



# NetWare 6.0 Web Infrastructure

NetWare Cool Solutions Article  
by Joe Harmon



## Table of Contents

- [Part 1: Installation](#)
- [Part 2: Deployment](#)
- [Part 3: Increasing the Amount of RAM that Tomcat Can Use](#)
- [Part 4: NetWare WebAccess Configuration](#)
- [Part 5: iFolder Administration](#)
- [Part 6: Apache](#)
- [Part 7: Apache Web Server Installation and Deployment Considerations](#)
- [Part 8: Tomcat](#)

# Part 1: Installation

---

## INTRODUCTION

- Basic Web Infrastructure products - These are usually individual products that perform a specific function.
  - Novell Portal Services
  - NetWare WebAccess
  - Apache
  - Tomcat
  - NetWare Enterprise Web Server
  - iManager
  - NetStorage
- Extended Web Infrastructure - Basic Web Infrastructure services working together to provide a solution.
  - File Access
  - Application Access
- The information in this article will be geared towards understanding the complexity of Web Infrastructure on NetWare 6.0. In essence, we will see how to integrate Basic Web Infrastructure (Web products) into Extended Web Infrastructure (solutions).
- This presentation is formatted to address the needs of novice and advanced users.

## AGENDA

- Installation
  - Pre-installation
    - Pre-requisites to installation
  - NetWare 6 product installation
    - NetWare 6 installation considerations
    - Installation of NetWare 6 products
  - Post-NetWare 6 product installation
    - Post-Installation considerations
    - Post-Installation of NetWare 6 products

# INSTALLATION

## Pre-installation

### Pre-requisites to installation

(Pre-requisite section was taken from <http://www.novell.com/documentation/lq/nw6p/index.html>)

- **Minimum System Requirements**
  - A server-class PC with a Pentium\* II or AMD\* K7 processor
  - 256 MB of RAM (512 MB if running NPS)
  - A Super VGA display adapter
  - A DOS partition of at least 200 MB and 200 MB available space
  - 2 GB of available disk space outside the DOS partition for volume SYS:
  - One network board
  - A CD drive
  - A USB, PS/2\*, or serial mouse (recommended but not required)
- **Recommended System Requirements**
  - A multiprocessor PC with a least two Pentium III 700 MHz or higher processors
  - 512 MB of RAM
  - A Super VGA or higher resolution display adapter
  - A DOS partition with 1 GB of available space
  - 4 GB of available disk space outside the DOS partition
  - One or more network boards
  - A bootable CD drive that supports the El Torito specification
  - A USB, PS/2, or serial mouse
- **Upgrade System Requirements**
  - The server to be upgraded must be running one of the following:
    - NetWare 5.1 with Support Pack 2 or later
    - NetWare 5 with Support Pack 6 or later
    - NetWare 4.2 with Support Pack 8 or later
    - NetWare 4.0 with Support Pack 8 or later
  - A server-class PC with a Pentium II or AMD K7 processor
  - 256 MB of RAM
  - A Super VGA display adapter

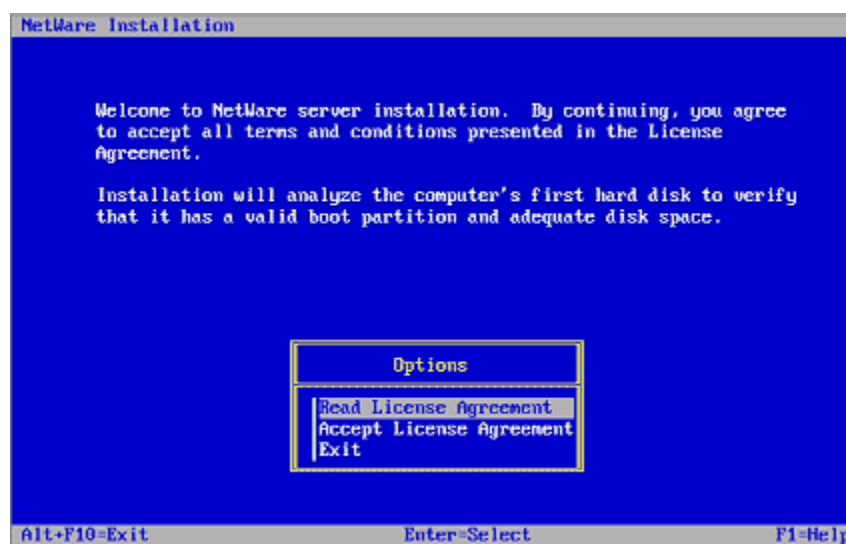
- A DOS partition with 35 MB of available space
  - 2 GB of available disk space on volume SYS:
  - One network board
  - A CD drive
  - A USB, PS/2, or serial mouse (recommended but not required)
- **Software and Other Requirements**
- *NetWare 6 Operating System CD*
  - *NetWare 6 License/Cryptography* diskette
  - Supervisor right at the [Root] of the eDirectory™ tree
  - Supervisor right to the container where the server will be installed
  - Read right to the Security container object for the eDirectory tree
  - DOS and CD drivers (required if the computer does not boot from CD) You can make a bootable floppy diskette using the MKFLOPPY.BAT program located in the INSTALL directory of the *NetWare 6 Operating System CD*.
  - Client connection utilities (optional, for installing from a network):
  - Novell? Client™ for DOS and Windows\* 3.1x (optional, for installing from a NetWare server running IPX™).
  - IP Server Connection Utility (optional, for installing from a NetWare server running IP only). For instructions, see PRODUCTS\SERVERINST\IPCONN.TXT on the *Novell Client CD*.
  - IP address and domain names (required for connecting to the Internet):
  - An IP address
  - An IP address of a domain name server
  - The name of your domain
  - Network board and storage device properties, such as the interrupt and port address (required if not included in NetWare)
- **Prepare the Network**
- Run NetWare Deployment Manager (NWDEPLOY.EXE), located on the *NetWare 6 Operating System CD*.
  - Complete all relevant tasks in the Prepare the Network section.

## NetWare 6 product installation

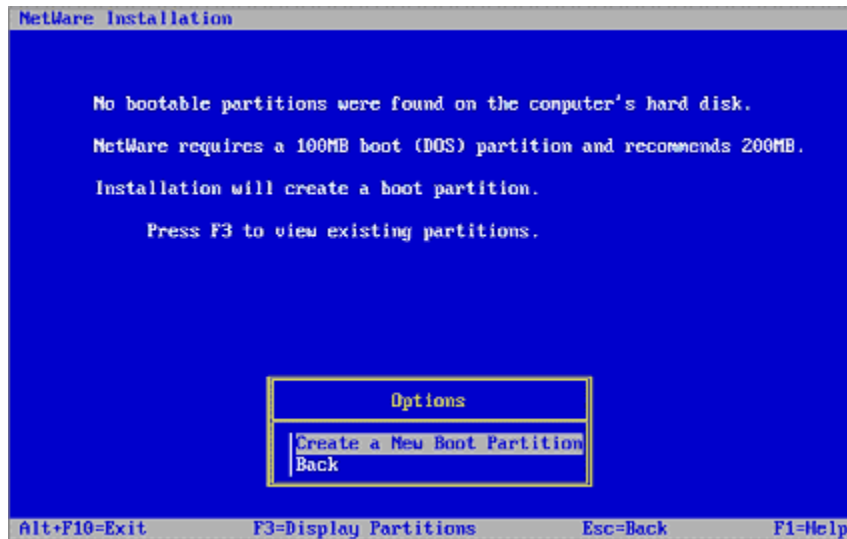
### NetWare 6 Installation Considerations

- Is SSL up and running on my server, and is my Certificate Authority functioning properly?
- Have I considered the replica ring in which server (containing eDirectory) is going to reside? There are issues that will be discussed throughout the manual on this subject.
- Am I inserting this into an existing tree, performing an upgrade, or a migration? If so, have I properly prepared my server for the installation?
- Have I decided what services I want to run on this server and taken into consideration any performance or service conflicting issues?
- Am I going to use a single IP address or a multiple IP addresses?
- What ports are going to be used for which service?
- Who are these services going to be made available to?
- Where are these services going to be made available? (intranet, extranet, or both)

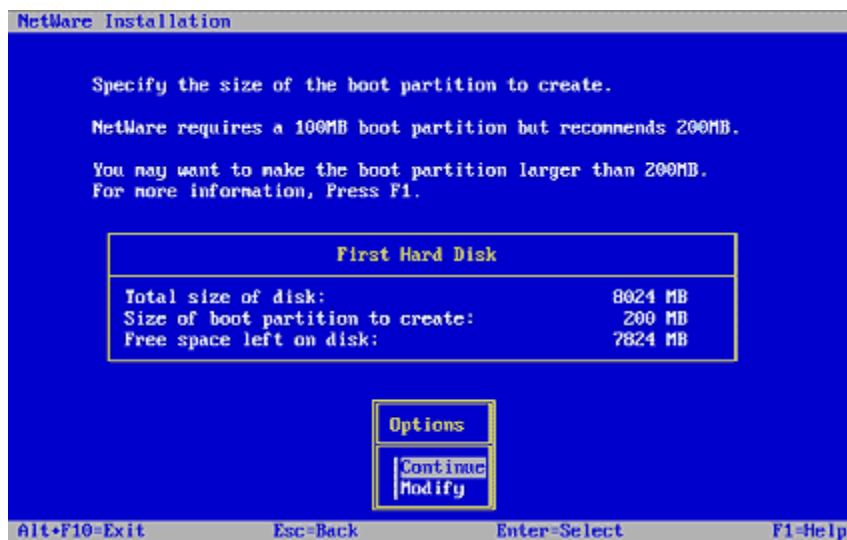
### Installation of NetWare 6 Products



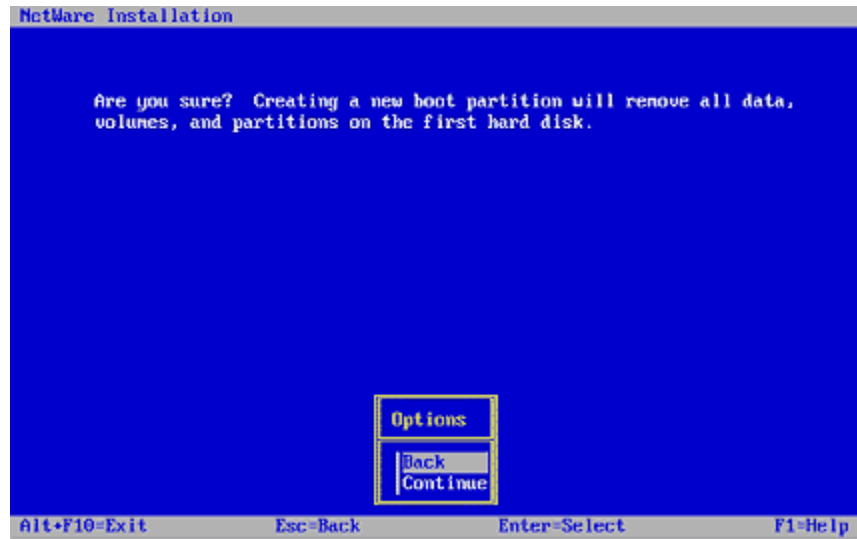
(1) Choose ACCEPT LICENSE AGREEMENT and hit [ENTER] to continue.



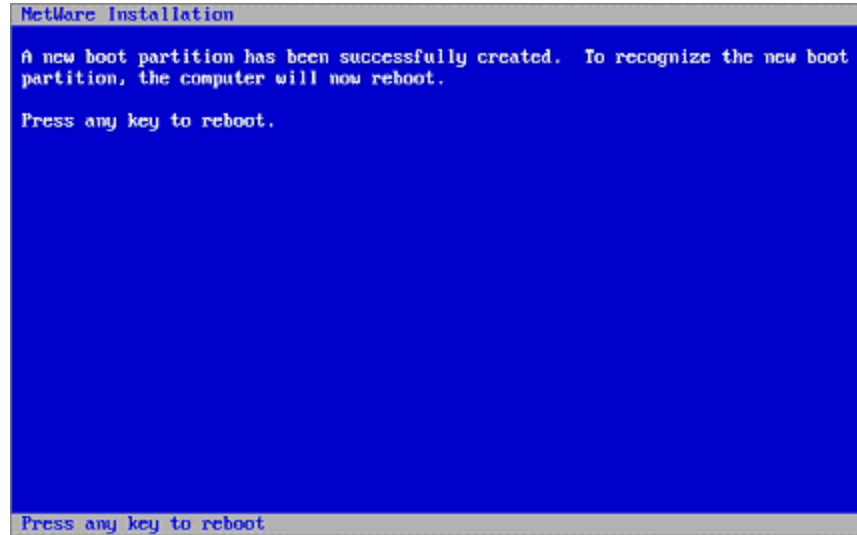
(2) In this instance we want to create everything fresh with this NetWare 6 installation. Therefore, choose the option to CREATE A NEW BOOT PARTITION and hit [ENTER] to continue.



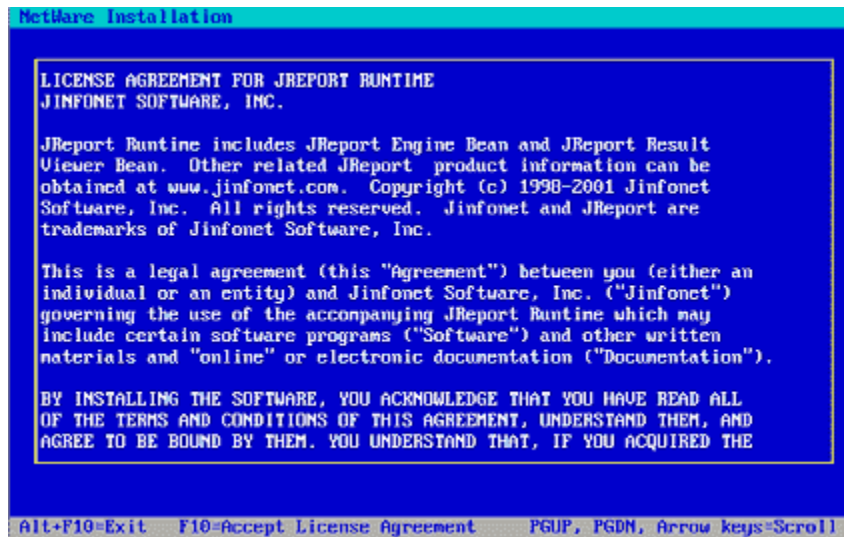
(3) The size of the boot partition can be modified if desired. I have heard it suggested that you have twice the disk space as you do RAM on your server. However, I have found that if you need to take a core dump of your server your disk space can be as little as half the amount of RAM on your system so long as you take the core dump without file cache. Select CONTINUE and hit [ENTER] to move on.



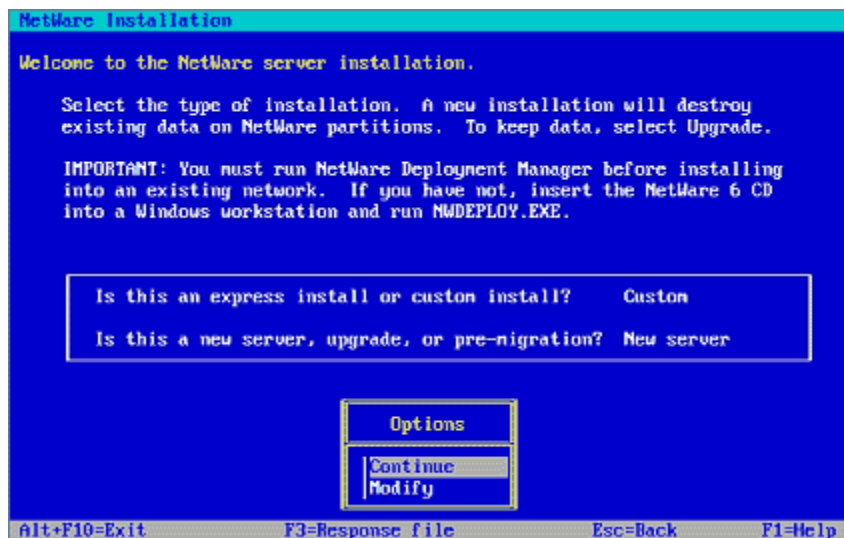
(4) Choose CONTINUE and hit [ENTER].



(5) After the new boot partition has been created, press any key to reboot the server.

