

ZENworks 6.5 Server Management

Upgrading and Patching Methodologies

www.novell.com

AUGUST 2005



Novell.

Executive Summary.....	3
Mechanics of Tiered Electronic Distribution.....	3
Introduction to the Software Package Engine.	4
Patches and Upgrades Available Today.....	6
Novell NetWare Service Pack.....	6
Open Enterprise Server Upgrade: NetWare to NetWare.....	9
Novell eDirectory Patching.....	14
Windows.....	16
GroupWise Upgrades and Patching.....	16
GroupWise Upgrading.....	18
GroupWise 6.0 SP4 to 6.5 SP4 Upgrade for NetWare.....	20
GroupWise 6.0 SP4 to 6.5 SP4 Upgrade for Windows.....	20
Novell Java Virtual Machine for NetWare Patching.....	21
Upgrading and Patching Your ZENworks Server Management Environment.....	22
Upgrading Tiered Electronic Distribution Using Itself.....	24
Appendix A: Localized Silent Install Scripting Example.....	26
Appendix B: Terms Used in This Document.....	32
Appendix C: Credits and Legal Attribution.....	33
Author.....	33
Contributors and Reviewers.....	33

EXECUTIVE SUMMARY

In today's business environment, manually controlling servers for periodic updates and tasks is impractical. It is a well known fact that automating repetitive tasks can significantly lower a system's total cost of ownership.

Typically, when administrators patch a given server, they need to remotely control each server, copy the patch, launch the patch installation and then monitor its completion. The time involved can be anywhere between half an hour and two hours—and sometimes even more. Multiply that by hundreds of servers and many patches—the costs to your business can skyrocket.

Not all patches can be slowly rolled out over time, either. Security patches need immediate attention, as do those that resolve the downtime that causes revenue loss. Additionally, scheduling each patch based on priority is crucial in reducing administrative costs. For example, giving your virus-update patch a higher priority over a service pack is imperative.

From a physical-networking perspective, things become even more complex. Sending the same patch five times to five servers instead of once and then forwarding it on can cause slow response times for your users. On the other hand, when you don't fully utilize the same WAN bandwidth, it is wasted during off-peak hours. The same arguments can be applied to upgrades, too. You can save much time and effort by automating application upgrades or even upgrades to the operating systems themselves.

Often, you may need to tailor each patch or upgrade for every server its going to run on. You may need to establish set or registry parameters prior to running the upgrade or change a few text files to localize the install script or upgrade application in order for it to run. Making such changes manually for every server in your organization is cost-prohibitive. However, performing these tasks with ZENworks Server Management software packages is an easy and efficient process.

This white paper addresses all of these concerns by showing how the award-winning Novell ZENworks software can tier, schedule and automate the distribution of Novell patches. With ZENworks Server Management 6.5 policy and distribution services, you can use Novell templates for sending patches out, tailor them to suit a particular environment or even write and compile your own patches for any third-party software you may run on your servers. Almost any software patch can be programmed into a ZENworks Server Management software package.

In particular, this paper addresses two configurations of software packages from Novell: the uncompiled software package, which is a template for modifying to your own environmental needs, and the compiled software package, which contains the files and logic for the installation but cannot be modified.

Mechanics of Tiered Electronic Distribution

ZENworks Server Management employs tiered electronic distribution, which is the main conduit for moving content, policies and patches throughout your server network. The simplest configuration for tiered electronic distribution is where one server operates as the distributor. The distributor's role is to maintain, build and schedule the sending of content policies and patches to all other servers, which are called subscribers. A subscriber's role is to receive, extract and apply content to its server.

The distributor controls who gets what content when. It schedules the building of patches, content and policies, including when it sends the data across the physical network. To ensure that tiered electronic distribution

does not duplicate bandwidth usage, it employs a mechanism known as “checkpoint” restart. In other words, when necessary, it stops a trans-network delivery at a scheduled time and then later continues where it left off.

Content, policies and patches are packaged into single units called distributions. Distributions compress the files and provide a single reference for all of your content. The design allows the distributor to build delta distributions; in other words, only the changes are sent across the wire, rather than the whole distribution.

Tiering and controlling the paths these distributions take as they move through your network can save CPU resources and WAN utilization. Here, intermediate servers are used as “holding stations.” Their job is simply to receive content from and send it to other leaf servers. If the patch is not destined for them, they do not extract and apply it; they simply hold it and forward it on to the appropriate server. These servers are called parent subscribers.

Today, most patches come as prepackaged units, as in the case of NetWare support packs. They can be downloaded from the Web as a single executable unit. Linux receives its updates in the form of a Red Hat Package Manager (RPM). Patches and products from Microsoft are prepackaged in the form of an Microsoft Installer. Both Red Hat Package Managers and Microsoft Installers contain files and logic for installation.

While ZENworks 6.5 Server Management provides the ability to leverage all of these package types, Novell also provides customers its own package type—server software package—for delivering the installation logic and files for patching Novell servers through ZENworks Server Management. The entire patch and the logic to install it is compiled into a single file with the extension .CPK, then placed into a distribution that can be transported through your tiered electronic distribution network.

Introduction to the Software Package Engine.

The server software package is one type of ZENworks Server Management distribution that tiered electronic distribution can push out through your network. The file first starts out as an uncompiled package (.SPK). For example, the eDirectory software package for NetWare is named *eDir873x_NetWare_Patch_ZFS.SPK* prior to compilation. This file includes the logic for the content and pointers to where to find the files. The compiled file (xxx.CPK) includes the content and the logic ready to apply to the subscriber-configured server. Files do not have to be part of the package as you will see in the Open Enterprise Server upgrade software package. You will separate the files into simple file distribution to be sent out. The software package is manifest in a Logic- only Package (.CPK).

The software package processor is available on all server platforms on which ZENworks Server Management runs, namely SUSE LINUX, NetWare 5 through NetWare 6.5, and Windows servers. Please see the online documentation for an exact list of what service packs are required prior to installation.

Each server software package consists of one main component and subordinate components. The main component contains variables and requirements that apply to the entire package. The subordinate components also contain requirements, but they apply only to one subsection of logic. Each component can perform the features outlined in the table below.

Component Features of Software Packages

Feature	Description	Example
Base component processes on requirements with specific values	<ol style="list-style-type: none"> 1. Operating system 2. Memory 3. Disk space 4. Set commands 5. Registry 6. Files 7. PRODUCTS.DAT 	This is useful if you want just one software package to apply to multiple operating-system versions. Each component can be set to execute only for that operating-system version.
Execute the following items prior to each component processing	<ol style="list-style-type: none"> 1. Load or unload NetWare loadable module commands 2. Load Java class 3. Start or stop services 4. Pre-execute a system, NetBasic or Perl script 	This is useful when you back up a directory. Before you back up the files, you can unload the NetWare loadable module, Java process or services loaded from this directory.
Copy files around the server volumes or up to the server from the package	Copy or move files on the local server or copy new files from the server software package to the server.	This is useful in copying files you are about to use as upgrade or patch. It is also useful in backing up files before patches and in placing them patch directories so they are ready to patch.
Manipulate strings within existing text files (or those just copied onto the server)	<ul style="list-style-type: none"> • Search and replace words in files. • Utilize an extensive array of tools. • Append or prepend. • Add words after or before. • Add lines before or after. • Nest variables here to add granularity. 	See the Open Enterprise Server upgrade section for a good example of how to use this feature. The Open Enterprise Server strings needed a silent install script written specifically for each server. (Use this feature and rewrite them when they are run.) This is also useful for changing the /etc/hosts setting on all servers.

