Provide the binary data.

The *Data* box is a hexadecimal editor. There are three unlabeled columns:

First: Binary counter of the number of hexadecimal characters, beginning with 0000.

Second: Hexadecimal data, eight entries per row.

Third: Plain text ASCII characters corresponding to the hexadecimal data.

You can enter data in either the second or third column. As you enter data in one the second (hexadecimal) column, the corresponding characters are displayed in the third (text) column, and vice versa.

F.3 Expand String

NetWare only. Currently not supported.

F.4 (Default)

This is usually the first data entry for a key.

You must first use the Key registry setting option to create the path to the key that holds the (Default) entry.

To configure a (Default) entry:

1 In the box displaying your key tree, select the location where you want the *(Default)* entry made.

2 Select *(Default)* from the drop-down box, then click *Add*.

(Default) is displayed.

3 With the *(Default)* entry selected, select a condition for making the registry change:

Create
Delete

4 Enter a string in *Data*.

F.5 DWord

DWords are based on hexadecimal code that is represented in Double WORD format. For example:
`0x00100022`

You must first use the Key registry setting option to create the path to the key that holds the DWord information.

To configure a DWord entry:

1 In the box displaying your key tree, select the location where you want the DWord entry made.

2 Select *DWord* from the drop-down box, then click *Add*.

New Value #1 is displayed.

3 Change the default DWord name to the name that you need.

4 Select a condition for making the registry change:

Create
Delete

5 Enter the DWord string in *Data*.

F.6 Multi-Value String

NetWare only. Currently not supported.

F.7 String

String values are easy-to-read sequences of words or numbers within quote marks.

You must first use the Key registry setting option to create the path to the key that holds the String information.

To configure a String entry:

1 In the box displaying your key tree, select the location where you want the string data inserted.

2 Select *String* from the drop-down box, then click *Add*.

New Value #1 is displayed.

3 Change the default string name to the name that you need.

4 Select a condition for making the registry change:

Create
Delete

5 Enter the string in *Data*.

Client Access in Linux

<div style="text-align: right; font-size: 2em;">G</div>

Tiered Electronic Distribution sends application objects and content to Linux servers using file systems such as Reiser, Ext2, Ext3, and NSS (OES Linux only). However, NCP share names used in a Desktop Application object are not directly supported on Linux servers because Tiered Electronic Distribution cannot read NCP share names.

Do one of the following to ensure client access to files on Linux servers:

- Section G.1, "Using Samba," on page 435

  Use this option if you are running Samba.
- Section G.2, "Using NCP Shares," on page 435

  Use this option if you are not running Samba.

G.1 Using Samba

If you are accessing a Linux Subscriber server (including OES Linux) using a Windows client, access can be provided via a Samba share. Desktop Application objects hosted on that server will use the Samba share name in the UNC patch of the Application object.

To use Samba for share recognition:

1 On your Linux server, install Samba.

  Refer to your Linux documenation for instructions on installing and configuring Samba.

2 Configure Samba by including a Samba share name (such as SYS) with the following path:

  `/usr/novell/`*share*

  where *share* is the Samba share name, such as `sys`.

3 Make sure Samba is running on the server when you send Desktop Application Distributions.

  When a Desktop Application Distribution is configured with a golden application containing a path such as `\\192.68.1.203\sys\firefox`, the `sys` volume is interpreted as a share on the Linux server, allowing the Firefox application to be written to the `/usr/novell/sys/firefox` directory.

4 Repeat Step 1 through Step 3 for each Linux Subscriber and Distributor server where access to a Samba share name is required.

  Include servers that have a location where:

  - The application files are to be written to (Linux Subscribers)
  - The application files are to be distributed from (Linux Distributors)

G.2 Using NCP Shares

If you do not want Samba running on your OES Linux Subscriber server, you can ensure that a Desktop Application Distribution will succeed by including the NCP share names in the Samba

configuration file. Samba does not need to be running in order for Tiered Electronic Distribution to use the `smb.conf` file.