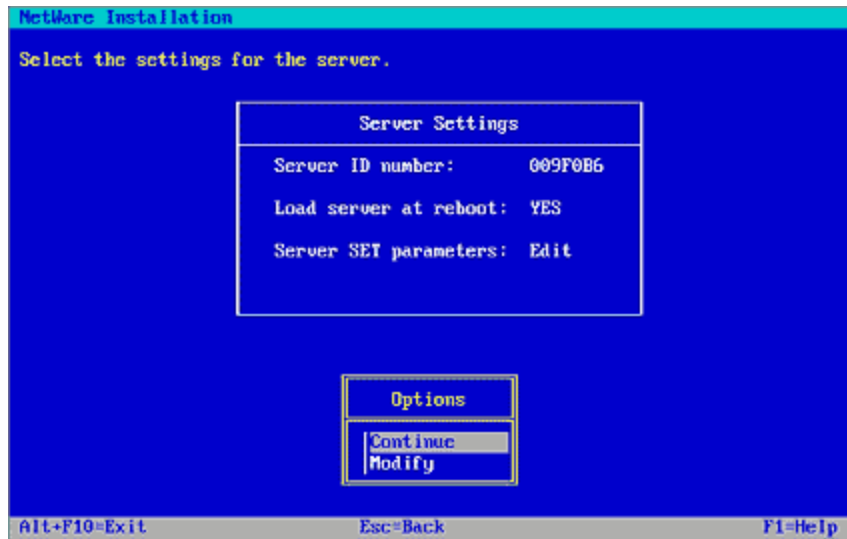


(6) Now we are ready to start the installation. Hit [F10] to accept the license agreement.



(7) We want to modify this screen. EXPRESS is the type of install that comes up by default. We want to change this option to CUSTOM so we can select the basic web services to be installed. We will leave the other option at NEW SERVER. Choose CONTINUE and hit [ENTER].

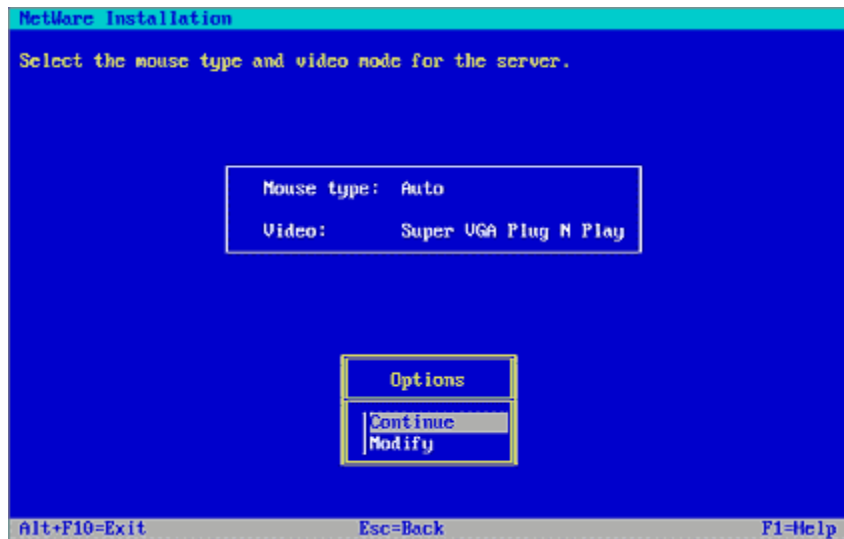
**\*\* NOTE \*\*** If choosing the option of UPGRADE or PRE-MIGRATION, you will need to take other factors into consideration. For example, who is the Certificate Authority (CA) of the Tree and is the CA functional? Does SSL currently work on this server? NetWare 5.1 had very few products that relied on SSL, so you may or may not know whether it is working properly. Even if this is a fresh install, if this is not the first server in the tree then you will want to verify that the CA is functional. NetWare 6 Web Infrastructure relies heavily on SSL, so if it is not functional, neither will be most of your Web Infrastructure after the install. Some other considerations should be content and whether or not the IP address will change. These can all have dramatic effects on the server installation and the functionality of the services after the installation.



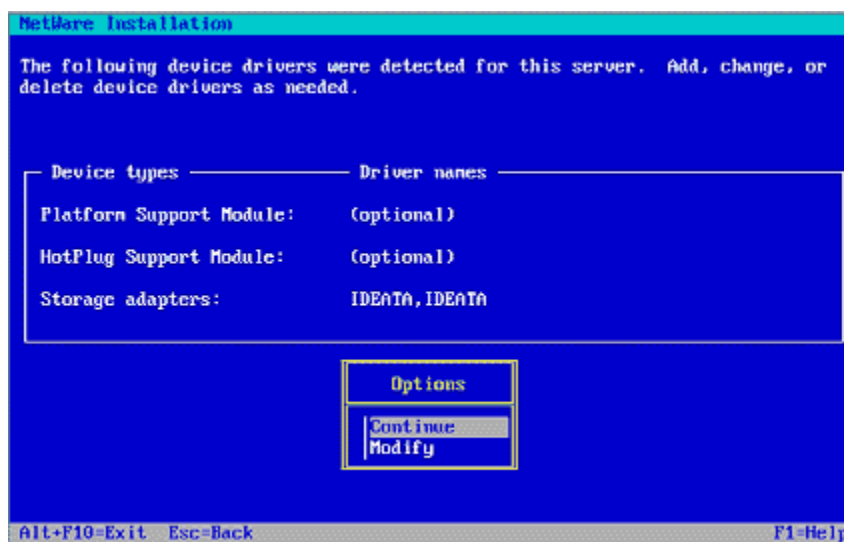
(8) Accept the defaults, select CONTINUE, and hit [ENTER].



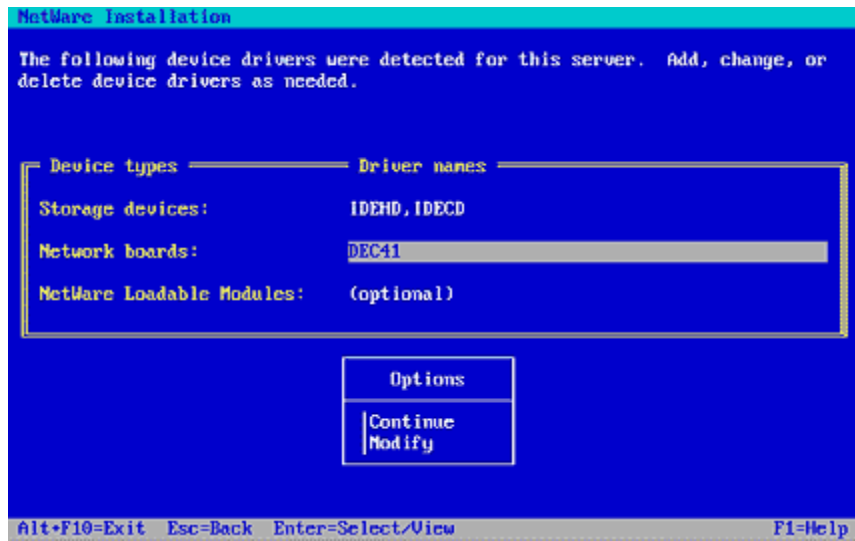
(9) Although it is not going to be covered in this training, it is important to note that if a language other than English is used, other products (such as NPS) will have a separate language configuration. For our purposes select CONTINUE and hit [ENTER].



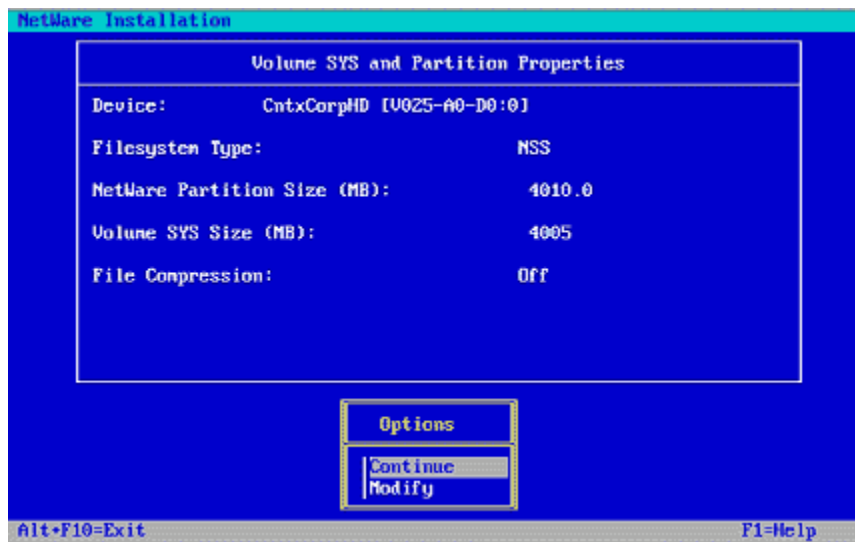
(10) Accept the defaults, select CONTINUE, and hit [ENTER].



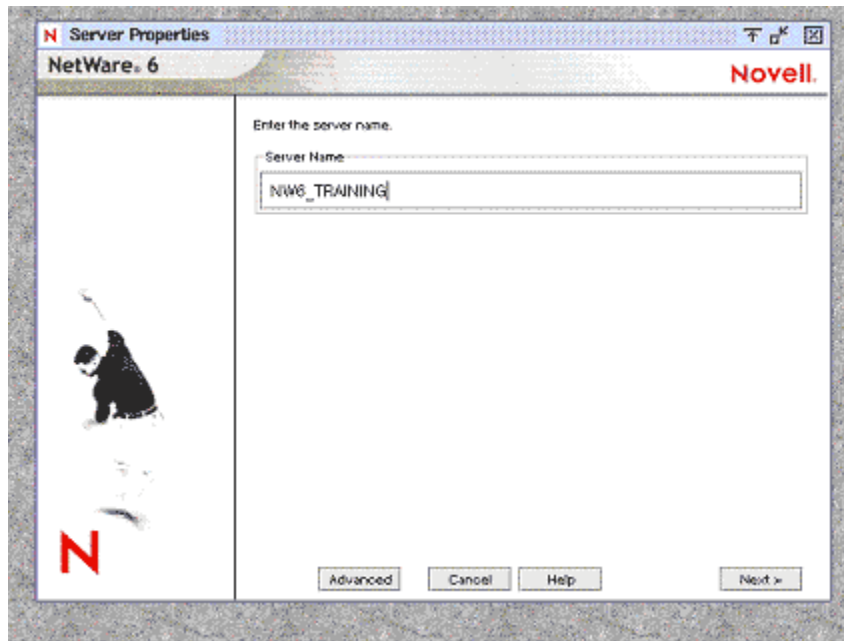
(11) Accept the defaults and hit [ENTER] to continue.



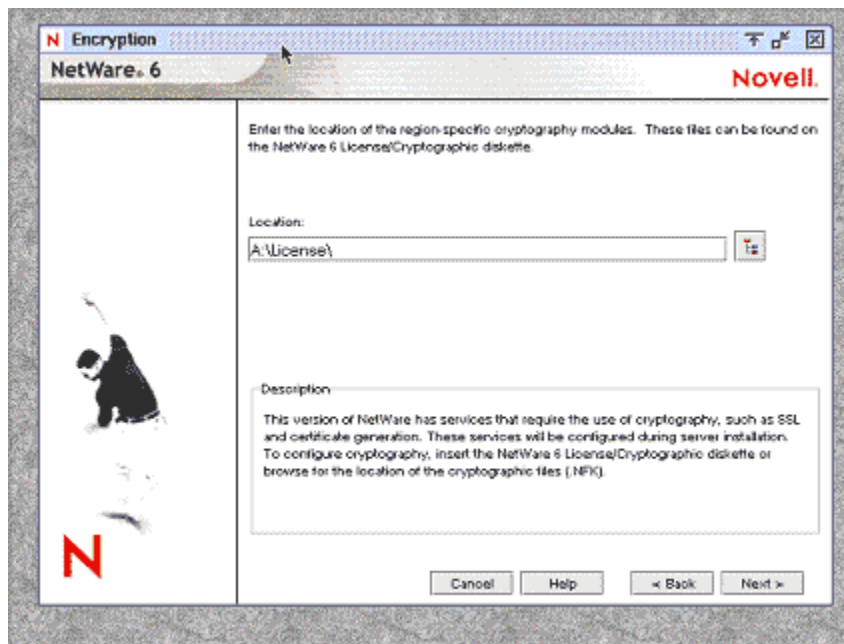
(12) Select the proper Network board and hit [ENTER] to continue.



(13) Create the desired size of the SYS volume and then hit [ENTER] to continue.

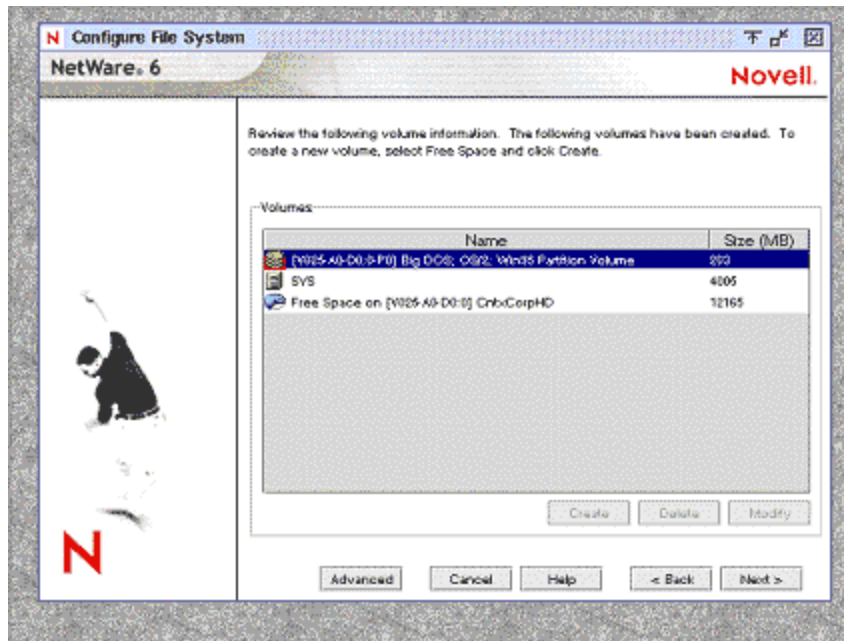


(14) Now enter in the name that you want for your server. The name of the server can either match the host name that you will be using or it can be different. For example, let's say that our server and host name are WEB and our domain is NOVELL.COM -- then our full DNS name would be WEB.NOVELL.COM and our server name would be WEB. The only advantage to this is that it makes it easier to remember which full DNS name belongs to which server. It is not required and will not be done with this install, but is mentioned as a common practice.

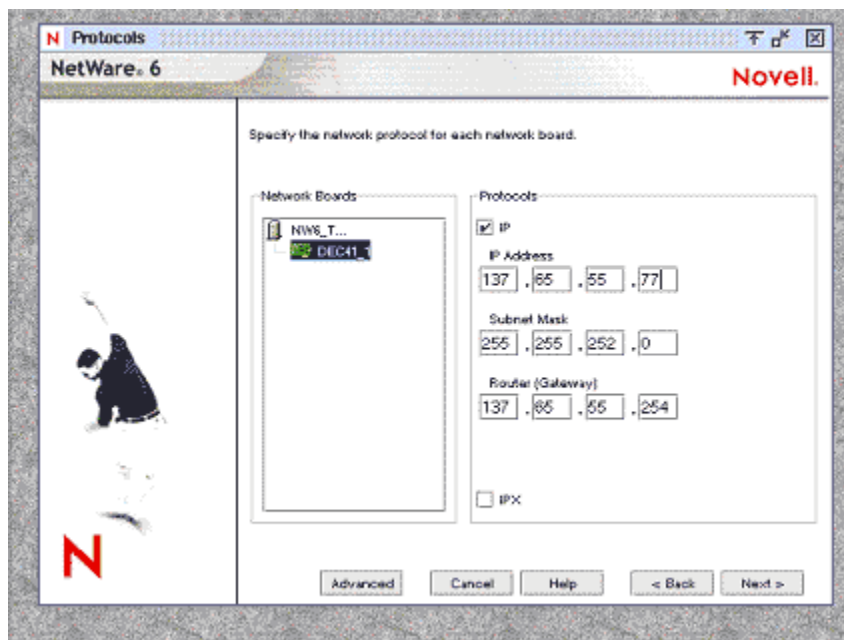


(15) Insert the license and select [NEXT] to continue.