Feature	Description	Example
Apply a common "Set Parameter" configuration to NetWare servers and set, remove or modify registry settings	Standardize settings, set and/or remove settings, and install or remove applications (Windows or NetWare registry-setting changes)	The upgrade compiled packages for network-management agents use these to insert new registry settings.
Add, replace or modify PRODUCTS.DAT entries on a NetWare server	Allow the installation of new applications	This feature is commonly used in NetWare and eDirectory patches.
Perform post-installation tasks	<ol style="list-style-type: none"> 1. Load or unload NetWare loadable module commands 2. Load Java class 3. Start or stop services 4. Run Red Hat Package Manager, Microsoft Installer or IPS scripts <p>(NOTE: To run a these, load the post-installation script and add the command line arguments. See the sections on eDirectory patches and the policy and distribution patches for examples of both.</p>	<p>See the ZENworks Server Management patch compiled package for an example of running Red Hat Package Managers.</p> <p>The eDirectory patches on NetWare launch IPS scripts with arguments; see these sections for details.</p> <p>See the eDirectory patches on Windows section for an example of example running a Microsoft Installer update with a software package.</p>

A graphical user interface will help you configure content application. You can set restrictions on each component; they can be as simple as a registry setting, a file, an entry in SYS:SYSTEM\PRODUCTS.DAT, or it can be as general as an operating-system version. For example, you could restrict a patch to be applied only to NetWare 6.5 but not to 6.0 or 5.1. If necessary, restrictions can be just as granular with the whole server software package.

Of course, Novell provides an array of ready-made compiled packages. ZENworks patches, Java update patches and ZENworks agent installations are available for download via the Novell Web site. You merely need to insert the relevant compiled package file into a distribution, schedule it, assign it a channel and send it out.

PATCHES AND UPGRADES AVAILABLE TODAY

Novell NetWare Service Pack

NetWare patching has been available as a ZENworks Server Management software package since ZENworks for Servers 3.0.2. It allows for the same installation logic as you would have on a normal NetWare support pack but switches to silent mode as soon as it is run within a ZENworks Server Management subscriber.

The main feature is an ability to authenticate to eDirectory using the CPHELPER.NLM module. This negates the dual need to both copy down the support pack and authenticate to eDirectory for each server patch. It includes two Java processes that can monitor a “sister” server to send alerts if the server never recovers from the reboot. The final task it performs is to launch the service pack’s own SPACK.IPS script in unattended mode.

Customizing the Software Package

Novell Technical Services has released this as a software package rather than as a compiled package. This format allows customers to tailor it for their specific needs. For example, if unloading backup software prior to running the support pack minimizes the user intervention for each server, you could easily customize the software package to achieve this prior to compilation.

Because Novell is not privy to your volume naming schemes, user IDs and tree names, you need to define these variables in the software package. The other variables should be left as the default values.

SPK Variables

Variable	Description
DEST_VOL	This is the location where the entire NetWare support pack will be extracted to prior to your running the installation.
USER_ID and PASSWORD	Use these two variables to define the user ID and password that the service pack uses to authenticate into eDirectory with. SECURITY NOTE: Disable this user ID after you have rolled out the service pack.
MINUTES_TILL_TIMEOUT	Leave this at the default time-out of 120. It refers to the the number of seconds the sister server will wait until it times out and starts sending traps to alert you of a server down condition.
ALERT_SERVER	Typically, this will be your ZENworks Management and Monitoring Site Server, HP-OpenView, Tivoli and so forth. It can be any device you are using to receive SNMP traps from your network.
ALERT_TREE	This is the tree that contains your NetWare servers.
ERROR_LOG	This is the location where you want the service pack to log its errors.

SPK Components

The software package includes eight components plus the main component.

Component Name	Description	Tab Name
Launch CPHELPER.NLM	This component launches CPHELPER.NLM with the variables %USER_ID% and %PASSWORD%. Each administrator must predefine these variables in the main component of the software package for their own eDirectory tree.	Post-installation
Copy NWSP	This component allows you to browse to your own directories where you are keeping a particular NetWare support pack. Once compiled, this component instructs the package processor to extract the full NetWare support pack to the location where you have defined the %DEST_VOL% variable. This is also defined in the main component of the software package.	Copy files
<p>NOTE: The next four components copy files and set up various Java processes to monitor the reboot. For this to work, you will need to be using a server-down policy for the server that monitors the reboot and the server getting patched. The PDF document contained within this patch shows you how. If you do not use a SNMP monitoring service in your network, components three through six can be deleted.</p>		
Copy SNMP Alertserverclass	This copies the required Java classes to the server in order to run the next three components.	Copy files
Run StartAlertserver	This runs the Java class.	Post-installation
Run AlertServers.ncf	This runs the NCF.	Post-installation
Delete Alertservers	You can delete the file because it is no longer needed.	Copy Files
Create the ZFS entry	The NetWare service pack installation script looks for a "ZFS" entry in SYS:SYSTEM\PRODUCTS.DAT to know when to use its unattended installation mode. Insert it into this component to make it ready to run the patch.	Products.dat

Component Name	Description	Tab Name
Launch the support-pack installation	This launches NWCONFIG and passes the following parameters in.	Post-installation
NOTE: Most of these values are kept in variables that you define in the main component of the software package as previously described.		

Open Enterprise Server Upgrade: NetWare to NetWare

This solution utilizes the silent-install abilities of Novell Open Enterprise Server. You will nest variables inside the software package that search and replace variables inside of the silent-install scripts of Open Enterprise Server. Each of the three text files required for running a silent upgrade of Open Enterprise Server needs to have specific values in place for each server on which it is run. Employ the subscriber's own default variables to place local values in the Open Enterprise Server upgrade scripts and files. All the variables used for this solution are outlined at the end of this section.

This software package is used to employ the silent install capacity of Open Enterprise Server. A full unabridged response file and example IPS and NCF files are found in Appendix A. Here, however, details are provided on the mechanics of the package and on how to manipulate the upgrade script variables that Open Enterprise Server will use to upgrade itself.

With this solution, you will need to modify, compile and test the software package using your own variables. Because this requires marrying two sets of variables and technologies, the recommended steps are listed in detail below.

Before you begin, you need to decide how to deliver the installation files to each server. This can be done using a plain ZENworks Server Management file distribution, which is recommended for two reasons:

- The files can be compiled into the compiled package and distributed at the same time, but the compilation takes some time and the compiled package file ends up being 1 GB in size.
- Deploying the files manually allows for easier troubleshooting as you can separate the mechanics of Open Enterprise Server installation from the software package processing.

Steps

1. As mentioned in the introduction, use the deployment manager to generate the three silent install files (RESPONSE.RSP, OES2UPG.NCF and RESPONS.RSP). Remember to replace the default RESPONSE.NCF file with OES2UPG.NCF.
2. To configure a silent install, launch the Open Enterprise Servers deployment manager. Then choose "Automate a installation" under "Install/Upgrade options." Then choose "Response File Generator." This guides you through the process of creating a silent install script. In using the tips below, you will have three files: response.rsp, response.ips and response.ncf. The first two files you'll manipulate. The last one is generated with the software package and named UPG2OES.NCF.
NOTE:The deployment manager of Open Enterprise Server launches off of the Operating System Open Enterprise Server CD-ROM.
3. Choose the option "Upgrade server locally."
4. (Optional) Choose whether or not you want to use a template.
5. When you are asked for the location of the source files, enter "sys:\oes." This value is written to the .ips file and will be changed if required by the software package variables. You will search for this exact string in component three of the software package.