To edit the Samba configuration file:

1 On your OES Linux server, open the following configuration file in a text editor:

```
/etc/samba/smb.conf
```

2 Copy an existing share section to the end of the file and paste it as many times as you have NCP shares to add, then edit the copied sections to be:

```
[share1]
comment = for access to the NCP share
path = /usr/novell/share1
write list =@ntadmin root
force group = netadmin
create mask = 0644
directory mask = 0755
[share2]
comment = for access to the NCP share
path = /usr/novell/share2
write list =@ntadmin root
force group = netadmin
create mask = 0644
directory mask = 0755
```

where *share1* and *share1* are the NCP share names, such as sys and apps. Note that the share names are used both within the brackets and in the path. The share names in the `smb.conf` file and in NCP must be the same.

NOTE: The last four lines for each new share setion do not need to be edited from whatever is contained in the original copied section. The values shown above in these lines are not used by Tiered Electronic Distribution.

To display which NCP shares are available, enter `ncpcon` on the OES Linux server console and type `help` or `--h` for a list of the NCP command line options.

3 Save the file and exit the text editor.

When a Desktop Application Distribution is configured with a golden application containing a path such as `\\192.68.1.203\sys\firefox`, the `sys` volume is interpreted as a share on the OES Linux server, allowing the Firefox application to be written to the `/usr/novell/sys/firefox` directory.

4 Identify the servers that have a location where:

   ◆ The application files are to be written to (Subscribers)

   ◆ The application files are to be distributed from (Distributors)

then do one of the following:

   ◆ If you have only a few servers to configure, repeat through for each Linux Subscriber and Distributor server where access to an NCP share name is required.

   ◆ If you have very many servers that need this configuration change, use a Text File Changes policy (in the Distributed Server Package) to roll out an updated `smb.conf` file to them. For more information, see .

Configuration Planning Worksheet

H

Use the following worksheet to log configuration information as you plan how to set up your distribution system. You might need to attach lists for some items.

This worksheet is designed to print best from the PDF version of the documentation.

IMPORTANT: Do not use this planning worksheet by itself to configure Policy and Distribution Services, even if you feel experienced enough to do so. There are some required configuration steps that are not covered in this worksheet, because planning is not needed for those steps. Use the sections under Section 1.2, "Configuring Your Distribution System," on page 50 as your guide for performing the actual configuration of Policy and Distribution Services.

Configuration Information	Instructions
Installing Additional Distributors, Databases, and Subscribers	If you do not have additional Distributors, databases, or Subscribers to install, skip to worksheet item 12.
1) Tree for the Distributor and ZENworks Database objects:	Provide the name of the eDirectory tree for installing the Server Management objects. For more information, see Section 1.1.3, "Understanding Your Network Topology," on page 37.
2) Distributor server names:	Provide the server names for each server that you want to be a Distributor. Distributor servers build and own the Distributions. For more information, see "Distributor Properties" on page 38.
3) Subscriber server names:	Provide the server names for each server that you want to be a Subscriber. Subscriber servers receive and extract the Distributions. For more information, see Section 1.1.5, "Other Subscribers To Be Installed?," on page 41.

Configuration Information	Instructions
4) Database server names:	Provide the server names for each server where you want to install the Server Management database, which can be installed on NetWare and Windows servers. You can have multiple databases for Policy and Distribution Services, but only one per server. Also specify the purpose for each database, or a Distributor identifier for each database if they will each be used the same way. For more information, see "Whether a Distributor Server Will Host a Server Management Database" on page 39.
5) Installation paths for Distributors' software:	Provide the path where you want the Distributor software installed. The default is `\zenworks` for both NetWare and Windows servers. For more information, see "Software Installation Paths" on page 39.
6) Installation paths for Subscriber software:	Provide the path where you want the Subscriber software installed. The default is `\zenworks` for both NetWare and Windows servers. For more information, see "Software Installation Paths" on page 39.
7) Distributors' properties, where different than the installation defaults:	Edit the following information for your Distributor servers: ◆ Distributor object's name (the default is Distributor_*server_name*) ◆ Distributor's context ◆ Distributor server's working directory For more information, see "Distributor Properties" on page 38.