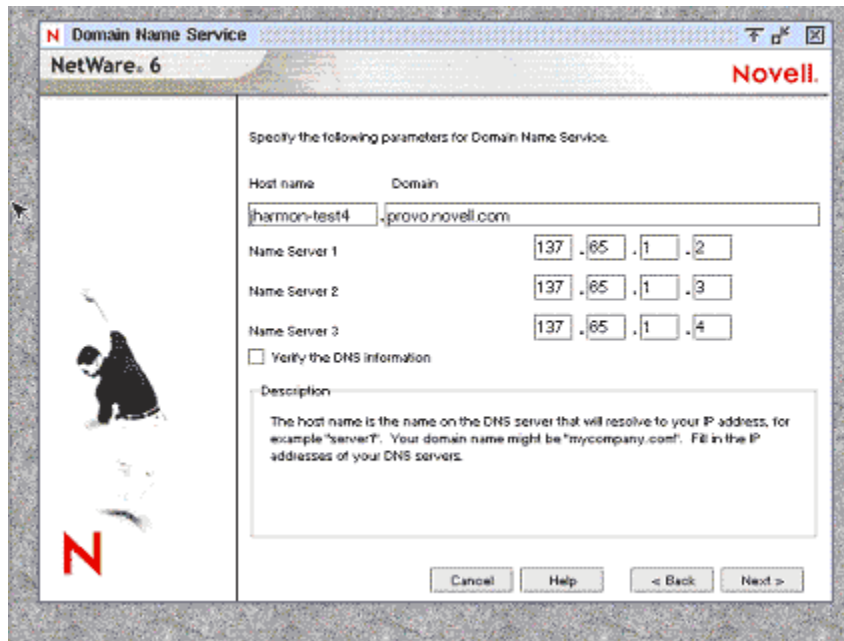
**\*\* NOTE \*\*** The only real considerations with licenses is to know that if your licenses are limited and you go over that limit then web services that require a licensed connection can fail. Or if you are installing a demo license that will expire, and it does expire, then web services will fail to load.



(16) There are several services that can grow in size which may fill up your SYS volume. You may want to create another volume that will allow for this growth. Some of these services may include iFolder (file storage and log files), Apache (web site and log files), NES (web site and log files), NetStorage (file storage), etc. If you desire to create another volume, select the FREE SPACE section and select CREATE. Once the volume has been created, select [NEXT] to continue.

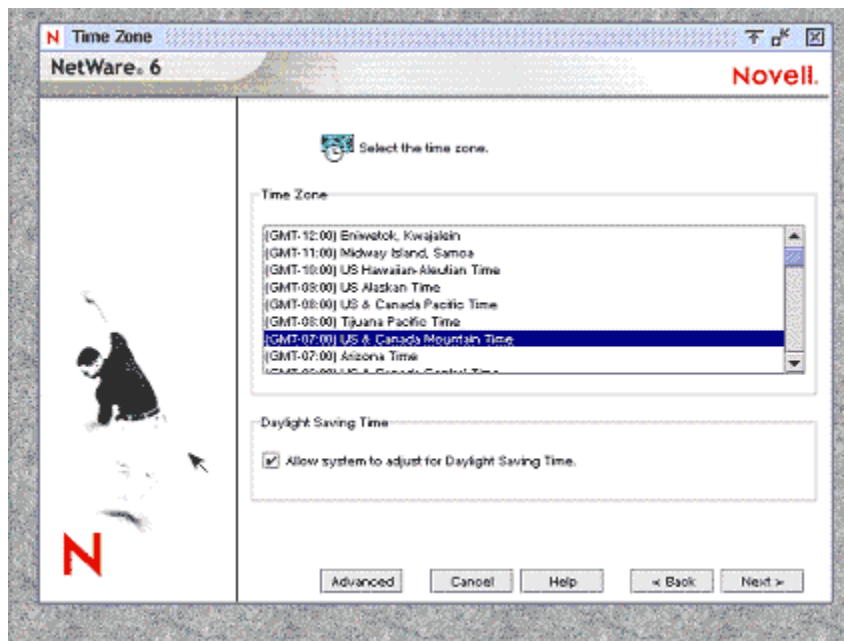


(17) I want you to pay close attention to the IP address that I am binding to my NIC. This will have significant impact later on during the install. Just remember that it ends with a 77.

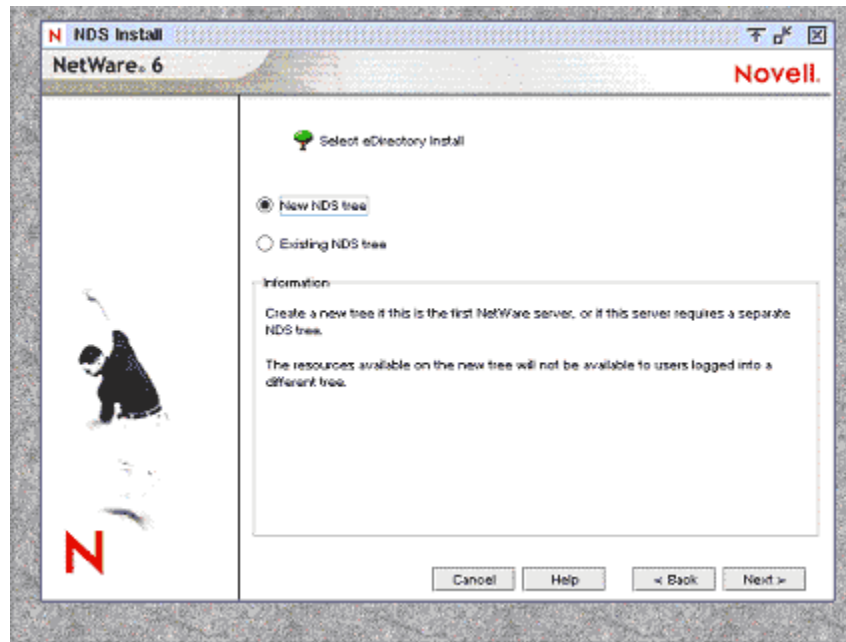


(18) Remember in step 13 that we discussed the option of having the server name and the host name the same. Well to show the point that they don't have to be the same (and because my IP address is already registered with a DNS name that doesn't match the server), my server name will be NW6\_TRAINING, my host name will be JHARMON-TEST4, my domain will be PROVO.NOVELL.COM, and my full DNS name will be JHARMON-TEST4.PROVO.NOVELL.COM.

**\*\* STOP! \*\*** If you are putting in a non-registered DNS name or non-existent DNS servers, then you may adversely affect any service that is configured with that DNS information. In other words, if your DNS does not resolve and your configuration files are configured with a non-resolvable DNS information, then your web services WILL NOT WORK without significant modifications. So now you might ask, what if I am in a test environment and do not have DNS setup? Is there a way to configure the services to only listen on the IP address? The answer to this is YES, but you will need to understand that if this is moved to production some reconfiguration will be needed for services that require reverse proxy. If DNS is not available then you can skip past this screen with the understanding that you may affect services that require DNS resolution.



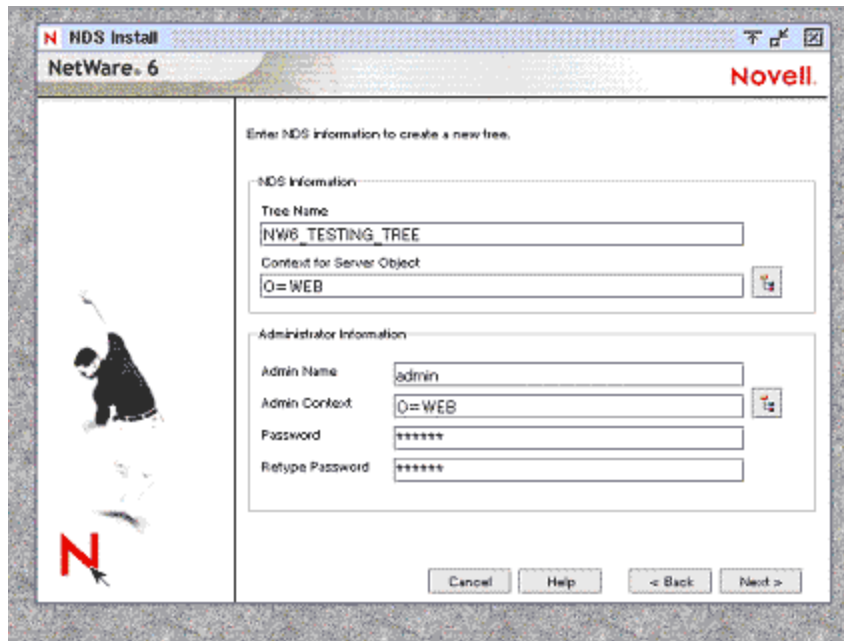
(19) The most important thing to note here is that NetStorage can be adversely affected if time is not setup properly. I had a server that was two hours behind on its time and a workstation that was unable to access NetStorage via IE (but could through Netscape) because the time on the server was behind the time on the workstation. This was the only time that I ever saw this issue, but I did want to mention it for reference sake.



(20) For this training we will select the option to create a NEW NDS TREE. Be aware that if you are installing NetWare 6 into an existing tree, certain things will need to be prepared before that happens. We already spoke about the issues with SSL, but there are also issues with SCHEMA. It is not the design of this training to go into these issues separately but rather to make you aware of potential downfalls during the install.

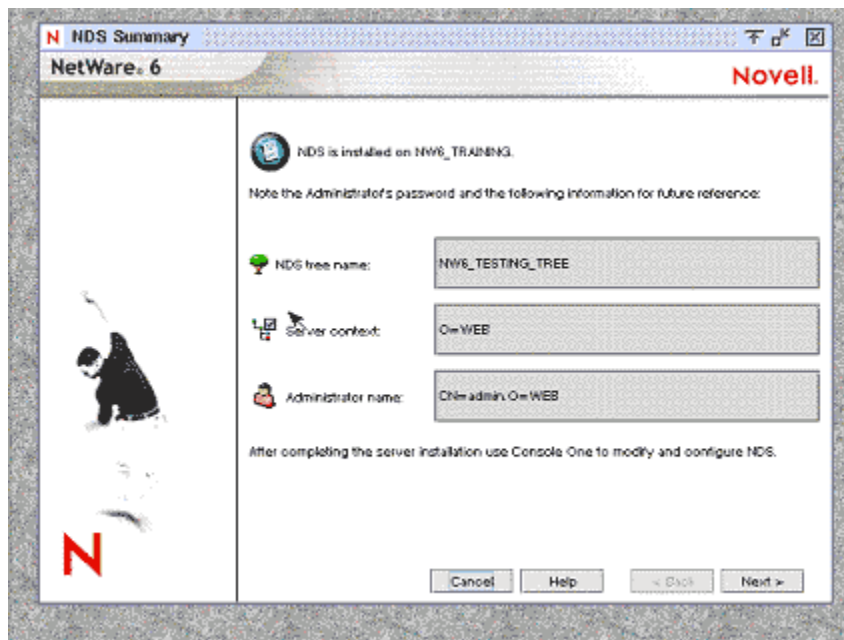
**\*\* IMPORTANT \*\*** Products that require eDirectory on NetWare 6 can have a problem installing schema if a version of DS 7.x or lower is contacted during the extension of the schema. The reason this could be a problem is that some NetWare 6.0 products have AUX classes. These classes are not understood by versions of DS 7.x or lower. The best solution for this situation is to have eDirectory in its own partition or in the same replica ring as other eDirectory servers. If you have to have a mixed replica ring then be sure that the master replica is held by eDirectory. OnDemand and Novell Portal Services are two products that have seen this problem.

**\*\* NOTE \*\*** DS versions 8.73 and 8.77 that can be installed with NetWare 5.1 are NOT eDirectory. This is a common misconception. eDirectory versions are 85.12a, 86.2, etc.

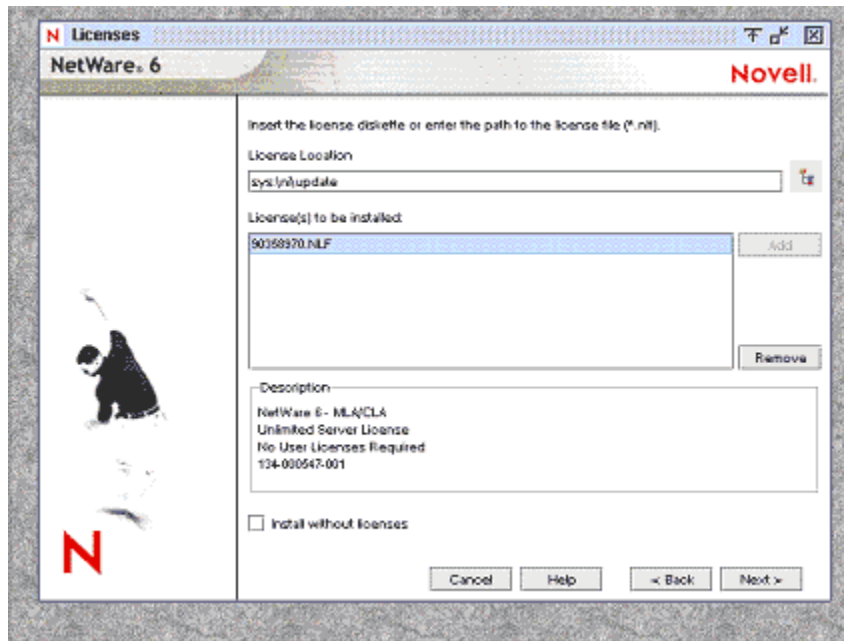


(21) Now we will need to put in the name of the tree and the top O as well as the admin's password.

**\*\* IMPORTANT \*\*** There are certain services that require access to the root of the tree. Two of these services are iManager and NetWare WebAccess. If you are installing as a container admin and do not have rights to the root of the tree, then the installation of these products can fail.

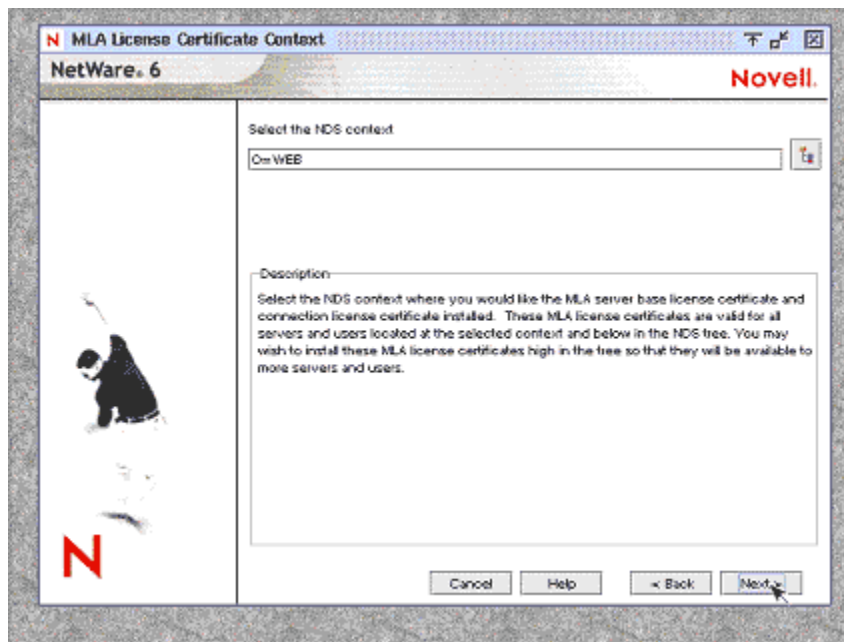


(22) This next screen is just verifying your information. Select [NEXT] to continue.