6. The next screen requests for other variables: server, IP address and context. Fill them in. The values you enter here are meaningless because you have built the software package to localize all of these variables when it runs on each server; it runs its own NCF file to launch the installer. This applies to both source and destination eDirectory information.

NOTE: These values are editing in the sections NWI:MISC and MWI:NDS sections of the response.rsp file, which is removed and replaced for each subscriber.

7. Choose your license file location.
8. Specify the backup information you want. (Specifying none preserves disk space.)
9. Choose "Yes" for automating your reboot.
10. Choose "No" for "allow unsupported disk drivers."
11. Choose "Default" for the upgrade type.
12. Select "Open Enterprise Server" for the type of installation you require.
13. Choose the product options you want.

NOTE: On the screen where it says "Save response file," you need to choose these options:

- <Location on your Windows desktop> where you will temporarily store the response file before copying it up to the server. For example: c:\oes
 - <destination volume>/oes for the "Response file and licenses install location. (This is the location that the installer will look for to know what values to use on the silent install.) For example, for a simple install running from sys:\oes directory, choose sys:\oes at this time.
14. Copy the contents of both install CDs to c:\upgrade\oes unless you are going to use a file distribution. If you chose to use a file distribution, <anyvolume>OES is recommended. Both the product.ni and postinst.ini files should be in that directory now. When you copy the second CD-ROM production over the top of the first, you will be warned about overwriting files. Ignore these.
 15. Copy response.ips and response.rsp created in step 1 to the same directory as the source files. You don't need response.ncf because the software package will build it from scratch creating OES2UPG.NCF.
 16. Edit the response.rsp file to remove both the [NWI: MISC] and [NWI: NDS] sections.
 17. If you decided in step one to use a file distribution, create one using these source files and distribute to a common location on each server. The documentation assumes SYS:\OES, but this location is set by a variable in the software package, so it can be on a different volume. To change this for your environment, go to the main component of the software package called Open Enterprise Server Upgrade and click on "Tab Variables." The variables in question are "dest_vol" and "dest_dir".
 18. Copy the CPHELPER.NLM file from support.novell.com to c:\upgrade. One can be found here: <http://support.novell.com/cgi-bin/search/searchtid.cgi?/2969517.htm>
 19. Copy the OESUPGRADE.SPK to c:\upgrade.
 20. Insert software package. For more information on how to do this, see www.novell.com/documentation/zenworks65/index.html?page=/documentation/zenworks65/sadmin/data/acwmjjm.html#acwmjjm
 21. Adjust the six variables on the package.

NOTE: If you chose to use a file distribution to distribute the Open Enterprise Server upgrade software, you must ensure the dest_vol and dest_dir variables match the location of the distributed files. These variables are explained later.
 22. If you chose to use a file distribution, remove the "Copy OES Files" component from the software package.
 23. If you chose to have the CD-ROM install files included in the compiled package, follow the instructions displayed on the description tab of the component "Copy OES Files."
 24. If you chose to include the files and logic in the same compiled package, add all of the files and folders from c:\upgrade\oes to the "Copy OES Files" component.
 25. Compile. If you get exceptions in the compilation, load ConsoleOne in debug mode to troubleshoot.
 26. (Optional) You can choose to add a component that unloads your virus scanner NetWare loadable module or any other NetWare loadable module you might want to prior to the upgrade. Simply insert a component before the "Launch OES Upgrade" component and use the Post-installation tab to load or unload the same.

27. It is important to test your work on a test NetWare 6.0 server as any bad syntax errors for the variables used can have disastrous effects on your server. Copy both the files and the software package to your test server and run manually using "package process." If you use file distribution, ensure that the files have distributed to this server successfully first before you process the software package. The software package will fail with exceptions as it looks for a sys:\oes\ directory if you don't.

NOTE: Keep in mind that each time you run the tests, the text files response.rsp and the NCF and IPS files are being modified. This has created multiple sections and modifications to these files that will dilute the integrity of your tests. You may want to paste in new copies of these files each time you use the package process command in your tests.

28. Distribute the software package. See the online documentation at www.novell.com/documentation/zenworks65/index.html?page=/documentation/zenworks65/sadmin/data/acwmjjm.html#acwmjjm for detailed instructions.

Advanced Notes

1. The extraction of the compiled package and the initial file copy take a long time; the server may appear to have stalled. Be patient.
2. The distribution and the resultant file extraction end up being over 1 GB each. The volume used must be able to accommodate this amount of data. If the SYS volume is used, make sure there is enough room for the distribution *and* the added requirements of the upgrade.
3. To reduce the size of the distribution, try to delete unused products from the products directory before setting up the copy step. Do this carefully as some products are dependent on others.
4. Once the upgrade is complete, delete the following directories:

- sys:\oes (or whatever was actually used)
- The ZENworks Server Management distribution directory—This must be done manually with version 3.02 but will happen automatically when the server is removed from the channel with 6.5.

NOTE: In both versions of the product, the extracted distribution is never deleted. A second software package can be sent with a component to remove this if necessary.

5. For upgrading Open Enterprise Server NetWare servers in a cluster, add this component table and details. There's a good Knowledgebase article (TID), number 10074377, that describes why it's set so low for cluster services. Setting back to the default only masks the issue, but the issue exists only during the actual upgrade process. As long as cluster services are unloaded (which this script does), there's no reason to have it set to anything other than the NetWare default during that time. Since the timing change is only changed at the command line, it is not persistent. So when the server does its final reboot and rejoins the cluster, it is back to its regular and appropriate setting of eight seconds
6. If during the upgrade process if you have trouble restarting, you can use the down server upgrade process. Boot from the 6.5 install CD-ROM. Hit any key to have it stop. One of the options presented is to add parameters to the installation. Here you add [INST-upgrade RF=sys\oes\response.ips] You must have at least attempted an upgrade for this to work. (In other words, do not try this without a failed upgrade in place.)

Components of the Open Enterprise Server Upgrade Software Package OESUPGRADE.SPK

Component Name	Description	Tab Name
Open Enterprise Server upgrade	Main component contains variables used for launching scripts and those nested to replace in the upgrade scripts and response files.	Variables

Component Name	Description	Tab Name
Launch Cphelper	Launches the CPHELPER NetWare loadable module that facilitates logging in. It's the same NetWare loadable module that NetWare Support Packs have historically used.	Copy files, Post-Installation
Adjust IPS file	The IPS script that launches the upgrade needs to be localized in case volumes other than SYS are used to store the Open Enterprise Server upgrade software.	Text files
Customize response file, step 1	This customizes the response.rsp file using nested variables from this software package. Replace NWI:MISC and NWI-NDS sections. See the section "Variables you localize or change for each server deployment" for more information.	Text files
Customize Response File, step 2	Remove the fully qualified distinguished name for the server and leave context only.	Text files
Create launch NCF	This creates a replica of the response.ncf (but with localized variables) and places it in the Open Enterprise Server directory.	Text Files
Copy Open Enterprise Server Files	This copies the CD-ROMS contents to each server.	Copy Files
Launch Open Enterprise Server upgrade	This launches the NCF file created above.	Post-installation

Component to Add for Clustering

Add before the last component.