Configuration Information	Instructions
8) Subscribers' properties, where different than the installation defaults:	Edit the following information for your Subscriber servers: ◆ Subscriber object's name (the default is Subscriber_*server_name*) ◆ Subscriber context ◆ Subscriber server's working directory For more information, see Section 1.1.5, "Other Subscribers To Be Installed?," on page 41.
9) Installation paths for Server Management database software:	Provide the path where you want the zfslog.db file located. The default is \zenworks\database. For NetWare servers, we recommend not using the sys: volume because the database file can become very large. We also recommend that you install the database software on a server where the Subscriber software is also installed so that you can use the Database Purge option. For more information, see "Software Installation Paths" on page 39.
10) Database object name:	Either accept the default names, or provide ones that will help you to identify the databases' purposes. For more information, see "Whether a Distributor Server Will Host a Server Management Database" on page 39.

Configuration Information	Instructions
11) Database object Container:	We recommend you use the same container where your other Tiered Electronic Distribution objects reside. For more information, see "Whether a Distributor Server Will Host a Server Management Database" on page 39.
Configuring the Distributors for a Mixed eDirectory Environment	If you do not have a mixed eDirectory environment, skip to worksheet item 13.
12) IP address of server in eDirectory 8.*x*:	Provide the IP address of a server in the tree using eDirectory 8.*x*. This can be the Distributor server's IP address, if that server is running eDirectory 8.*x*. For more information, see "Whether Distributors Might Exist in a Mixed eDirectory Environment" on page 40.
Installing Inter-Server Communications	If you do not need to set up inter-server communications, skip to worksheet item 14.
13) Subscriber servers outside your secured network:	Inter-server communications security might be needed if your Distributor and Subscriber servers communicate with servers outside your secured network. For more information, see "Determining Whether You Need Inter-Server Communications Security" on page 44.

Configuration Information	Instructions
Installing NICI on Windows, Linux, or Solaris Servers	If you do not need to install NICI to these servers, skip to worksheet item 15.
14) Windows, Linux, or Solaris servers (Distributor or Subscriber) that will be involved with Distribution encryption:	List the Windows, Linux, or Solaris servers that will either build (Distributors) or extract (Subscribers) encrypted Distributions. For more information, see "Determining Whether You Need Encryption Security for Windows Servers" on page 45.
Configuring the Distributor Routing Hierarchies	
15) Distributors' routing hierarchies of tiered Subscribers:	Create a chart of tiered Subscribers for each Distributor showing how you want your Distributions to be distributed on your network. Distributors can use Subscribers in other Distributor's routing hierarchies. However, a Subscriber should only be used once in a given Distributor's hierarchy so that an end-node Subscriber only has one distribution path for receiving a particular Distribution. For more information, see Section 1.1.6, "Determining the Distribution Flow," on page 42.

Configuration Information	Instructions
Configuring Parent Subscribers	
16) Subscriber/parent Subscriber assignments (end-node Subscribers associated with a parent Subscriber):	Create Subscriber lists where each parent Subscriber delivers Distributions. You should assign each end-node Subscriber to a parent Subscriber, except where you want the end-node Subscriber to receive its Distribution directly from the Distributor. For more information, see "Selecting Subscribers for the Distribution Routes" on page 43.
Creating and Configuring Subscriber Groups	If you are not using Subscriber Groups, skip to worksheet item 19.
17) Subscriber Group object name:	Provide a unique name for the Subscriber Group. For more information, see "Subscriber Groups" on page 47.
18) Subscribers to be in this group:	Provide a list of Subscribers that need the same Distributions from the Channel where the group is subscribed.

Configuration Information	Instructions

Creating the Policy Package Distributions

Configuration Information	Instructions
19) Distributions, their types, and their Distributors:	Create a list of your Distributions. For each Distribution, include the Distribution type, object name, and servers that need the Distribution. The Distribution types are: • "Desktop Application" on page 32 • "File" on page 33 • "FTP" on page 33 • "HTTP" on page 34 • "MSI" on page 34 • "Policy Package" on page 35 • "RPM" on page 36 • "Software Package" on page 36 For more information, see Section 1.1.2, "Selecting Your Distributions," on page 32.