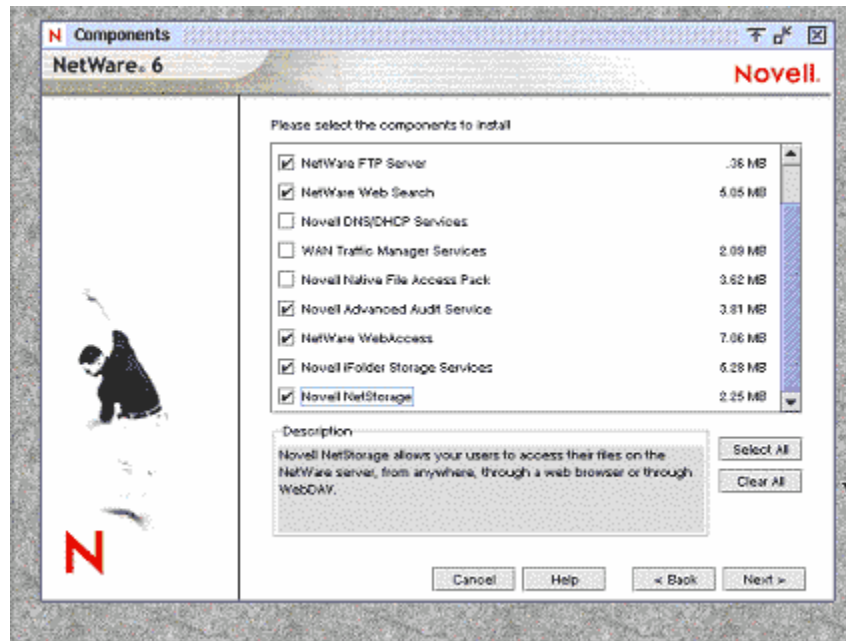
(23) Select the license and click [NEXT] to continue.

\*\* NOTE \*\* In step 15 we discussed the issue with installing without a license. Please refer to that step if you have any questions about the INSTALL WITHOUT LICENSES option that is located at the bottom of this screen.



(24) Here we are just asking where you want to install the license. For this training, accept the default and click [NEXT] to continue.

\*\* NOTE \*\* If your licenses are limited, be aware that ROLE BASED SERVICES (used for iManager) require a licensed connection. The licenses need to be placed properly in order for you to access iManager.



(25) The only services which are selected but can't be seen are iPrint and the NetWare Enterprise Web Server (NES). I want to stress selecting these products because of the impact they can have on Web Infrastructure. Novell Advance Audit Services was left selected by mistake in this screen shot. Although it will have no affect on the products being installed, you don't need to select it for this training. Be sure the following products are selected then click [NEXT] to continue.

- NetWare Enterprise Server
- iPrint
- NetWare FTP Server
- NetWare Web Search
- NetWare WebAccess
- Novell iFolder Storage Services
- Novell NetStorage

**\*\* IMPORTANT \*\*** There are a couple of services that do not show up during the original install, but they will be installed. One is Novell Certificate Server (for SSL), and the other is Web Manager (installs Tomcat and Apache).

(26) Single IP Address VS. Multiple IP Address.

**\*\* NOTE \*\*** NetWare 6 has the ability to use Multiple IP Addresses or a Single IP Address to load handle its web servers. The reason for this is simple. Both Apache and the Enterprise Server use the same ports (80 and 443). This brings us to the reason for the choice between using a single IP address or multiple IP addresses.

If you have a single IP address you will need to assign ports other than 80 and 443 to one of the services (if both are being installed). The issue here is that port 80 represents HTTP and port 443 represents HTTPS. So long as you are specifying HTTP or HTTPS there is no need to place the port number at the end of the URL. The ports are assumed. If you specify the single IP address option you will need to specify the port at the end of the URL. Example: http://192.168.0.1:1000. If you specify the multiple IP address option, you can assign port 80 and 443 to another IP address, thus eliminating the need to place the port at the end of the URL.

**\*\*IMPORTANT\*\*** You can allow the same IP address to listen on multiple ports. You can allow the same port to listen on multiple IP addresses. You CANNOT allow the same IP address to listen on the same port multiple times.

Since there are several services within NetWare 6 that want to use the same ports, you are given the option to have one IP address with multiple ports or multiple IP addresses with the same ports. The following services on NetWare 6 try to use port 80 or 443.

- NetWare Enterprise Server
- Apache
- iFolder
- iPrint

**\*\* IMPORTANT \*\*** If you choose the option for SINGLE IP ADDRESS, you will need to decide which ports you want to use. Keep in mind that the ports may already be taken by another product. For a list of common ports, see [NetWare 6 Port Assignments](#). By default, if the NetWare Enterprise Web Server is installed, it will take over port 80 and 443. If desired you can change this over to the Apache-based Services. However, one thing to keep in mind is that you will have to configure Apache through a configuration file vs. the Enterprise Server being configured through a GUI interface.

**\*\* IMPORTANT \*\*** Whichever service owns port 80 and 443 will also receive the NetWare 6 home page. If assigned to the Enterprise Server, the NetWare 6 home page will be copied out to the SYS:/NOVONYX/SUITESPOT/DOCS directory. If assigned to Apache-based Services then the NetWare 6 home page will be copied out to the SYS:/APACHE/NWDOCS directory. If you desire to keep the NetWare 6 home page and host your own home page on port 80 and 443, then it is suggested that you choose the multiple IP address option. You can still choose the single IP address option and host both the NetWare 6 page and your own home page, but it will require additional configuration after the installation.

**\*\* IMPORTANT \*\*** Be sure that you engrave this next point into your mind. There is a section for the IP address, DNS name, and ports to be used for each service. During the install the section for DNS name is used to configure most of the web services configuration files. This is why the services will fail if the DNS name does not resolve. I spoke about this in step 18. If you are running a test box and do not want to worry about DNS resolution, then you can place the IP address in the host field as well as in the IP field. This will then configure your files with the IP address and not the DNS name. The only time where you will run into a problem is if you need to setup reverse proxy for outside access. Reverse proxy will need information within certain products to be the DNS name and not the IP address. But for testing purposes this should be fine.

Configure IP-based Services

You have selected to install multiple IP-based services that are typically installed on separate servers. You can use a single IP address for these services if you modify the default port numbers, or you can retain the default port numbers and specify multiple IP addresses. Please click Help for additional information.

Single IP Address  
 Multiple IP Addresses

NetWare Enterprise Web Server  
137 . 65 . 55 . 77 | jhamon-test4.prove.novell.com | 80 | 443

Apache-based Services  
137 . 65 . 55 . 77 | jhamon-test4.prove.novell.com | 51080 | 51443

Novell iFolder Storage Services  
137 . 65 . 55 . 77 | jhamon-test4.prove.novell.com | 52080 | 52443

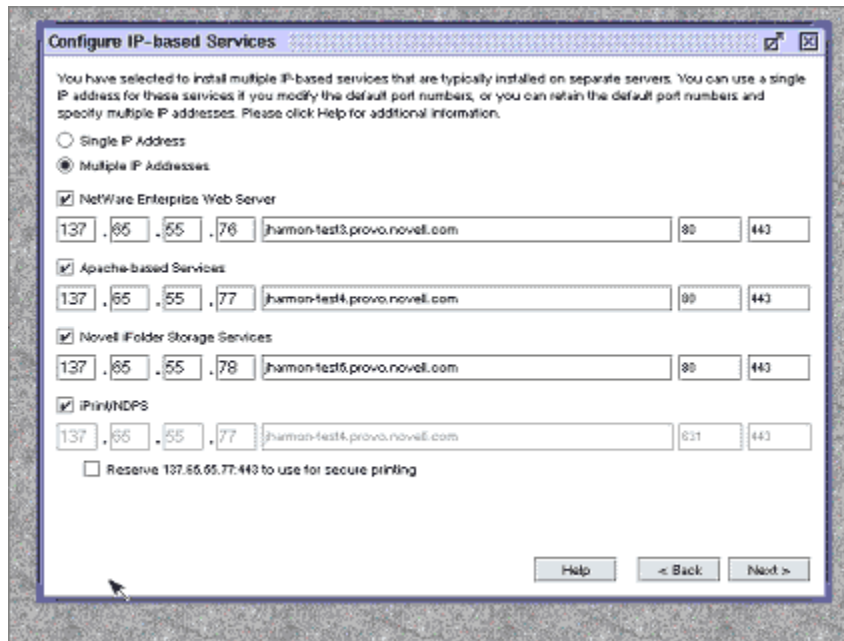
iPrint/NDPS  
137 . 65 . 55 . 77 | jhamon-test4.prove.novell.com | 631 | 443

Reserve 137.65.55.77:443 to use for secure printing

Help < Back Next >

(26a) Single IP address option

**\*\* NOTE \*\*** In this instance port 80 and 443 are defaulting to the Enterprise Server. If left this way the NetWare 6 home page will be given to the Enterprise Server. If this is an upgrade then your INDEX.HTML file will be renamed. If you had an existing Enterprise Server installed, it is suggested that you give port 80 and 443 to Apache-based Services. That way Apache will be hosting the NetWare 6 home page. The last thing I want you to note here is the secure port for iPrint. iPrint will take 443 for its secure port. This is required by the RFC. If you load a Web Service and iPrint on the same box you will need to change port 443 for the web service. iPrint is grayed out and does not allow you to change the port.



### (26b) Multiple IP address option

**\*\* REMEMBER \*\*** Back in step 17 you were asked to remember what IP address was being given to the server. I gave the server 137.65.55.77 for its IP address. By default the main IP address will be given to the first one on the list. If the Enterprise Server is installed it will default to that service. In this instance I will change that to the Apache-based Services. This is not necessary, but will cause a lot less confusion. Remember that if you already have the Enterprise Server installed and this is an upgrade, the NetWare 6 home page (which is an INDEX.HTML) will rename any existing INDEX.HTML that exists. This is another reason to give the main IP address to Apache. See a more detailed explanation below.

If you choose the multiple IP address option, there are a few things you should note to avoid confusion. The main IP address that you give to the server will be picked up by the NetWare Enterprise Web Server by default. The main IP address will also host the NetWare 6 home page.

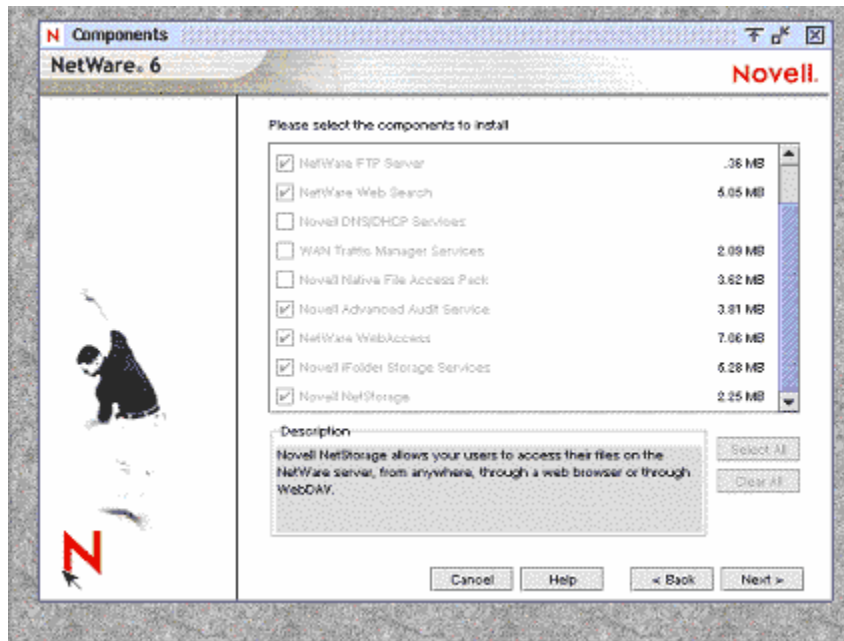
**Option 1** - If the main server IP address is given to the NETWARE ENTERPRISE WEB SERVER, the following will occur:

- (1) The NetWare 6 home page will be run by the NetWare Enterprise Web Server under the SYS:/NOVONYX/SUITESPOT/DOCS directory.
- (2) Apache Services will be available on the secondary IP address. However, the Web Manager (which runs through Apache) will be listening on the main IP address (the same one the Enterprise Server is using) on port 2200.
- (3) Apache will be available to run your home page under the SYS:/APACHE/NWDOCS directory since it is not using the NetWare 6 home page. However, if you plan on using the NetWare Enterprise Server to host your pages you will lose the NetWare 6.0 home page.

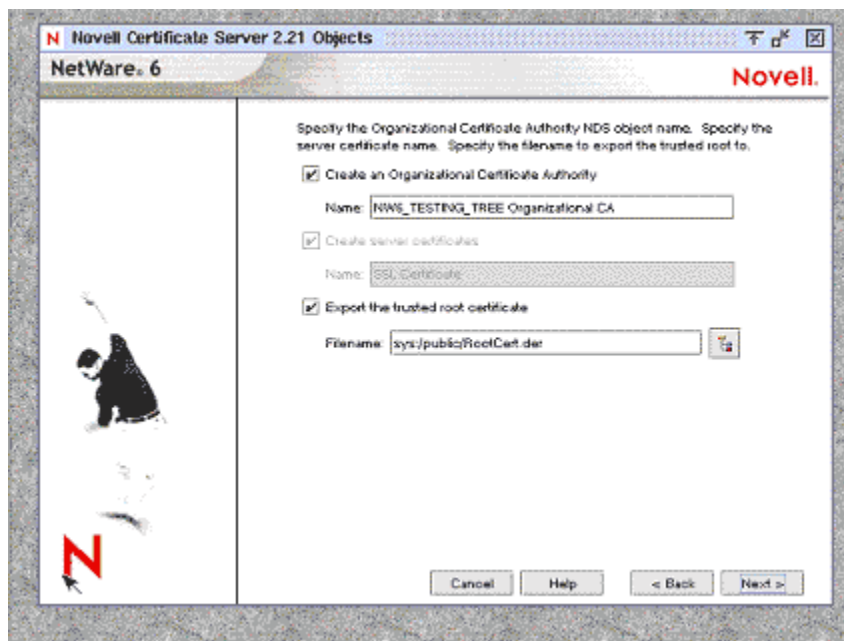
**Option 2** - If the main server IP address is given to APACHE-BASED SERVICES, the following will occur:

- (1) The NetWare 6 home page will be run by Apache Services under the SYS:/APACHE/NWDOCS directory.
- (2) Apache Services will be available on the main IP address along with the Web Manager running on port 2200 on that IP address.
- (3) Apache will not be able to run your home page by default because it will be hosting the NetWare 6 page. The Enterprise Server will be available to host your home page at this point under the SYS:/NOVONYX/SUITESPOT/DOCS directory.

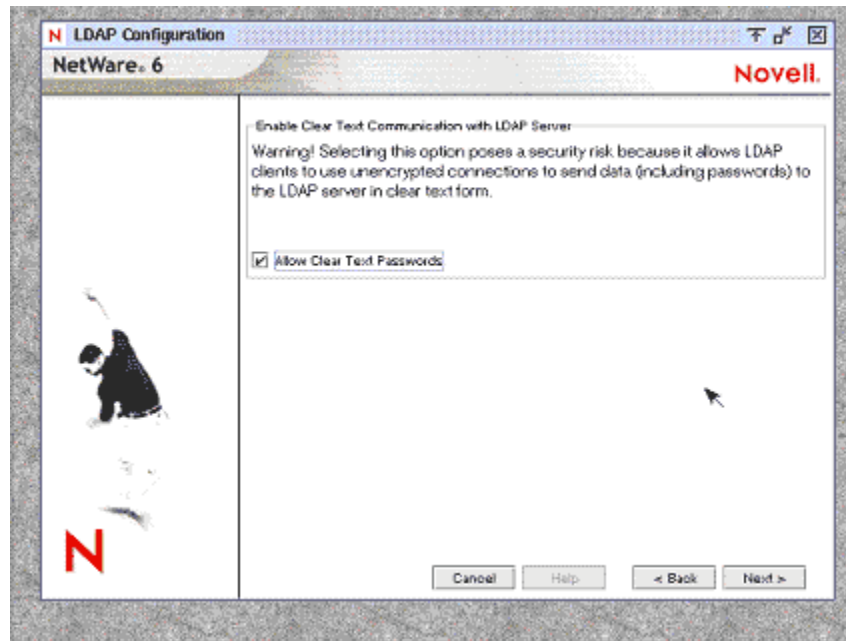
Therefore, if you want Apache to host your pages on port 80 and 443, give the Enterprise Server the main server IP address so that Apache will be open on port 80 and 443 (Option 1). If you want the Enterprise Server to host your pages on port 80 and 443, give Apache-Based Services the main Server IP address (Option 2).



(27) This next screen just shows the components that will be installed. Click [NEXT] to continue.