Component Name	Description	Tab Name
Unload cluster services	Leave the cluster and change the CPU Hog Timeout setting from eight seconds (cluster default) to 60 seconds (NetWare default).	Post-installation Script (NCF): cluster leave uldncs.ncf set CPU hog timeout amount=60

Variables for Open Enterprise Server Upgrade Scripts and Packages

System Variables	No Configuration Necessary—Default on All Subscribers
IP_ADDRESS	IP address of server
SERVER_NAME	Name of server
SERVER_DN	FQDN of server
Variables within the Software Package (Adjustable to Suit Your Environment)	
DEST_VOL	The destination volume where the Open Enterprise Server upgrade software is stored
DEST_DIR	The directory where the Open Enterprise Server upgrade software is stored
TREE	Your eDirectory tree name
USER_ID	Your Admin user ID or equivalent NOTE: If you install products with this Open Enterprise Server upgrade, this user will be the default user. In the case of iManager, this is the only user that can add modules.
USER_CONTEXT	The context for your admin ID or equivalent
PWD	The password for your admin ID or equivalent (Keep in mind you need to change this at the end of your upgrades as this is stored in clear text.)

Customize Response File Step Components

This component replaces two sections of the response.rsp file: [NWI-MISC] and [NWI-NDS]. You will employ nested variables techniques that localize the scripts and response files for each server. These are outlined in the sections below; example files are included in Appendix A. The values inside the percentage keys are the nested variables. In other words, the values of these variables be used for each of the variables going forward in the Open Enterprise Server upgrade scripts.

[NWI-MISC]
Source Server Name=%server_name%
Source Location Path=%dest_vol%\%dest_dir%
Prompt=False
Source Domain Address=%ip_address%
Source Domain Address type=UDP
Source Tree Name=%tree%
Source Server Context=%Server-dn%
Source User Name=%user_id%
Source Context=%user_context%
Relogin Password=%pwd%
[NWI-NDS]
Tree Name=%tree%
New Tree=false
Server Context=%server_dn%
Admin Login Name=%user_id%
Admin Context=%user_context%
Admin Password=%pwd%
Prompt=false

Because one of the nested default variables returns the fully qualified domain name, and because the silent install requires merely the context of the server, you will manipulate this slightly by deleting the server name from the FQDN. See the component called “Customize Response File, step 2” for details on how to do this.

Novell eDirectory Patching

Starting with eDirectory 8.7.3.4, installation logic was put in place to allow for unattended patching of NetWare servers using ZENworks. The same ability was implemented for Windows starting with eDirectory 8.7.3.6. It is available for both the security and core eDirectory patches. In essence, you can now apply the patches to any eDirectory 8.7.3.x version using this method.

NetWare

The same pre-inserted product entry (ZFS) in the PRODUCTS.DAT file is required to instruct the patch that it should run in the unattended installation mode. In other words, the entire eDirectory patch is copied to the SYS

volume of the server, a “ZFS” entry is created in PRODUCTS.DAT and then the eDirectory patch is launched for NetWare using the NWCONFIG utility.

The patch contains a software package available for you to compile for the eDirectory patch version you have selected. Simply copy the eDirectory patch into the directory c:\EDIRPATCH and expand it. Using the ZENworks Server Software Package GUI, select the first component, go to the Copy Files tab, browse to the NetWare directory to select the entire directory and then compile it.

SPK Variables

Component	Description
DEST_VOL	This should remain set to the default value of “sys.”

SPK Components

Component	Description
Copying Files	This copies the eDirectory patch in its entirety up to the sys: volume.
Create the ZFS Entry	This creates the “ZFS” entry in preparation for the installation script to run.
Starting the eDirectory patch installation	Launch NWCONFIG with the following parameters: <i>b=sys:\NetWare\87patch.ips s=sys:\NetWare\ d=sys</i> Note: The parameters here are standard for launching any IPS script through NWCONFIG.NLM. b: indicates to the launch script to use eg: sys:\NetWare\87patch.ips. s: indicates the location of the source files that this script will install. d: indicates the destination root location to which the installer will copy the files.

Installation Mechanics

The STARTUP.ILS file checks for the existence of the temporary “ZFS” product entry in PRODUCTS.DAT and—if it exists—sets a “ZFS” variable to TRUE. This variable is used to determine if the script is running in the unattended mode, which allows message prompts and inputs to be skipped.

87PATCH.ILS is the core script that contains the main copy file commands that actually transfer the files from the patch into the relevant directories on the server.

If any errors occur, the script will display the error and will halt the installation. For example, if the patch is being installed onto a version of eDirectory that is not 8.7.3, including an early version of Novell Directory Services, the script will fail. (If this happens, you will see an error message.) You can also configure the software package to take care of this scenario.

At the completion of the script, the “ZFS” entry is removed from PRODUCTS.DAT. This is so that any other scripts that are subsequently run will not automatically run in the unattended installation mode.

In eDirectory 8.7.3.7 and later versions, two additional changes were made. The first change relaunches ZENworks Server Management after the installation is complete. This allows you to regain control of the server via ZFS. The second change is the check for the existence of product entry “ZFS_RST.” You can create this key by modifying the software package manually and adding the entry in the same manner as the “Create ZFS Entry” component. If this entry exists along with the “ZFS” entry, once the patch is successfully installed, a “RESET SERVER” command will be issued at the console. This resets the server, which will activate the new eDirectory version when the server starts up. Until eDirectory 8.7.3.7 is released with these changes, the reboot must be issued manually. Using iManager here is standard; in addition, you could also activate a policy to reboot the server overnight.

Windows

Beginning with eDirectory 8.7.3.6, the eDirectory team implemented a Microsoft-Installer-based installation. To allow for older Windows NT servers that may not have the latest Microsoft Installer updates, wrap this file inside a wise install-based executable. This enables unattended installations.

The software package file is named eDir873x_Win32_Patch_ZFSSPK. The software package contains two components with two variables defined in the main component.

SPK Variables

Variable	Description
DEST-PATH	This needs to be defined so you can store the executable.
EDIRPATCHVER	This needs to be defined for each patch version you define in the compiled package.

SPK Components

Components	Description
Copying files	This copies the file eDir873x_Win32.exe to the Windows server. Note: “X” is the number specified in the %EDIRPATCHVER% variable as above.
Running the patch	This launches the patch with these options: /I <path and filename for install.log> This creates and logs information and error messages. /qr ALLUSERS="1" This nstalls the patch in the unattended mode.

GroupWise Upgrades and Patching

With the advent of GroupWise 6.5, software packages that allow you to either patch or upgrade GroupWise have been written and tested. GroupWise 6.5 can be patched with SP4 and GroupWise 6 SP4 can be upgraded to GroupWise 6.5. The provided software packages can easily be modified to handle other patch or upgrade version combinations.

GroupWise Patching

GroupWise message transfer and post office agents can be stopped simply by unloading the agents on NetWare servers or by stopping the named service on Windows servers. You can restart these agents by loading the agents with the correct startup file on NetWare servers and restarting the named service on Windows servers. This greatly simplifies the processes required to update the standard domain and post-office configurations. A standard configuration refers to a single domain and single post office running on a single server. The logic provided in these software packages can be modified as necessary to fit your environment.

It is always good practice to back up the domain and post office database when any update occurs. The sample patch packages include components that back up the recommended files.