Configuration Information	Instructions
Creating and Configuring the Channels	
20) eDirectory container for Tiered Electronic Distribution objects:	Container for creating and managing Tiered Electronic Distribution objects.
	You might have created a container during installation of Policy and Distribution Services. If not, you should create a container specifically for managing Tiered Electronic Distribution objects.
	For more information, see "Whether a Distributor Server Will Host a Server Management Database" on page 39.
21) Channel names:	Provide the names of the Channel objects that you need for your Distributions. We recommend a unique Channel for each unique Distribution or Distribution grouping.
	For more information, see Section 1.1.8, "Determining the Channels for the Distributions," on page 46.
22) Distributions for the Channels:	Create a list of which Distributions belong to which Channels.
	For more information, see Section 1.1.8, "Determining the Channels for the Distributions," on page 46.

Configuration Information	Instructions
Subscribing to the Channels	
23) Subscribers' Extract schedules:	Set extract schedules per Subscriber server according to when it would be best for each Subscriber to be extracting its Distributions. For more information, see Section 1.1.9, "Determining Subscribers' Subscriptions," on page 47.
24) Channel associations with Subscribers and Subscriber Groups:	Create lists where Subscribers and Subscriber Groups are associated with the Channels that have the Distributions you want them to receive.

Documentation Updates

This section contains information on documentation content changes that were made in this *Administration Guide* after the initial release of Novell® ZENworks® 7 Server Management. The information can help you to keep current on updates to the documentation.

All changes that are noted in this section are also made in the documentation. The documentation is provided on the Web in two formats: HTML and PDF. The HTML and PDF documentation are both kept up-to-date with the documentation changes listed in this section.

The documentation update information is grouped according to the date the changes are published. Within a dated section, the changes are alphabetically listed by the names of the main table of contents sections in the guide.

If you need to know whether a copy of the PDF documentation you are using is the most recent, the PDF document contains its publish date on the front title page.

The documentation was updated on the following dates:

I.1 March 29, 2007

Updates were made to the following sections:

Location	Change
"Determining the Distributor's Refresh Schedule" on page 49	Updated this section to clarify use of the Refresh schedule. For example, the default of Never is recommended because an infinite loop situation could be caused by a Refresh schedule that occurs more frequently than the time it takes to build or send a Distribution.
	Also updated the following sections with information related to this change:
	"Scheduling the Distribution and Refreshing the Distributor" on page 60 - Section 3.2.2, "The Basic Distribution Process," on page 88 "Scheduling" on page 97 - Section 3.3.5, "Manually Refreshing the Distributor," on page 110
"File" on page 119	Added the following paragraph to the IMPORTANT note in the section:
	Also, if a NetWare server is the target for a File Distribution, you might encounter an error due to code page differences where extended characters are used (such as ê, ë, ì, or í). The information in "Extended Characters in Directory Paths" on page 294 in the Desktop Application Distribution section is also applicable to File Distributions.

Location	Change
"MSI" on page 121	Added the following note to the section: **IMPORTANT:** Because an MSI Distribution recursively gathers all of the files from the MSI file's location, if you have multiple `.msi` files in a given location, all other files and subdirectories contained therein are gathered once for each `.msi` file. The distribution gathering process cannot determine which other files or subdirectories belong to each `.msi` file, so you can end up with a much larger MSI Distribution file than is necessary. Therefore, instead of storing your `.msi` files in one directory, place each into its own subdirectory with its own supporting files and subdirectories.
Section 3.10.5, "Sending Distributions: Firewall and Cluster Issues," on page 178	Added this section concerning distribution issues for firewalls and clusters.

I.2 August 16, 2006

Updates were made to the following sections:

- Load/Unload Actions
- Security in Policy and Distribution Services
- Server Policies
- Tiered Electronic Distribution

I.2.1 Load/Unload Actions

The following changes were made in this section:

Location	Change
Section D.2, "Load Java Class," on page 421	Updated this section with example information to make it more clear how to use the Load Java Class feature.