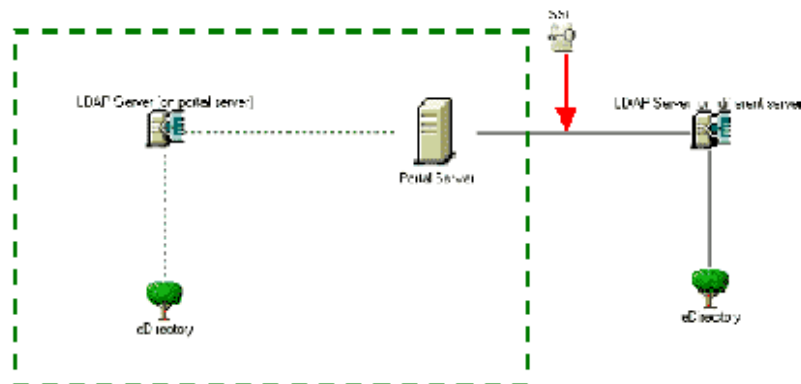


(28) In this situation you will notice that we are creating a CA. The reason for this is because this is the first server into the tree. If the CA is properly installed then you won't have many of the issues that can occur with SSL. If the CA portion of the screen is grayed out then you already have a CA. If you already have one then make sure that it is functioning properly before you continue. For our training installation, this will be the only server in the tree so the CA should be fine.

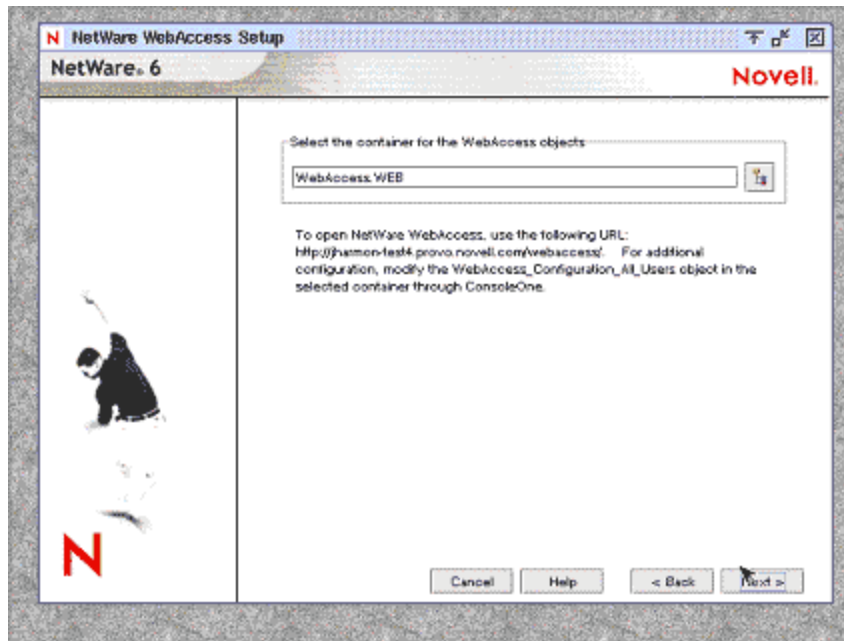


(29) This section has caused much confusion. The warning on this screen explains that if you select this option it will create a security risk. Well, that is true to a point and we will discuss this in much more detail under the deployment section, but let's quickly discuss the advantages, disadvantages, and when we should and should not deploy this option. To do this we will look at an example of a product that uses LDAP for its communication.

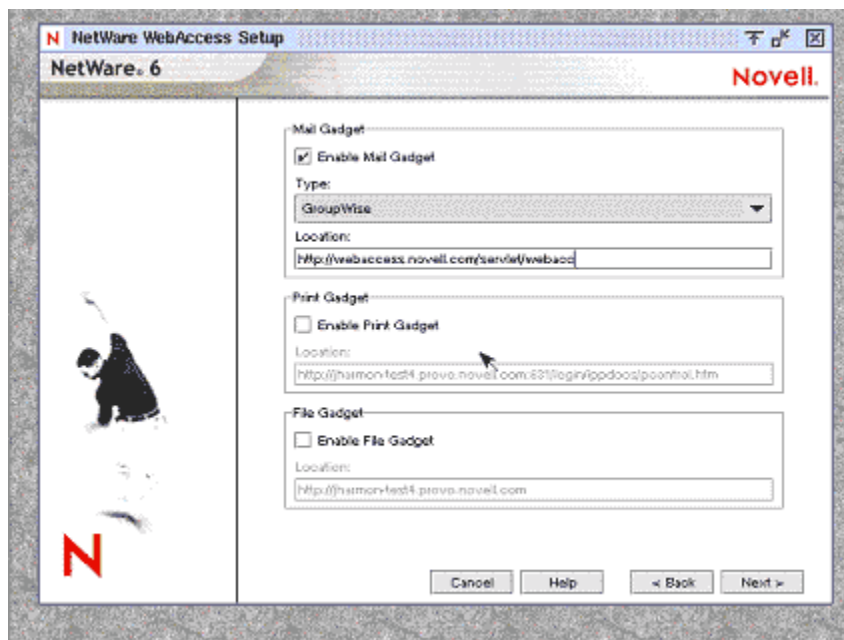
Novell Portal Services (NPS) uses LDAP for communication between the Portal Server and the LDAP server. Therefore, depending on your deployment, your LDAP and Portal Server may or may not be on the same box. Remember we are talking about the communication between the LDAP server and the Portal Server. If they are on the same server then there is no reason for encryption. The communication all occurs on the same server. Remember the reason that we have encryption in the first place is to mask the communication so that if it is intercepted it can't be read. However, since encryption is a process of converting or hashing the information being sent, and then undoing that process, you take a performance hit. So the advantage to SSL is the encryption of information and the disadvantage is slower performance than clear text (non-SSL). The advantage to clear text is better performance and the disadvantage is that it is not encrypted.



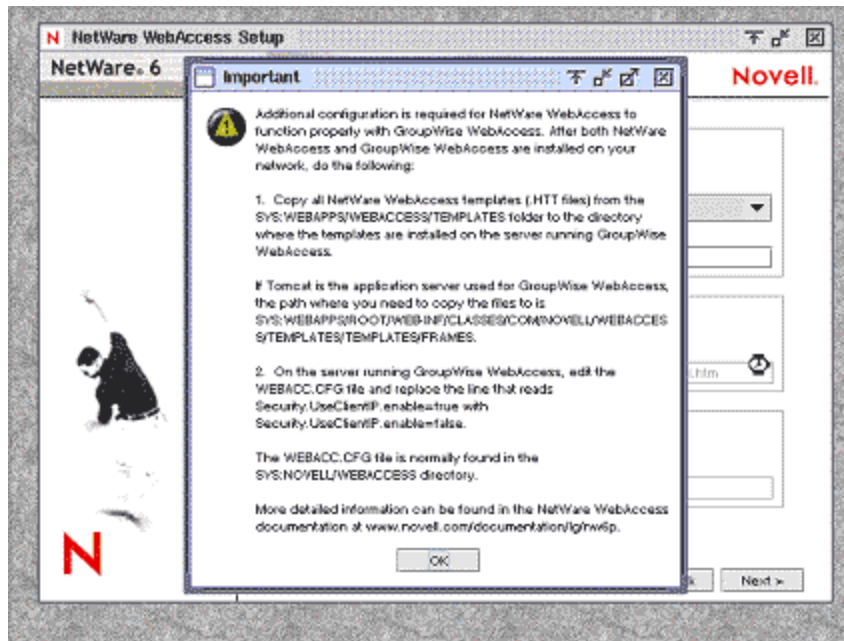
However, there is a place for both. The green box in the above illustration represents the same server. If LDAP and Portal are on the same server then there is no reason to encrypt the data. Why? Well who is going to be able to pick up that information? The server is not taking over the network, it is talking to itself. If LDAP is on a different server then you may want to implement SSL.



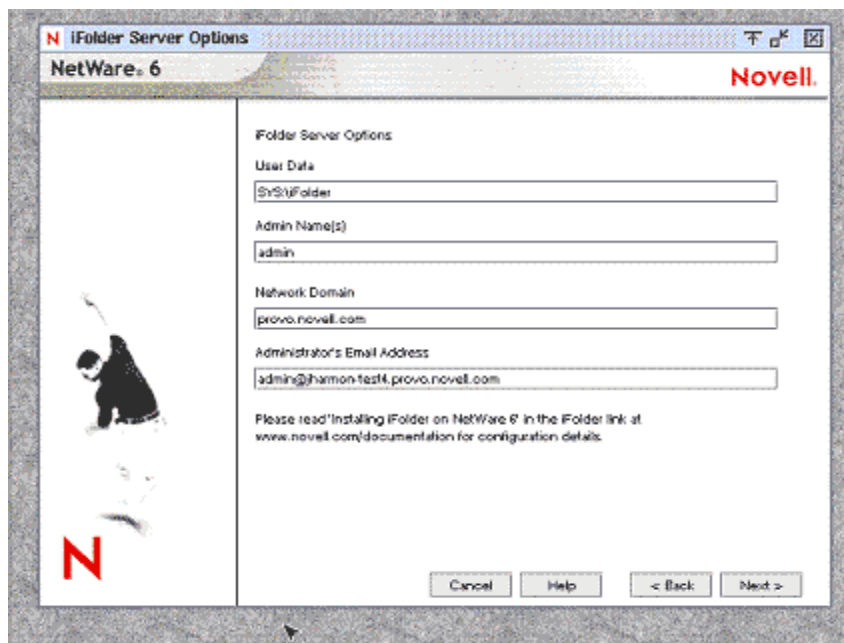
(30) First thing to note is that this is NOT GroupWise WebAccess. NetWare WebAccess was built off of Novell Portal Services. It ships with a few basic gadgets that provide services such as file access, email, printing, address book, etc. To create the objects needed to configure and run this service, we will need to decide where the WebAccess container will reside. You cannot change the name of the container. It will remain WebAccess.



(31) When you choose the option of multiple IP address vs. single IP address, make sure you know which service (Apache or Enterprise) will be running the NetWare 6 home page. NetWare WebAccess will reference several files that are contained within that same location. If the default configuration is kept and the NetWare 6 home page information is hosted by NES, NetWare WebAccess will require configuration after the installation is complete. The reason for this is that the NetWare WebAccess configuration will point to Apache by default.



(32) A very common issue is related to the screen shown above. What this is basically saying is that there are some template files that are required in order for NetWare WebAccess to be able to pull in the information from GroupWise WebAccess. If the files are not copied out to the proper location, or if they are not copied at all, then the users will receive the error that webaccess is unreachable.



(33a) Let's start with the User Data location in this screen. This is where the files are going to be stored for each user's iFolder account. If you leave this on the SYS volume, you have the potential of filling up your SYS volume.

(33b) If you decide to have more than one admin for administering iFolder you will need to add them under the ADMIN NAMES section. They must be separated by a semi-colon with no space. If you do not separate them by a semi-colon then it will read the admin, not know where how to read them as separate names, and you will not be able to login to iFolder administration application.

(33c) The NETWORK DOMAIN section is asking for a DOMAIN. Not a DNS name or a host name. Remember this formula:

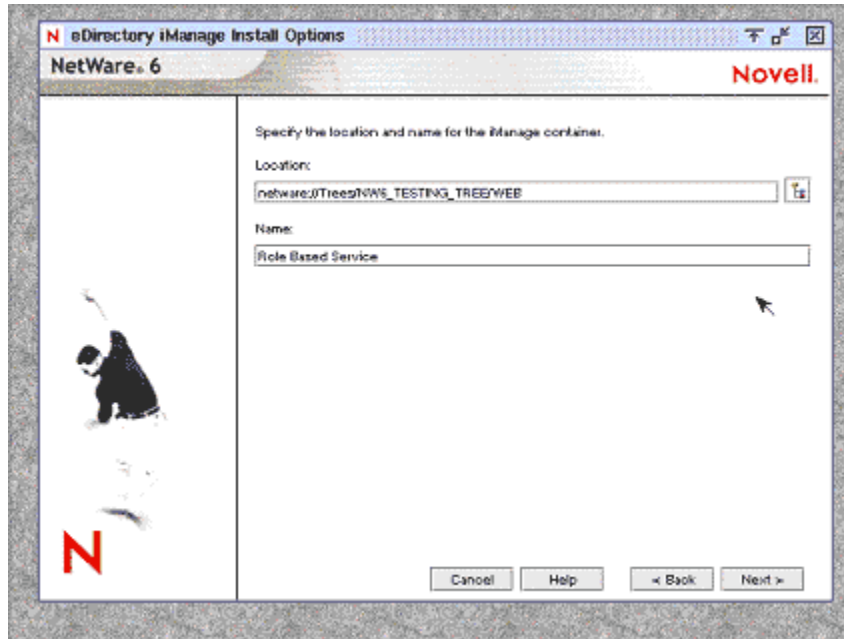
HOST NAME + DOMAIN = DNS NAME

Example:

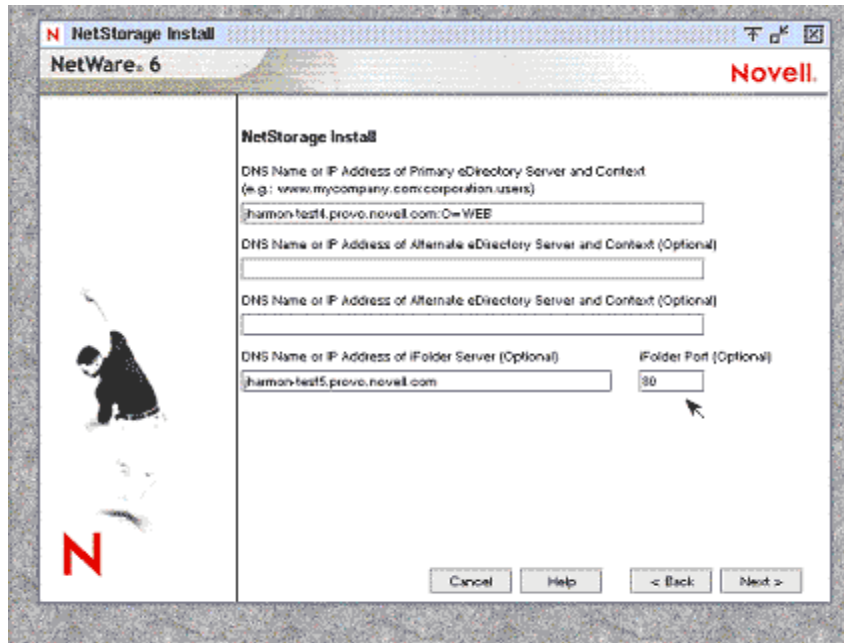
If a DNS name was NET.PROVO.NOVELL.COM, the host would be NET and the domain would be PROVO.NOVELL.COM. This should pick up the domain from the domain that was setup during the DNS resolution screen that was seen in step 18.

(33d) The ADMINISTRATORS EMAIL ADDRESS is simply for Apache's error screens. If an error occurs it will tell you to contact your administrator at the email address that is listed in this box.

**\*\* IMPORTANT \*\*** iFolder is NOT running on the default instance of Apache that is installed with the server. iFolder will be running its own instance of Apache on the server in protected memory.

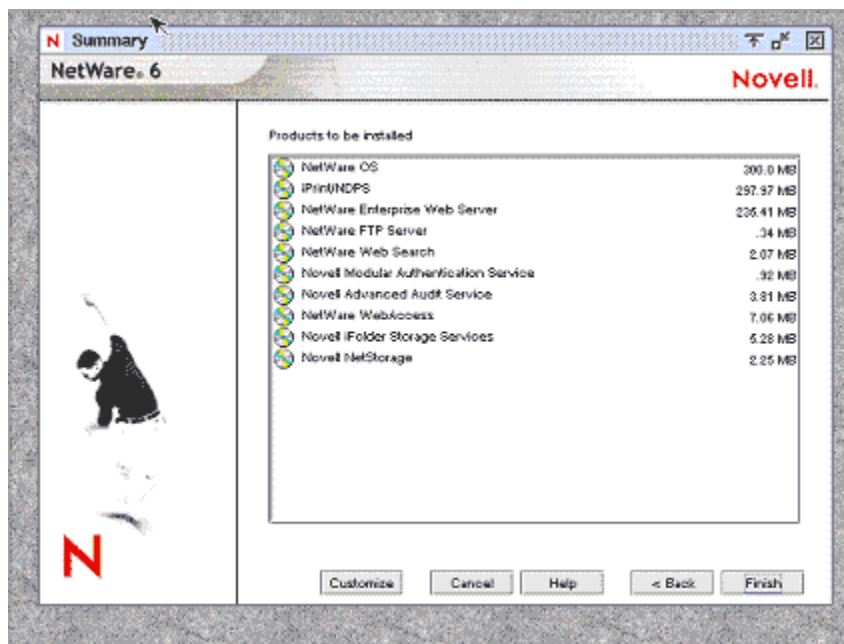


(34) If I can stress anything here it would be rights. In order for you to create this object and to properly assign roles you will need to be installing the server with rights to the root of the tree.