The GroupWise software package component structure for both NetWare and Windows servers are the same, although there are separate packages for each server operating system. The main reason for separate packages is the differences in starting and stopping the agents.

1. As each agent is has unique startup files on each server, there are a number of variables that need to be defined at each subscriber. This extra work on the front end provides greater flexibility and allows for simpler, more general server software packages. Once these variables are defined, all future GroupWise patches and upgrades can use them to reduce and simplify future updates.
2. Back up the files.
3. Copy the patch files in place.
4. Restart the services.

As each agent is has unique startup files on each server, there are a number of variables that need to be defined at each subscriber. This extra work on the front end provides greater flexibility and allows for simpler, more general server software packages. Once these variables are defined, all future GroupWise patches and upgrades can use them to reduce and simplify future updates.

GroupWise 6.5 SP4 software package mechanics.

The GroupWise 6.5 patch for both NetWare and Windows includes the logic for a particular support pack. This can be updated for future support packs by modifying the "Copy Patched Files" component.

Subscriber Object Variables

Variable	Description	Example
DOM_NAME	GroupWise domain name	mydomain
OM_PATH	Domain path including the volume or share name path	vol:mail\mydom
DOM_CONFIG	Name of the domain startup file	mydom.mta
PO_NAME	GroupWise post-office name	mypostoff

Variable	Description	Example
PO_PATH	Post-office path, including volume or share name and path	share\mail\mypo
PO_CONFIG	Name of the post office startup file	mypo.poa
GWAGENTS_PATH	Volume or share name and path to the GWAgents location	sys:\system\gwagents
BACKUP_PATH	Volume or share name and path to the backup directory location	share\backup

Note: All variables defined above end without a slash.

Software Package Components (NetWare)

Component	Description
Stop GroupWise NetWare services	Unload GroupWise. Run the NCF scripts that unload your GroupWise NetWare loadable modules—NLMS
Back up existing files	Make backup copies of the existing agent, database and dictionary files uses the values of GWAGENTS_PATH, DOM_PATH, PO_PATH and BACKUP_PATH
Copy patched files	copy patched files to the correct locations uses the values of GWAGENTS_PATH, DOM_PATH and PO_PATH
Startup GroupWise NetWare agents	Load MTA and POA using the startup file parameters (Uses the values of DOM_CONFIG and PO_CONFIG)

Software Package Components (Windows)

Component	Description
Stop GroupWise Windows services	Stop the named GroupWise services (see note below); this uses the values of DOM_NAME and PO_NAME. (See this component for details.)
Back up existing files	Make backup copies of the existing agent, database and dictionary files; uses the values of GWAGENTS_PATH, DOM_PATH, PO_PATH and BACKUP_PATH
Copy patched files	Copy patched files to the correct locations. Uses the values of GWAGENTS_PATH, DOM_PATH and PO_PATH
Start up GroupWise Windows services	Start the GroupWise named services. Uses the values of DOM_NAME and PO_NAME.

Component	Description
<p>Note: The POA service may take longer to fully stop than the Windows service interface properly recognizes. Add a pre-distribution action to the distribution used to deploy this package to also stop the POA service. This will begin the process of stopping this service before the package calls for this stop. This workaround may be required in your environment.</p>	

GroupWise Upgrading

This software package automates the GroupWise upgrade process for secondary domains and post offices that are running in a standard configuration. A standard configuration refers to a single domain and single post office running on a single server. The logic provided in these software packages can be modified as needed to fit your environment.

Your primary domain (and a post office on the same server) should be upgraded manually using the installation software with GroupWise 6.5. This ensures that all initial steps (schema upgrade, software distribution directory, ConsoleOne and so forth) are upgraded correctly. Additional gateways will need to be updated using the installation routines provided with the GroupWise software.

This software package is based on using the primary software distribution directory for the source of files used to upgrade the domain and post office agents, as well as the domain and post office dictionary files. The source of these files needs to be modified in the file copy components (component three) of the package before you compile the package for distribution.

Upgrading GroupWise is similar to the GroupWise patching process. The main difference is unloading a different set of files than what is copied as part of the upgrade process. Backing up existing files moves some files rather than copying them to keep your production directories clean without old files remaining.

GroupWise message transfer and post office agents can be stopped simply by unloading the agents on NetWare servers or stopping the named service on Windows servers. Restarting these agents can be done by loading the agents with the correct startup file on NetWare servers and restarting the named service on Windows servers.

It is always a good practice to back up the domain and post office database when any update occurs. The sample patch packages include components that back up the recommended files.

The GroupWise software packages component structure for NetWare and Windows servers are the same, although there are separate packages for each server's operating system. The main reason for separate packages are the differences in starting and stopping the agents.

1. Stop the services.
2. Back up the files.
3. Copy the upgrade files in place.
4. Restart the services.

As each agent is unique with unique startup files on each server, there are a number of variables that need to be defined for each subscriber. This extra work on the front end provides much greater flexibility and allows for simpler, more general server software packages. After these variables are defined, all future GroupWise patches and upgrades can use them to greatly reduce and simplify future updates.

Subscriber Variables: To Be Defined on Each GroupWise Subscriber Object.

Component	Description	example
DOM_NAME	GroupWise domain name	mydomain
DOM_PATH	Domain path including the volume or share name path	vol:mail\mydom
DOM_CONFIG	Name of the domain startup file	mydom.mta
PO_NAME	GroupWise post office name	mypostoff
PO_PATH	Post office path, including the volume or share name and path	share\mail\mypo
PO_CONFIG	Name of the post office startup file	mypo.poa
GWAGENTS_PATH	Volume or share name and path to the GWAgents location	sys:\system\gwagents
BACKUP_PATH	volume or share name and path to the backup directory location	share\backup

NOTE: All variables defined above end without a slash.

GroupWise 6.0 SP4 to 6.5 SP4 Upgrade for NetWare.

The GroupWise 6.0 to 6.5 upgrade for NetWare includes the logic for version 6.0 SP4 to 6.5 SP4. This can be updated for future support packs by modifying the “copy patched files” component. If your existing system is not GroupWise 6.0 SP4, you can modify the “stop services” and “backup files” components to match your system.

Components of the Software Package

Component	Description
Stop GroupWise NetWare services	Unload the GroupWise 6.0 SP4 NetWare loadable modules
Backup existing files	Make backup copies of the existing agent, database and dictionary files (uses the values of GWAGENTS_PATH, DOM_PATH, PO_PATH and BACKUP_PATH)
Copy patched files	Copy patched files to the correct locations (uses the values of GWAGENTS_PATH, DOM_PATH and PO_PATH) See this component for details to add patch files before you compile.

Component	Description
Startup GroupWise NetWare agents	Load MTA and POA using the startup file parameters (Uses the values of DOM_CONFIG and PO_CONFIG)

GroupWise 6.0 SP4 to 6.5 SP4 Upgrade for Windows

The GroupWise 6.0 to 6.5 upgrade for Windows includes the logic for version 6.0 SP4 to 6.5 SP4. This can be updated for future support packs by modifying the copy patched files component. If your existing system is not GroupWise 6.0 SP4, you can modify the stop services and backup files components to match your system.