I.2.2 Security in Policy and Distribution Services

The following changes were made in this section:

Location	Change
Section 7.3.5, "TCP/IP Addresses and DNS Names," on page 323	Removed the note concerning not using underscores in server names, as this restriction no longer applies.

I.2.3 Server Policies

The following changes were made in this section:Section 7.3.5, "TCP/IP Addresses and DNS Names," on page 323

Location	Change
"Verifying Community String Changes" on page 230	Added this new section which provides instructions on using TCPCON to view community string changes.

I.2.4 Tiered Electronic Distribution

The following changes were made in this section:

Location	Change
"Message Notification Levels" on page 186	Updated this section, further explaining the messaging level types.
"Managing Message Notification Level Log Files" on page 187	Created this section from information previously contained in "Message Notification Levels" on page 186.

I.3 July 14, 2006 (Support Pack 1)

Updates were made to the following sections:

- Desktop Application Distribution
- Distribution Types
- Novell iManager
- Requirements for Server Software Packages
- Schedule Types
- Server Policies
- Tiered Electronic Distribution

I.3.1 Client Access in Linux

The following changes were made in this section:

Location	Change
Appendix G, "Client Access in Linux," on page 435	Added this new section which provides instructions on setting up client access on Linux servers (including OES Linux) to NCP share names used in Desktop Application objects.

I.3.2 Desktop Application Distribution

The following changes were made in this section:

Location	Change
Section 6.3.2, "Creating the Distribution," on page 297	Removed documentation for the *Delete previous revision before receiving next* field, which is no longer used, and added documentation for the e-mail fields that replaced it.

I.3.3 Distribution Types

The following changes were made in this section:

Location	Change
Section A.2, "File," on page 393	Updated the information in this section.
Section A.2.8, "Maintain Trustees," on page 397	Added the following sentence: If synchronization is enabled for directories in a Distribution, the trustees of those directories are also synchronized.
Section A.3, "FTP," on page 397	Updated the information in this section.
Section A.4, "HTTP," on page 400	Updated the information in this section.

I.3.4 Novell iManager

The following changes were made in this section:

Location	Change
Section 2.4.1, "Setting Up Passwords for Remote Web Console," on page 73	Added this section, which documents the addition of passwords for Remote Web Console in iManager.
Section 2.4.2, "Managing the Distributor Agent," on page 77 and Section 2.4.3, "Managing the Policy/Package Agent," on page 80	Updated these sections with password entry steps.

I.3.5 Requirements for Server Software Packages

The following changes were made in this section:

Location	Change
Section E.1, "Operating System," on page 424	Updated the table for the *Major*, *Minor*, and *Revision* fields with newer information and the OES platform.

I.3.6 Schedule Types

The following changes were made in this section:

Location	Change
Appendix B, "Schedule Types," on page 409 and Section B.5, "Never," on page 411	Updated the information for the *Never* option.

I.3.7 Server Policies

The following changes were made in this section:

Location	Change
"Server Down Process" on page 226	Added the following note:
	IMPORTANT: For the Windows, Linux, and Solaris platforms, if you down the server from its console, this policy is not recognized. Instead, you must down the server using the *Actions* option in *Remote Web Console* in iManager so that this policy can be applied.

I.3.8 Tiered Electronic Distribution

The following changes were made in this section:

Location	Change
"Maximum Revisions" on page 116 and Section 3.4.4, "Creating a Distribution," on page 124	Removed documentation for the *Delete previous revision before receiving next* field, which is no longer used, and added documentation for the e-mail fields that replaced it.

I.4 December 9, 2005

Page design is reformatted to comply with revised Novell documentation standards.

I.5 October 7, 2005

Updates were made to the following sections:

 • Desktop Application Distribution

I.5.1 Desktop Application Distribution

The following changes were made in this section:

Location	Change
Section 6.3.2, "Creating the Distribution," on page 297	In Step 7b, added information for the new option: Overwrite Existing Target Folder Object Attributes.