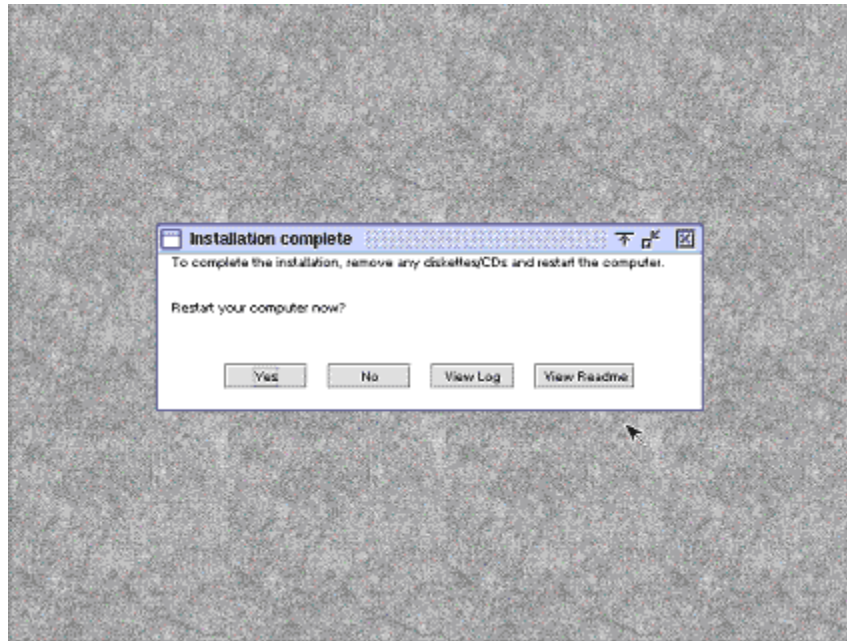


(35) NetStorage is a new type of configuration. Most of its configuration is done in the registry. In order to be able to edit that information after that installation is complete you will need to put on SP1.

**\*\* IMPORTANT \*\*** Remember, if you have configured your server with a non-registered DNS name then it will affect the performance of some products. NetStorage is one of those products. If NetStorage is configured with a non-registered domain name then it will not function.



(36) This screen is just giving you one last chance to back up and change anything if needed. Click [NEXT] to continue.



(37) After the installation has completed, be sure that you remove any diskette and CD from the server and then click YES to restart the server.

## Post-NetWare 6 product installation

### Post-Installation considerations

**\*\* NOTE \*\*** The consideration that will need to take place here is covered in the deployment section. I am covering this here in an effort to show what will happen if it is not considered at this point and time.

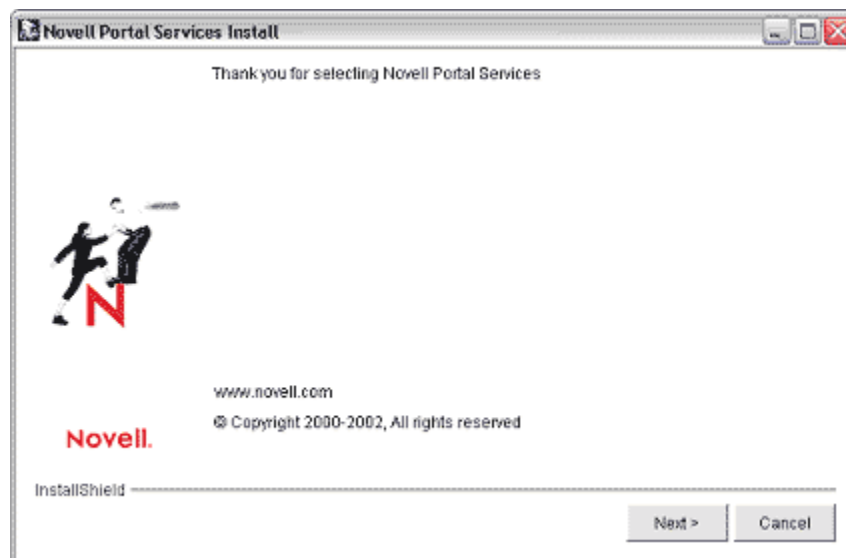
### Post-Installation of NetWare 6 products

- How to install Novell Portal Services 1.5 on NetWare 6 with Tomcat and Apache.

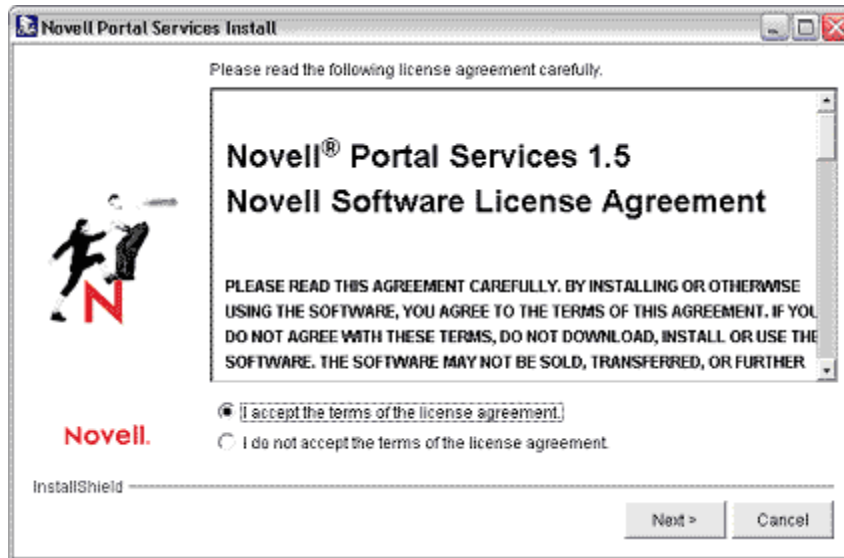
**\*\*NOTE\*\*** Unlike Novell Portal Services (NPS) 1.01, NPS 1.5 can be installed from the NetWare 6 GUI. It can also be installed from a workstation on the network if desired. However, you will need JVM 1.3 or higher to install this product. Since NetWare 6.0 has JVM 1.3 by default, we will perform this installation from the NetWare 6 server. To do this, either place the NPS 1.5 CD in the CDROM and then type CDROM at the server console, or copy the CD over to a volume on the server and run it from there. Then from the server console prompt, type in the following command:

```
JAVA -JAR VOL:/PATH/NPS_SETUP.JAR
```

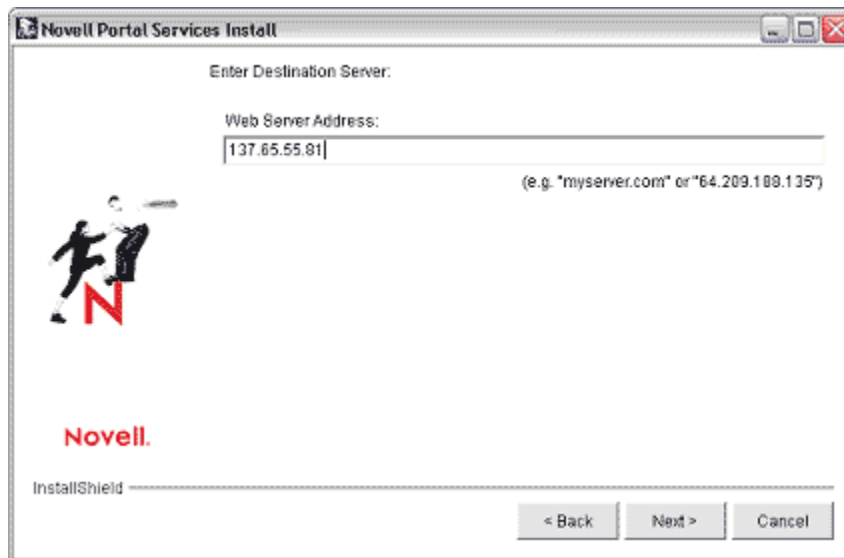
This will launch the GUI and will start the installation. If the GUI does not start after a few moments you will need to check the path that you are typing in.



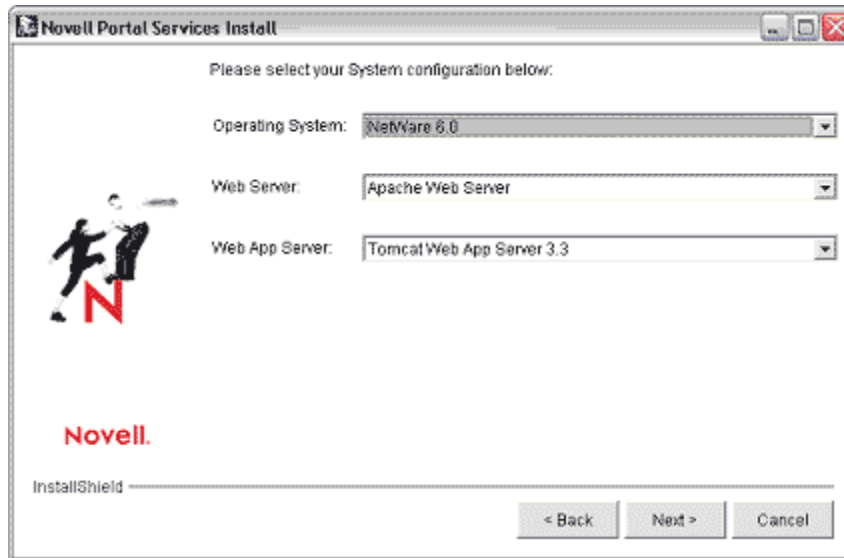
(1) The initial screen that you will be presented is the Novell Portal Services (NPS) welcome screen. Just click next to continue.



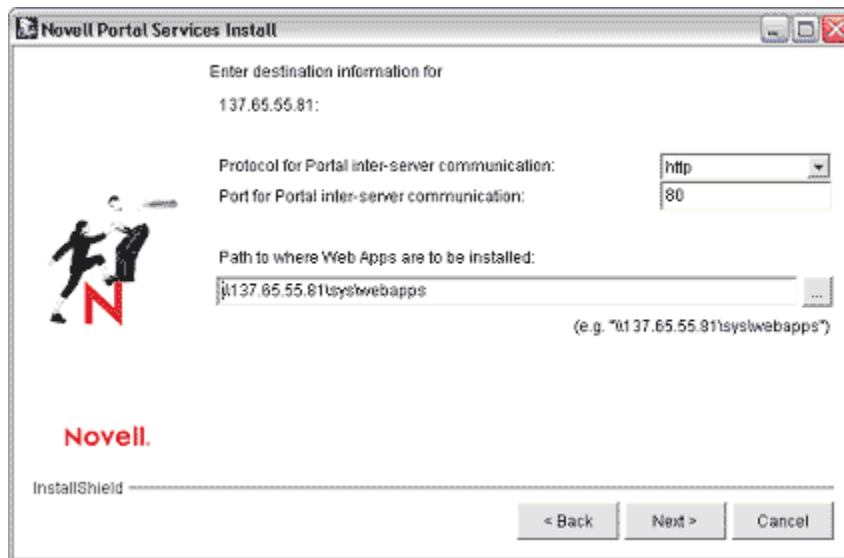
(2) The next screen is the license agreement screen. After reading the agreement, if you agree with the license, click next to continue.



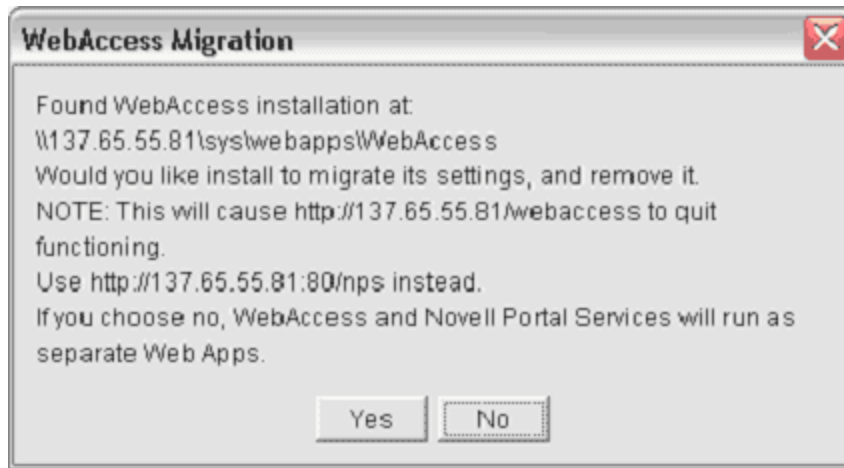
(3) Here we will enter the destination of the web server. The web server can be either on the same server or running on a different server. Either the IP address or the domain name (DNS name) can be used. However, if you use the domain name, DNS resolution will need to have been previously setup for the web server's IP address.



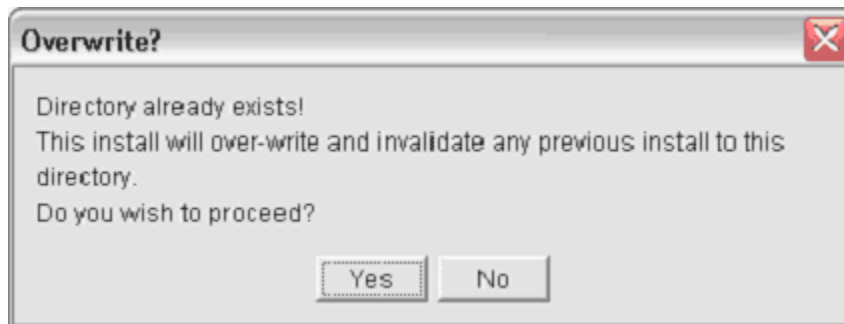
(4) Next we will choose the operating system, web server, and web application server. For this installation, we will choose NetWare 6.0, Apache, and Tomcat. Novell Portal Services 1.5 is not supported on NetWare 5.1 or lower.



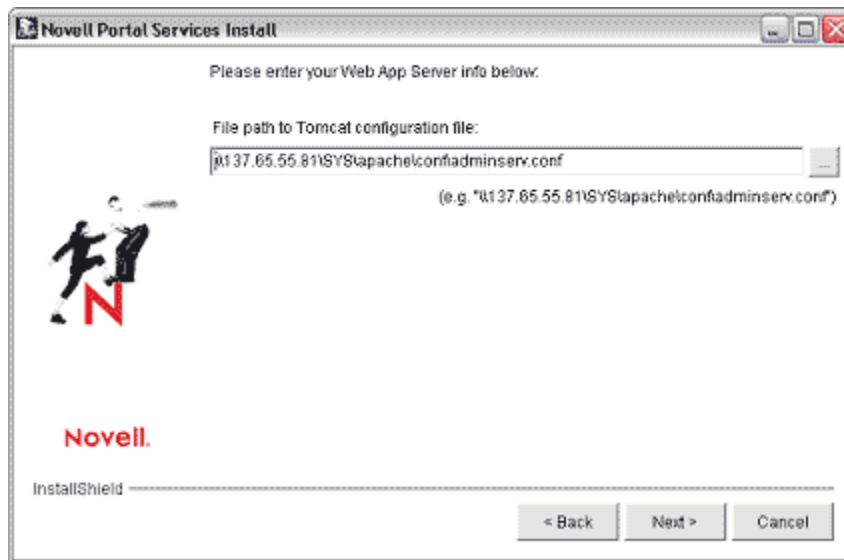
(5) The first section within this dialog box is the "Protocol Inter-Server Communication" section. This section is for encryption between the portal server and services running on other servers. This is not to be confused with encryption between the web server and a user's browser. This is a service that would be provided with the web server, iChain, or similar service. The next section is for pointing to the directory structure within the web application server that will hold the NPS web application. Click NEXT to continue.