Components of the Software Package

Component	Description
Stop GroupWise Windows services	Stop the named GroupWise Services (see note below) (Uses the values of DOM_NAME and PO_NAME)
Back up existing files	Make backup copies of the existing agent, database and dictionary files (Uses the values of GWAGENTS_PATH, DOM_PATH, PO_PATH and BACKUP_PATH)
Copy patched files	Copy patched files to the correct locations (Uses the values of GWAGENTS_PATH, DOM_PATH and PO_PATH)
Start up GroupWise Windows services	Start the GroupWise named services (Uses the values of DOM_NAME and PO_NAME)

Note: The POA service may take longer to fully stop than the Windows service interface properly recognizes. Add a pre-distribution action to the distribution used to deploy this package to also stop the POA service. This begins the process of stopping this service before the package calls for this stop. This workaround may be required in your environment.

Novell Java Virtual Machine for NetWare Patching

This software package is provided as a compiled package. With the release of each Java Virtual Machine support pack, Novell provides the compiled package file via the support.novell.com/downloads site and via the developer.novell.com/ndk site. You merely have to download the file, insert it into a distribution and send it out to your NetWare servers.

The file name for the compiled package is `jvm142sp<support pack #>.cpk`. It is also available on the support.novell.com site at: <http://support.novell.com/cgi-bin/search/searchtid.cgi?/2971431.htm>

Components of the Java Virtual Machine Software Package

Check Java Virtual Machine version	Copies in CHKJVM.NLM, CHPJVM.NLM, the STOPZFS.NCF and the JAVA.ZIP file. Runs the CHKJVM.NLM that will leave a file on the sys volume letting the next nlm, CHPJVM.NLM. Run if it's a compatible Java Virtual Machine version to patch.
Stopx	Closes the NetWare GUI
Clean up	Cleans out the NLMs that it doesn't need anymore
Restart Java	Launches the STOPZFS.NCF file, which runs the following: A. STOPZFS.NLM: Gracefully closes down ZENworks Server Management B. CHPJVM.NLM: Registers all the running Java processes, shuts down the Java Virtual Machine, unzips java.zip and then brings the Java Virtual Machine up and restarts all of its previously running java processes.

NOTE: Initially, the only supported patch is from Java Virtual Machine 1.4.x to the latest support pack of Java Virtual Machine 1.4.2. There is no need to reboot the server when using this compiled package.

Upgrading and Patching Your ZENworks Server Management Environment

Each major upgrade with ZENworks Server Management contains two methods for the upgrade. One is via a software package and one is via the more traditional setup GUI. Outlined below are details about where you can find information about all of these forms of software packages.

Upgrading Server Inventory

This software package allows you to automatically upgrade the following Server Inventory components:

- Inventory server on NetWare 6.0 or later and Windows 2000 server
- Inventory Agent on NetWare 5.1 SP7 or later and Windows 2000 server

The file name is ZFS65_invrem.cpk. For additional information, see this link:

www.novell.com/documentation/zenworks65/sminstall/data/btebck1.html#btebck1

Upgrading Remote Management

This software package provides a means to upgrade Novell ZENworks for Servers 3.x Remote Management to ZENworks 6.5 Remote Management. For additional information, see this link:

www.novell.com/documentation/zenworks65/sminstall/data/btfv1xn.html#btfv1xn

The file name is ZFS65_remmgmt.cpk.

NOTE: If you wish to upgrade both at the same time, use this file: *ZFS65_invspagn.cpk*.

NOTE: Each of these software packages log its successes and failures in text logs.

- On a NetWare a server: sys:\etc\cpk65logs\cpk65_invsrv.log, sys:\etc\cpk65logs\invsrv_nw_files.log and sys:\etc\cpk65logs\cpk_ndsupdate.log
- On a Windows server: %windir%\cpk65logs\cpk65_invsrv.log, %windir%\cpk65logs\invsrv_win_files.log and %windir%\cpk65logs\cpk_ndsupdate.log

Upgrading Management and Monitoring Services Agents

This set of software packages make extensive use of variables. Variables allow the extraction and installation of software to multiple file systems using only one distribution. For example, on NetWare, the advanced-trending agent may be installed on the DATA volume, but the same agent on Windows is in the D: drive. Using variables, one software package can extract to each operating system's file system.

The Traditional and Advanced Trending Agent

This agent is typically installed to a directory off the SYS volume called ZFS_agnt. However, in some cases on a NetWare server, it can be installed to sys:\system\ZFS_agnt. The filename is ZFS5_mms_mgmtagnt.cpk. On Windows, it is also installed to a directory called ZFS_AGNT, but it can of course be on any drive letter. With Linux, the same directory structure occurs but is not relevant because you use Red Hat Package Managers to install and upgrade on Linux.

Here is a summary of the variables that need to be defined:

Variables

Variables on NetWare	Description
OLDNMAPATH	This defines the path of the older version of the agent installed. For example: sys:\zfs_agnt
TARGETIP	This is the IP address of the Siteserver, HP OpenView or Tivoli management station. It defines the IP address where you want to send the trap.

Variables on Windows	Description
MSAGENTPATH	This defines the path to install the agents. For example: c:\zenworks.
REPLACEHOSTMIB	This defines whether you want to install the Microsoft * or Novell Hostmib MIB. "Yes" means Microsoft *. (1=Yes, 0=No) For example: 0.
TARGETIP	This the IP address of the Siteserver , HP OpenView or Tivoli Management station. It defines the IP address where you want to send the traps.

Variables on Windows	Description
GTRENDPATH	This is the subdirectory under the Windows directory (c:\winnt) where the Gtrend files are saved. For example: c:\winnt.

Upgrading the Advanced Trending Agent

Use the zfs65_mms_trendagnt.cpk file to install the advanced trending agent. All variables defined used in this package are the same as the traditional and advanced trending agents. For more discussion on these agents, see: www.novell.com/documentation/zenworks65/sminstall/data/btfywzb.html

The LANalyzer Agent

This agent allows for the installation and/or upgrading of the of RMON-based sniffer agents throughout your network. They can either run on NetWare or Windows servers. Using ConsoleOne you can run, analyze and save sniffer traces.

For the zfs65_mms_lanzagnt.cpk file, create and initialize the following package processor variables in the subscriber objects' properties:

Use the same variables as above for the agent.

NOTE: For a more detailed explanation on these. visit the online documentation at www.novell.com/documentation/zenworks65/index.html?page=/documentation/zenworks65/sminstall/data/btfywzb.html#btfywzb

Upgrading Tiered Electronic Distribution Using Itself

It's vital also to provide an automated mechanism for upgrading and patching the Tiered Electronic Distribution system. These two files provide one compiled package that can be used to upgrade any subscriber or distributor from ZENworks Server Management 6.0 (3.0.2) to ZENworks 6.5 Server Management.

No variables need to be defined on the subscriber object. However, it is worth noting some of standard variables/paths. Here, the word "variable" is really a constant for obvious reasons—they are predefined.

Predefined Variables of the Compiled Package

Variable	Value
target_path	sys:\temp\zfs65PolyDist\upgrade
target_path_nt	%systemdrive%\temp\zfs65PolyDist\upgrade
target_path_solaris	/usr/ZENworks
target_path_linux	/usr/ZENworks
target_jvm_netware	sys:\java

There are two files that are used to upgrade and they are found on the Companion 2 CD-ROM of the ZENworks suite. They are:

- ZFS65_polyDist.cpk. Contains all software for any subscriber operating system type you might have deployed.

- ZF2302_nw6_PDSfix.cpk. Needed as a make-ready for NetWare 6 servers to upgrade to ZENworks Server Management 6.5 via the above software package.

Components of the ZENworks Server Management Upgrade Package

There are 27 components for this software package. They are repeated for each operating system version that is supported, so use the requirements tab at the component level is used to ensure files and settings meant for NetWare don't get placed on Windows or Linux subscribers and vice versa.

Component	Description
Update tiered electronic distribution for <OS brand>	Updates the .jar files for Tiered Electronic Distribution. Mainly files in the <installed volume>Zenworks/pds/ted directory.
Update ZWS files for <OS brand>	Updates the XMLRPC side of ZENworks Server Management (ZEN web server). Mainly files in the <installed volume>ZENworks/ZWS directory.
Update ZENworks.properties files	Makes an entry in the ZENworks.properties file to track which version of ZENworks is present.
Update ZFS for <OS brand>	Updates the ZENworks Server Management files, including files for software packaging, policy processing and text file manipulation—mainly files in the <installed volume>ZENworks/PDS/smanager directory
Update misc files for <Os brand>	Updates common Java files
Update ConsoleOne files for <OS brand>	Updates the ConsoleOne files to 6.5 snap-ins
Update Sybase for <OS brand>	Updates the Sybase database
Update Java Virtual Machine/JRE for <OS Brand>	Updates either the Java Virtual Machine files for NetWare or the JRE files on Windows
Update version component	Copies in a utility that updates the NetWare and/or Windows registry
Check file utility for <OS brand>	Checks the files that are copied
Kill copy restart for <OS brand>	Runs an EXE or an NLM to restart the ZFS service (If it is on NetWare, it registers the Java processes before it restarts.)
Note: Linux updates come as Red Hat Package Managers but are copied down from within the compiled package and then run from the command line to install. Solaris comes as a package and is installed via the command line. Their components are as follows:	
Update 6.5 for Solaris	Copies in the JRE and ZFS packages and files as ready to update

Check file utility for Solaris	Restart for SolarisRuns j2re-1.4.1_03 to update JavaRuns Pkgadd -n -d novlzfs-6.5.pkgRuns / etc/init.d/novell-zfs startRuns Pkgrm -n ZFDted
Update 6.5 for Linux	Checks files
Check file utility for Linux	Restart for SolarisRuns j2re-1.4.1_03 to update JavaRuns Pkgadd -n -d novlzfs-6.5.pkgRuns / etc/init.d/novell-zfs startRuns Pkgrm -n ZFDted
Restart Linux	<ul style="list-style-type: none"> Runs /etc/init.d/zfs stop (stops the service) ● Runs rpm -i -force j2re-1_4_1_03-fcs-linux-i586.rpm ● Runs rpm -UVH novell-zen-zfs-6.5-0.i.386.rpm ● Run /etc/init.d/novell-zfs-start rpm -e ZFSTed

APPENDIX A: LOCALIZED SILENT INSTALL SCRIPTING EXAMPLE

The software package is designed to insert nested variables into two sections of the response.rsp file: [NWI:MISC] and [NWI:NDS]. Create the UPG2OES.NCF file for each server you run on and modify the respons.IPS script for the volume from which you launch. In short, use the four variables below, which exist on each subscriber to create your own localized versions of these three scripts and text files. See the tables in the Open Enterprise Server section in this document for more information.

Source Server Name = %SERVER_NAME%

IP address = %IP_ADDRESS%

Source server context = %SERVER_DN%

Server context = %SERVER_DN%

As you can see from the above example, the same variable— %SERVER_DN%—is used twice. This is because the silent install uses two variable names for one value. The install script wants only the context for which the server resides and not the fully qualified distinguished- FQDN. In the component “Customize Response File, Step 2,” strip off the server name and insert that value.

The Full Response.rsp file.

; The [Selected Nodes] section must remain synchronized with RESPONSE.TXT (excluding the prompt key).

; It also should be populated with the entries found in a custom upgrade with all products selected.

; The "Patterns" key should be set to "NONE."

[Selected Nodes]

prompt=false

Non-Changeable Products=Script,Imonitor,Portal,ConsoleOneProducts,6pkLdap,SMS,Novell Certificate Server,NWNMAS,Apache2 Admin Server,Native File Services,Tomcat4,NWEMBOX,PHP,Perl,NSN,Beans,NSBS2Copy,NSBS2Extra

Pattern Products=Script,Imonitor,Portal,ConsoleOneProducts,6pkLdap,SMS,Novell Certificate Server,NWNMAS,Apache2 Admin Server,Native File Services,Tomcat4,NWEMBOX,PHP,Perl,NSN,Beans,NSBS2Copy,NSBS2Extra

DestinationNode=HealthChecks

HealthChecks=NetWare Services

NetWare Services=NetWare OS,Patterns,Products,NWUpdateGroup

NetWare OS=NetWare65OS,NICl,Protocols,Time Zone,DS_Install,LicensePrompt,W0 install,NFA Install Module,Bash,RedCarpet,Rug,DOS_INST,SYS_INST,Support Pack

Bash=netware65RPM,bashFiles

RedCarpet=rcdFiles

Rug=python,rugFiles

Patterns=NONE

Products=Novell Certificate Server,6pkLdap,Imonitor,Apache2 Admin Server,Tomcat4,Apache2 Webserver,Portal,SMS,ConsoleOneProducts,NDPS,NWNMAS,Native File Services,DNSDHCP,NWEMBOX,Beans,NSN,Perl,PHP,OpenSSH,iManager2.5 Product

Novell Certificate Server=CertServ System Files,CertServ Public Files

Imonitor=imonitor_DFG

Apache2 Admin Server=AApache2,AAp2Conf,adminsrv,welcome

Tomcat4=Tomcat zip file,examples zip file,Tomcat admin configuration,Novell Tomcat Startup Scripts

Apache2 Webserver=Ap2webcf

Portal=portalzip,httpstkzip

SMS=SMSSystemFiles

ConsoleOneProducts=ConsoleOne,Reporting Snapin

ConsoleOne=c1_core,c1_win32,c1_nw.zip

Reporting Snapin=c1_rpt

NDPS=NDPS Server Files,NDPSResourceFiles

NDPS Server Files=Xfer IPP Login Files,IPP Login Files,IPrint Files,Gateway Files

NDPSResourceFiles=NDPS Banner,NDPS Font,NDPS Prndef,NDPSPrndrv

NDPSPrndrv=NDPSPrndrv-HP,NDPSPrndrv-Xerox,NDPSPrndrv-Kyocera,NDPSPrndrv-Lexmark,NDPSPrndrv-IBM,NDPSPrndrv-Oki,NDPSPrndrv-QMS

NDPSPrndrv-HP=HPDrivers

NDPSPrndrv-Xerox=XeroxDrivers

NDPSPrndrv-Kyocera=KyoceraDrivers

NDPSPrndrv-Lexmark=LexmarkDrivers

NDPSPrndrv-IBM=IBMDrivers

NDPSPrndrv-Oki=OkiDrivers

NDPSPrndrv-QMS=QMSDrivers

FTP Server=ftpfilezip

NetWare Web Search=NSearch1, Templates, Sample Templates

QuickFinder Server=QFSearch1, QFTemplates, QF Sample Templates

ipWanMan=dfgWanManSysSystem

NWNMAS=NMAS Server System Files, NMAS Methods

Native File Services=NFSNIS

NFSNIS=NFSNIS-NLMs

DNSDHCP=dnipFiles

iFolder=iFolder install module

iFolder install module=iFolder zip file, ifolder_en

NWEMBOX=embox_DFG

NWSNMP=SNMP_SYSTEM_DFG, SNMP_ETC_DFG

MySQL=MySQL install module

MySQL install module=Product zip file, Configuration zip file, PHP Configuration zip file, JDBC Driver zip file, Manage zip file