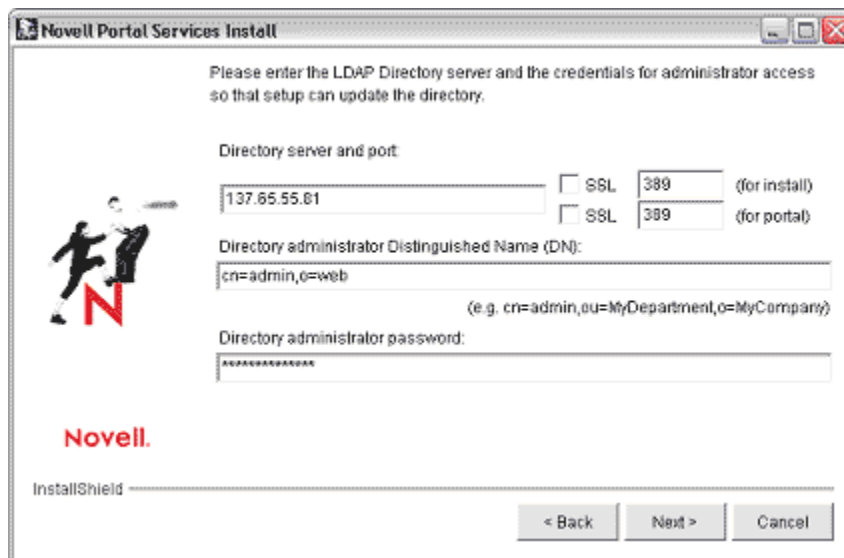
(6) In this dialog box you are asked whether or not you wish to upgrade NetWare WebAccess. If you choose to upgrade, NetWare WebAccess will no longer be accessed from the /webaccess URI. It will be accessed from the /nps URI. [If you are going to upgrade, you will need to look at the following document.] In this example we are going to choose NO to the upgrade. This will allow us to have Novell Portal Services running side by side with NetWare WebAccess. This means that NetWare WebAccess will be accessed from the /webaccess URI, and Novell Portal Services will be accessed from the /nps URI.



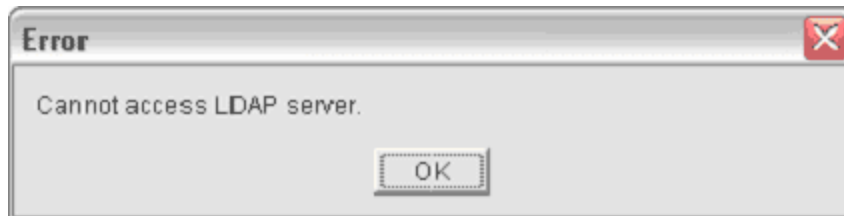
**\*\* NOTE \*\*** You will only get this message if an NPS directory exists under SYS:\WEBAPPS from a previous installation of Novell Portal Services. If you choose YES, the NPS directory will be overwritten. This message is not an upgrade option. This message is telling you that it will overwrite the directory. If you do not want to overwrite the directory, choose NO. In this example we will choose YES. Again, you will only receive this dialog box if you already had the NPS directory installed. If you do not get this dialog box, don't worry. You are not missing a step here.



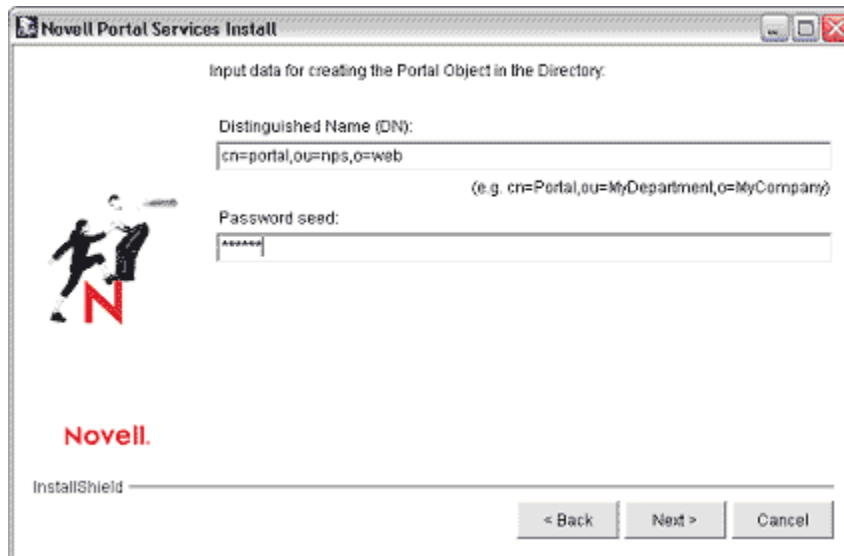
(7) The ADMINSEV.CONF file is the main configuration file for Apache on NetWare. It should pick this path up by default. If the path is incorrect, or if the file does not exist, the installation will not allow you to proceed. Click NEXT to continue.



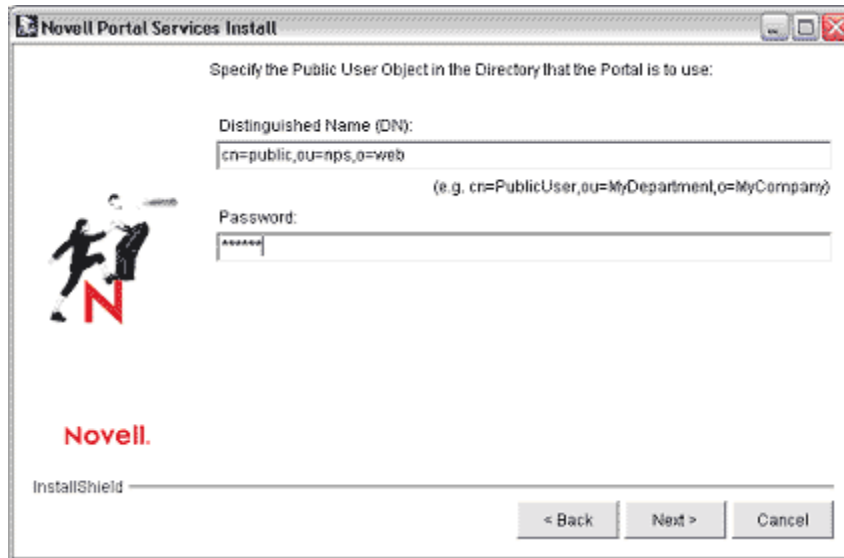
(8) Now you are prompted to setup the LDAP server. You will need a user with administration rights before proceeding. The DIRECTORY SERVER AND PORT section is where you input the IP address or Domain Name for the LDAP server. If you are going to use the Domain Name, you will first need to have DNS resolution setup. In this example we will use the IP address. To the right of this section are the SSL (encryption) options. The first one is if you are installing Novell Portal Services over a remote connection and you want to encrypt the install. The second is used if your LDAP server is not on the same server as your portal server and you want the information between the portal server and the LDAP server to be encrypted. Turning this option on will affect performance. The next two sections are for the administrator's user name and password. This section requires the fully distinguished, type-full name in LDAP format (using commas).



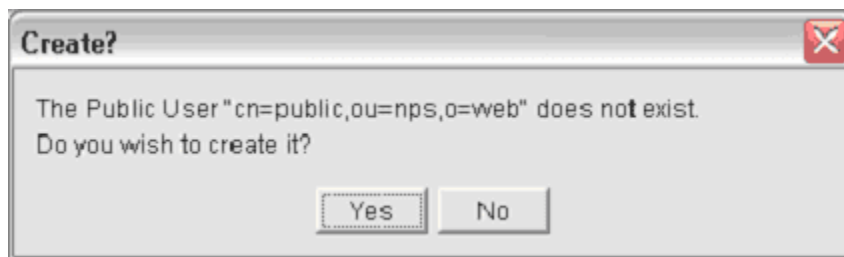
**\*\*NOTE\*\*** This error will occur if there is an issue resolving the administrator's user name and password. This can be caused for several reasons. One, you have chosen not to install over SSL and you do not have ALLOW CLEAR TEXT PASSWORDS checked in the LDAP Group object. Two, you are not using eDirectory. (eDirectory is installed on NetWare 6 by default). Three, your LDAP server is not running. Four, you are using NDS format instead of LDAP format (periods instead of commas). If you do not get this error, don't worry. This error will only occur if NPS is unable to resolve the administrator's name. [For more troubleshooting refer to the following document.]



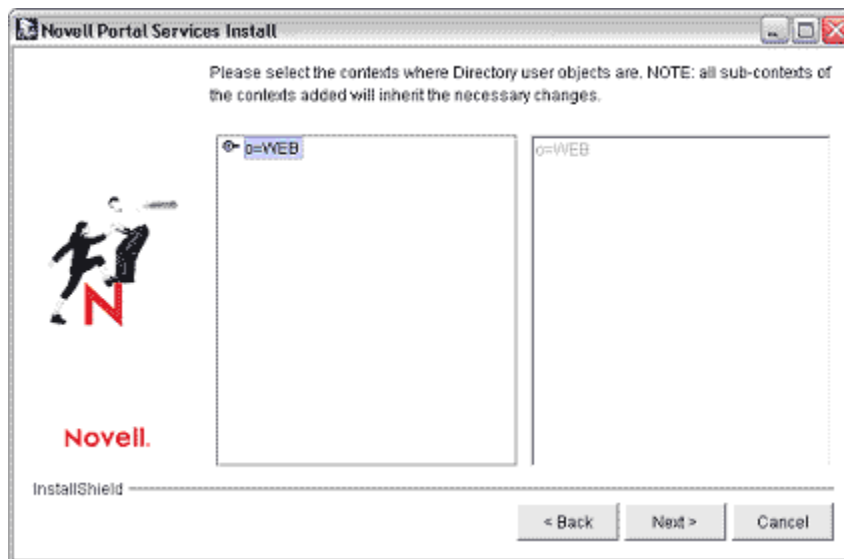
(9) Now we need to create the Portal Configuration Object (PCO). To make administration simple, it is suggested that you create an Organizational Unit to place all of your Portal objects in. This is not necessary to run Novell Portal Services, but will be done in this example. Using ConsoleOne, an NPS directory was created under the main Organization. Here we will place the PCO, gadgets, public user, etc. Again, this needs to be in LDAP format, fully distinguished, and type-full. In this example we will name the object 'Portal'.



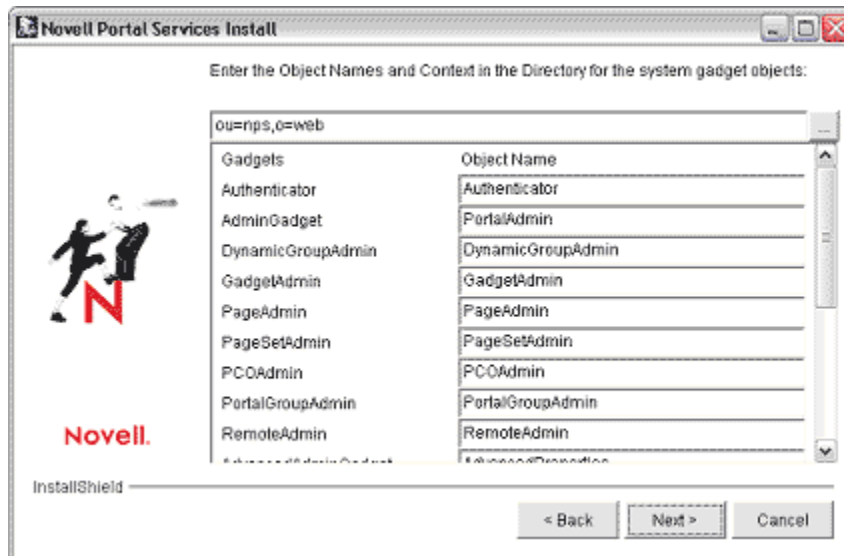
(10) Now we need to create a public user that will be used to hold the authentication gadget, which allows us to login to Novell Portal Services. Again, we will place this object under the NPS directory and the public user will need to be specified in LDAP format, using the fully distinguished, and type-full name.



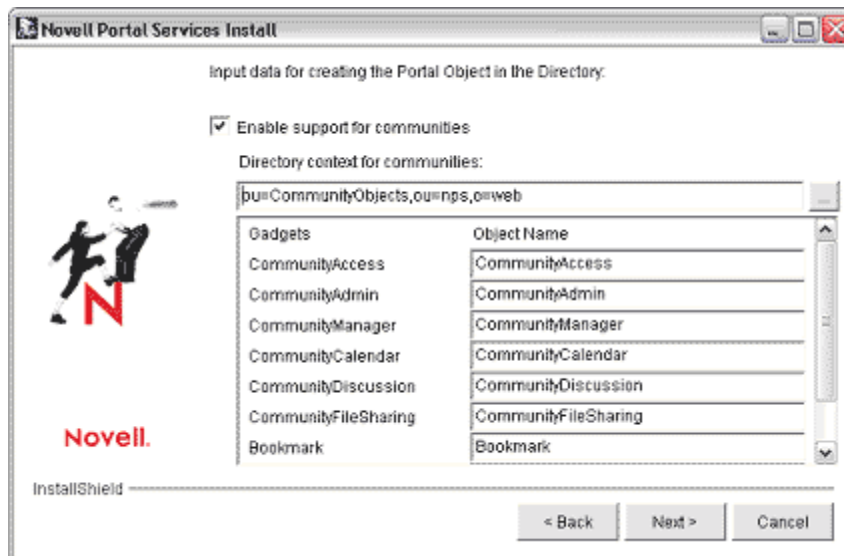
**\*\*NOTE\*\*** You do not need to worry about creating the public user before installing Novell Portal Services. If the user has not been created, you will receive the dialog bog asking if you want to create the user. Click YES to continue.



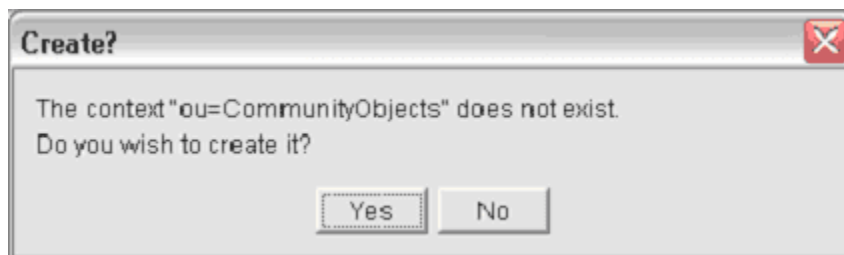
(11) This is where you specify the containers that you would like to use context-less login. If you have multiple O's at the same level, you can hold down your CTRL key and select more than one O. Rights flow down from there.



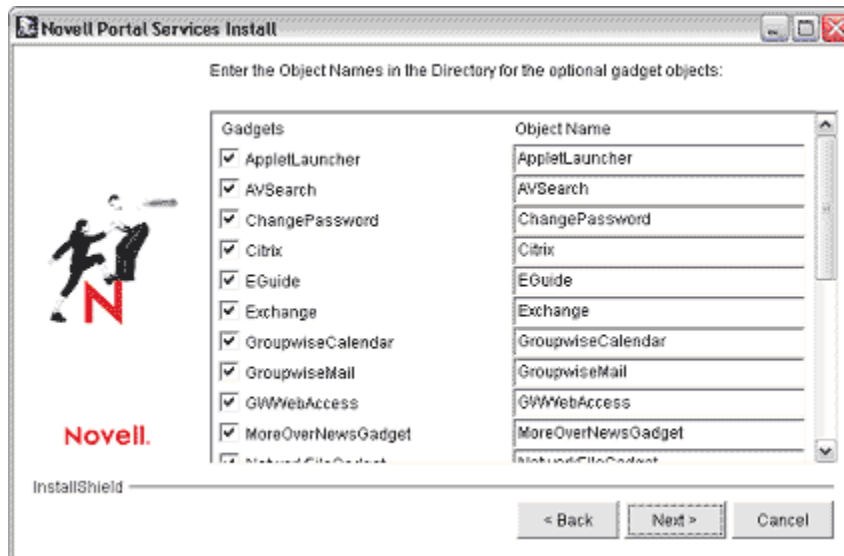
(12) You have the option to change the name and location of the gadgets being installed. For this example we will leave everything at their defaults and click NEXT to continue.