Tomcat5=tomcat5zip, tc5exmpl, nwtc5bin

j2ee=exteNd Application Server zip file, exteNd Application Server NetWare Overlay, exteNd Application Server NetWare configuration files, exteNd Application Server IP address Management zip file

audit=auditpa, auditin, Nsure auditing zip file, Nsure auditing Isc files, Nsure auditing applications zip file

Beans=BEANS_ZIP

iStorage=components

NSN=NSN install module,UCS install module

NSN install module=NSN Product zip file

UCS install module=UCS Product zip file

Perl=Perl install module

Perl install module=Perl Product zip file

PHP=PHP install module

PHP install module=PHP Product zip file

OpenSSH=SSH-Config,SSH-Core,SSH-Docs

Wan Connectivity=WanConnectZip,UpgradeFrom

eGuide=eGuideFiles

iManager2.0

Product=EXTEND_ZIP_FILE,EXTEND_IPCONF_FILE,EXTEND_CONF_FILES,EXTEND_PACKAGE_FILES,EXTEND_XAR_FILE

iManager2.5 Product=IMANAGER_ZIP_FILE,IMANAGER_CONF_FILES,IMANAGER_PACKAGE_FILES

VirtualOffice1.3=VirtualOfficePackages1.3,ConfigEG

Virtual Office=vofficeRPM,vofficewelcomeRPM,eGuideCf

NWUpdateGroup=NWUpdate

[Product List]

Essential Products=NetWare OS,HealthChecks,NetWare Services

[NW! :Install Options]

Upgrade=True

Express=TRUE

Prompt=FALSE

Backup=TRUE

Backup Dir=c:\nwserver.old

Use Unsupported Drivers=FALSE

[NW! :LdapConfig]

Prompt=false

ClearText=false

[SMS]

prompt=false

[NWNMAS]

Prompt=false

[Beans]

prompt=false

[NSN]

prompt=false

[Perl]

prompt=false

[PHP]

prompt=false

[J2EE]

prompt=false

[AUDIT]

prompt=false

[MySQL]

prompt=false

rootPassword=root

[RF Generator]

Install Type=Upgrade Local

NetWare Source Location=True

Response File Name=response.rsp

Response File Local Path=c:\

Response File Path=sys:\oes

[Health Check]

Display Selection Screen=False

Display Summary Screen=False

Prompt=False

[NOVELL:NOVELL_ROOT:1.0.0]

silent=true

Prompt=FALSE
closeScreen=SilentCloseScreen
allowCloseScreen=true
Reboot=TRUE
[NWI:License]
Install Licenses Later=true
License File=sys:\oes
[NWI:Server Settings]
Load server at reboot=TRUE
[Installation Type]
OES Install=TRUE
Prompt=FALSE
[Initialization]
Description=
DescriptionID=
[NWI:MISC]
Source Server Name=LAB08
Source Location Path=sys:\oes
Prompt=false
Source Domain Address=137.65.87.62
Source Domain Address Type=UDP
Source Tree Name=LAB08-TREE
Source Server Context=Novell
Source User Name=admin
Source Context=novell
Relogin Password=novell
[NWI:NDS]
Tree Name=LAB08-TREE
New Tree=false
Server Context=Novell

Admin Login Name=admin

Admin Context=novell

Admin Password=novell

Prompt=false

NOTE: As mentioned at the start of this appendix, here are examples of what these two files should look like.

UPG2OES.NCF
load nwconfig.nlm transport=udp u="admin" ds c="admin.novell" p=novell z=LAB08-TREE a=137.65.87.62 e=sys:\oes\response.log b=sys:\oes\response.ips d=C:\NWSERVER s=LAB08/sys:\oes,3 n=LAB08/sys:\oes
RESPONSE.IPS example:
@Version 1.00 Command icmd GetPath sysVol,1,'SYS:'," CopyFile 0,0,0,2,0, NWSRC, product.ni, ", ", sysVol, ni/data,"",0 CopyFile 0,0,0,2,0, NWSRC, 'install/jar/product.jar', ", ", sysVol, "",",",0 EraseFile 0,sysVol,ni\\nis30\\lib\\nisw*.*,", " EraseFile 0,sysVol,ni\\nis30\\lib\\osnisw*.*,", " Display 2, "Initializing the installation..." Console 'nissets',20 Console 'jar -J-envCWD=\\ -xvf sys:\\product.jar ni\\bin\\runsetup.ncf ni\\bin\\NIDelay.nlm ni\\lib\\NISwitch.class ni\\lib\\NISwitch.ver',30 ; NOTE: The switch file will handle the remainder of the unjar. Display 2, "Running the installation..." Console 'ENVSET NISRC=%{NWSRC}\\', 1 Console 'sys:/ni/bin/runsetup /RF=sys:\\oes\\response.rsp', 1
NOTE: To ensure that the search and replace goes smoothly, verify that the last line of this file looks like this: "Console 'sys:/ni/bin/runsetup /RF="
In other words, remove the text after it.

APPENDIX B: TERMS USED IN THIS DOCUMENT

Component: The software package of ZENworks Server Management organizes its logic into components. The logic flows top to bottom and left to right. The main component contains variables, restrictions and information that relates to the whole, while each individual component contains variables and restrictions that only apply to itself.

Distribution: The entity (read group of files) that contains policies, files and the compiled package or applications to be sent to each subscriber.

Distributor: The server that makes and distributes content, software packages, applications and policies. The server pairs with corresponding object in eDirectory.

IPS SCRIPT: A file extension that designates the install script for NetWare servers.

Products.dat: A btrieve file that contains applications installed to a NetWare server.

Subscriber: The server that receives, extracts and applies the content, software package or application. The server pairs with corresponding object in eDirectory.

Variables: A word that contains data that can be changed based on the location or administrators preferences. These can be found in install scripts and software packages.

NOTE: Any references to Novell Open Enterprise Server discuss only traditional NetWare—never Linux.

APPENDIX C: CREDITS AND LEGAL ATTRIBUTION

Author

Martin Irwin Global Resolution Engineer, ZENworks, Worldwide Support Services, Novell, Inc.

Contributors and Reviewers

Lynn Bendixsen Configuration Manager, Java Virtual Machine Team, Novell, Inc.

Jason Douglas Software Engineer, Java Virtual Machine Team, Novell, Inc.

Paul Hardwick Software Engineer, Novell NDS/eDirectory, Critical Problem Resolution, Novell, Inc.

Troy Gordon Global Resolution Engineer, ZENworks, Worldwide Support services, Novell Inc.

Darren Nelson Global Resolution Engineer, ZENworks, Worldwide Support services, Novell Inc.

Holly Newman Senior Systems Engineer, Jackson Enterprises

Mark Schouls ZENworks Product Manager, Novell, Inc.

David Stagg Primary Support Engineer, Novell Technical Support, Novell, Inc.

*Windows is a registered trademark of Microsoft Corporation. Linux is a registered trademark of Linus Torvalds.

Palm is a registered trademark of Palm, Inc. RIM and BlackBerry are trademarks of Research in Motion, Ltd. Tivoli

is a registered trademark of the IBM corporation. HP OpenView is a registered trademark of Hewlett-Packard Development Company. All other third-party trademarks are the property of their respective owners.