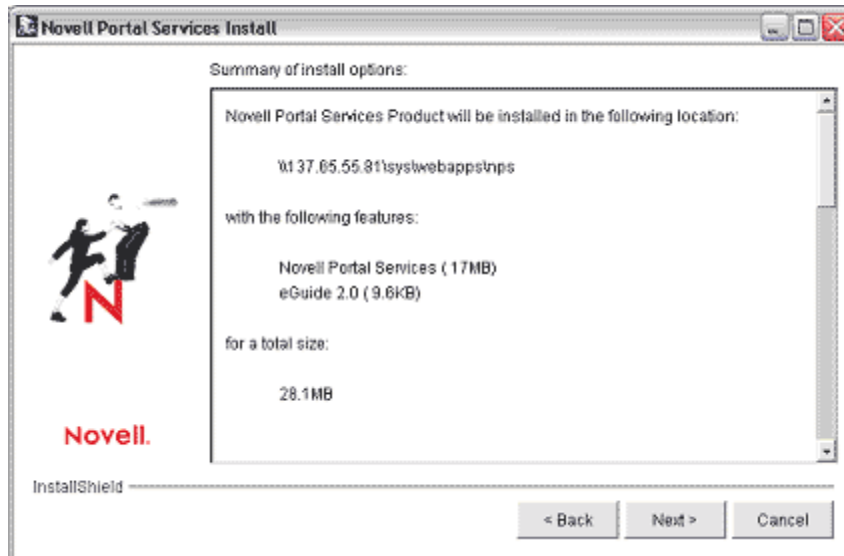
(13) By default the community's OU will be created under the O or OU that you have specified to hold portals gadgets. If you want community support, make sure that the ENABLE SUPPORT FOR COMMUNITIES box is checked. Again, you can change the name and location of the gadgets that will be installed if desired. For this installation we will accept the defaults and click NEXT to continue.



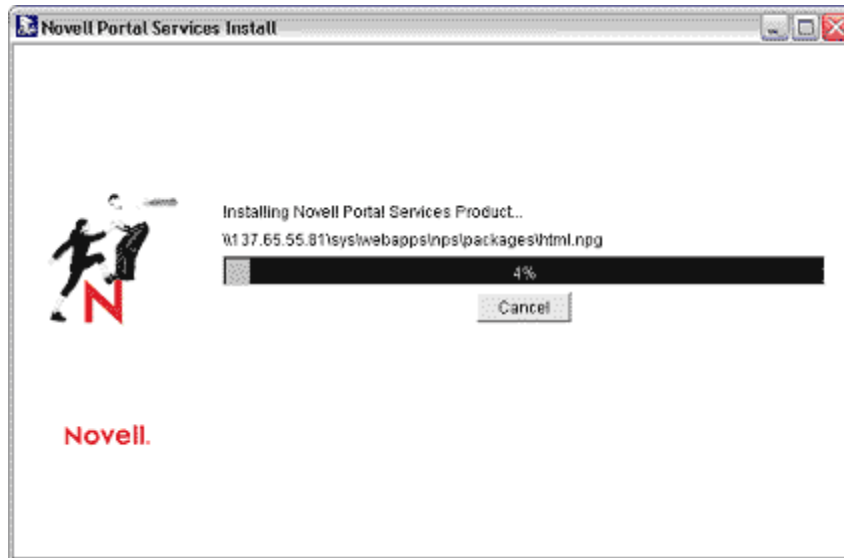
**\*\*NOTE\*\*** This dialog box will appear if you chose the option to install the communities OU. Click YES to create the OU and continue with the installation.



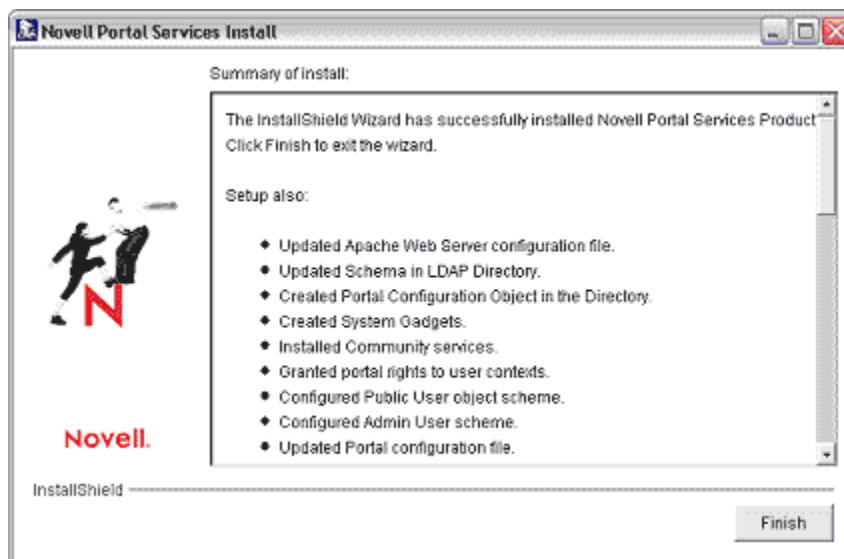
(14) If you have any other \*.NPG files that the installation can pick up, you will have the option to install them here. Click NEXT to continue.



(15) Here is the list of which components will be installed and where, along with the amount of hard-drive space NPS will consume. Click NEXT to continue.



**\*\*NOTE\*\*** Installation time will vary depending on the system you are installing Novell Portal Services from.



(16) The last screen is just a summary of what was installed. Click FINISH.

(17) Now we need to test the installation of Novell Portal Services. To do this we will go to the following URL: <http://DomainName/nps> -OR- <http://IPaddress/nps>

(18) Increasing the amount of RAM that Tomcat can use.

Once Portal Services is installed we will need to adjust the amount of Memory that is allocated to Tomcat. The reason for this is that applications such as NPS and eGuide are more memory-intensive web applications and will require a higher memory setting. If this adjustment is not done the users may receive a java.lang.outofmemory error.

(A) Open your TOMCAT33.NCF file found in the SYS:/TOMCAT/33/BIN directory. Here we will need to add two parameters to the file. One for the minimum heap size and one for the max heap size. The parameters are as follows:

-Xms128m -Xmx256m

(B) Now we will insert these parameters into the TOMCAT33.NCF file. At the bottom of the file you will find the following line:

```
java -envCWD=$TOMCAT_HOME -classpath $TOMCAT_CLASSPATH -Dtomcat.home=SYS:\tomcat\33
org.apache.tomcat.startup.Main -f sys:/tomcat/33/conf/nwserver.xml %1
```

(C) We will add the heap size parameters to this line, right after the "java" directive. The new line will look as follows.

```
java -Xms128m -Xmx256m -envCWD=$TOMCAT_HOME -classpath $TOMCAT_CLASSPATH -
Dtomcat.home=SYS:\tomcat\33 org.apache.tomcat.startup.Main -f sys:/tomcat/33/conf/nwserver.xml %1
```

**\*\* IMPORTANT \*\*** Each java process is allocated 512 meg of user space within the RAM that it can use. Some of that is used for java threads, socket communication, etc. So what is really available is not much more than 400 meg which can be allocated to a java process. This information can be seen at the server by typing in the command `JAVA -SHOW` and then finding out what the ID is for the Tomcat java process. Then at the server console you can type `JAVA -SHOWMEMORY#` with the # representing the Tomcat process ID. Then switch over to the logger screen and you should see how much memory is being allocated within the Tomcat process. An example is shown below:

```
JAVA -SHOW
```

```
Classname                                ID
=====
org.apache.tomcat.startup.main..... 390
```

```
JAVA -SHOWMEMORY390
```

```
Memory Statistics For Class: org.apache.tomcat.startup.Main
```

```
-----
Reserved Heap: 68161536
Committed Heap: 11124732
Reserved Virtual Memory Pool: 67108864
Committed Virtual Memory Pool: 11272192
NLM Data Memory: 225280
Per Thread Data And OS Stacks: 3403776
Virtual Memory Pool Overflow: 0
JVM Tracking Memory: 23381
Socket Communication Memory: 101616
```

```
-----
Total Committed Virtual Memory: 22622204
Total Physical Memory: 3528773
Total Committed JVM Memory: 26150977
```

**\*\* NOTE \*\*** The main statistics to look at here are the reserved heap and the committed heap. The reserved heap represents our -Xmx switch. This is the maximum memory that is allocated to the individual java process. What we show here by default is 68161536 which equates to 65 meg. The committed heap represents the -Xms switch. This is the minimum amount of memory that will be committed to the individual java process. In this instance we have 11124732 which equated to 8 meg of committed memory.

(D) To verify that our switches have changed our committed and reserved heap sizes, we will need to take down java and then restart tomcat. See example below:

```
JAVA -EXIT
```

**\*\* IMPORTANT \*\*** If the console comes back with `MODULE JAVA.NLM UNLOADED`, then you can startup tomcat by typing in `TOMCAT33` at the console prompts. If it comes back that it is still cleaning up resources in the background, and you have a console prompt, then you can type in `JAVA -EXIT` again to force java down.

(E) Once tomcat is started again, you can check the memory again and it should now show up.

```
Memory Statistics For Class: org.apache.tomcat.startup.Main
```

```
-----
Reserved Heap: 272633856
Committed Heap: 136314876
```

**\*\* TROUBLESHOOTING \*\*** If you do not show the new memory size, one of the following probably happened.

- (A) You reloaded Tomcat before JAVA was completely unloaded.
- (B) You specified a maximum heap (-Xmx) that was too large. The most that this can be is around -Xmx386m without addition configuration.
- (C) Your -Xms parameter is larger than your -Xmx parameter.

On NetWare 6 we have the option of loading the AUTOEXEC.BAT file with a -u switch to increase the amount of memory that can be used by the JVM. This switch can be used in conjunction with java heap parameters to dedicate more memory to memory-intensive Web Applications. Examples of memory-intensive web applications on NetWare 6 are Novell Portal Services 1.5 and eGuide 2.0

There are a few things that should be noted before using the -u parameter. The -u parameter allows you to specify a higher amount of memory than exists on your server. This could create problems if you are trying to specify memory that you don't have. So what should be done to avoid this? Well, don't exceed the amount of RAM that you have on your server with the -u parameter. In fact it would be well advised for you to not go above three-quarters of the amount of RAM that you have. Remember that the JVM is not the only application on the server that will be using the RAM. Take care to leave some RAM available for other processes.

(F) Edit the AUTOEXEC.BAT file found at the root of the C:\ drive. It should look similar to the example below:

```
C:
CD \NWSERVER
SERVER
```

(G) After editing the file to add the -u switch, it should look similar to the example below, with the exception that the number value may be different depending on the desired amount of RAM needed for the JAVA process.

```
C:
CD \NWSERVER
SERVER -u1000000000
```

**\*\* NOTE \*\*** In this example we are specifying 1 GIG of RAM to be used. The parameter is measured in bytes. This will get past the limit specified in step (4). You should now be able to increase your heap size above 400 MEG.

- o eGuide Installation

```
Synthetic Time is being issued on partition ".T=NW6_TESTING_TREE."
7-20-2002 10:16:17 am: DS-10110.20-270
Synthetic Time is being issued on partition ".T=NW6_TESTING_TREE."
NW6_TRAINING:cdrom
Loading Module CDROM.NLM [ OK ]
Auto-Loading Module CD9660.NSS [ OK ]
Moving ISO9660 CD volume 'EGUIDE' to active state
Volume EGUIDE set to the ACTIVE state.
Mounting Volume EGUIDE
** EGUIDE mounted successfully
ISO 9660 CD-ROM Support Module loaded successfully.
Auto-Loading Module CDHFS.NSS [ OK ]
MAC HFS CDROM Support Module loaded successfully.
NW6_TRAINING:volumes
Mounted Volumes Name Spaces Flags
SYS DOS, MAC, NFS, LONG NSS
_ADMIN DOS, MAC, NFS, LONG NSS P
EGUIDE DOS, MAC, NFS, LONG NSS RO P
3 volumes mounted
NW6_TRAINING:java -cp eguide:/install/netware/install.zip install
```

(1) To install eGuide on NetWare 6 we must use the class path (-cp) java command. At the server console prompt type the following:

```
JAVA -CP VOL:/INSTALL/NETWARE/INSTALL.ZIP INSTALL
```

This will launch the GUI installation.

**\*\* NOTE \*\*** eGuide 2.0 MUST be installed from the server. It cannot be installed via remote console.



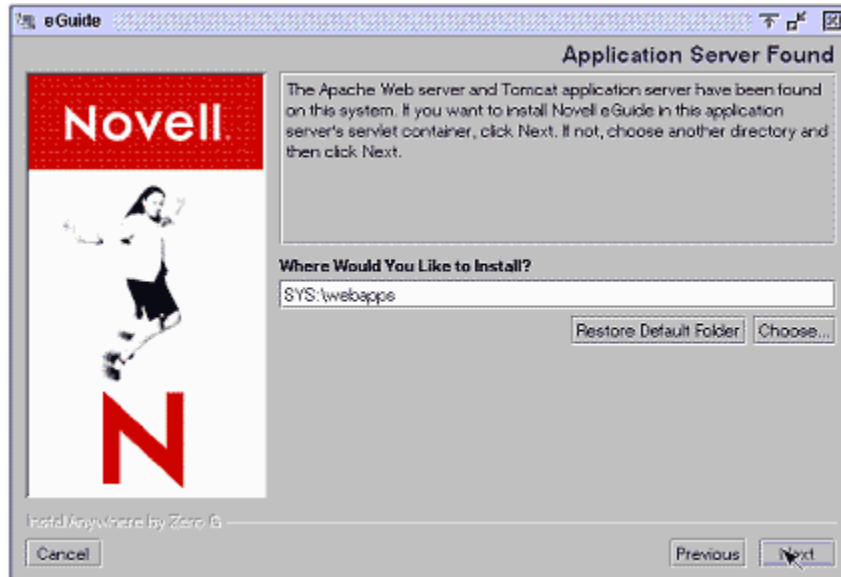
(2) Once the GUI has launched you will receive the Welcome screen. Just click [NEXT] to continue.



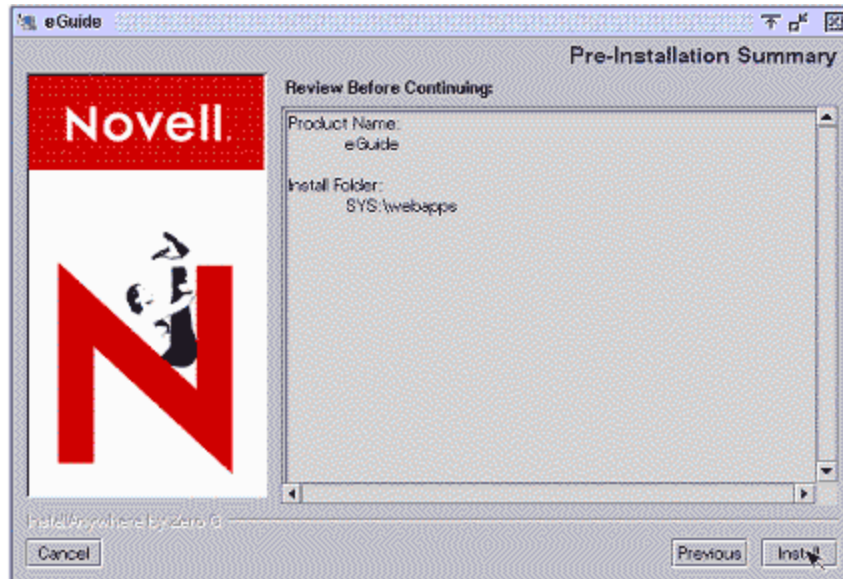
(3) Here we are explaining that NetWare 6 shipped with Apache and Tomcat pre-installed. Just click [NEXT] to continue.



(4) The next screen is the license agreement. Choose the Accept radio button and then click [NEXT] to continue.



(5) The next screen should pull up the SYS:\WEBAPPS directory structure by default. This directory is where Novell Web Apps (running on Tomcat) reside. Click [NEXT] to continue.



(6) This is your last chance to go back and change anything. (Although nothing should need to be changed.) Click [INSTALL] to continue.



(7) The last screen is just an installation complete screen. Click [DONE].

(8) eGuide can be a memory-intensive application. We would normally adjust the amount of RAM that is associated to Tomcat, but that was already done with Novell Portal Services.

(9) To have the installation take effect, type the following at the server console:

```
JAVA -EXIT
NVXADMIN
NVXADMUP
TOMCAT33
```

## Part 2: Deployment

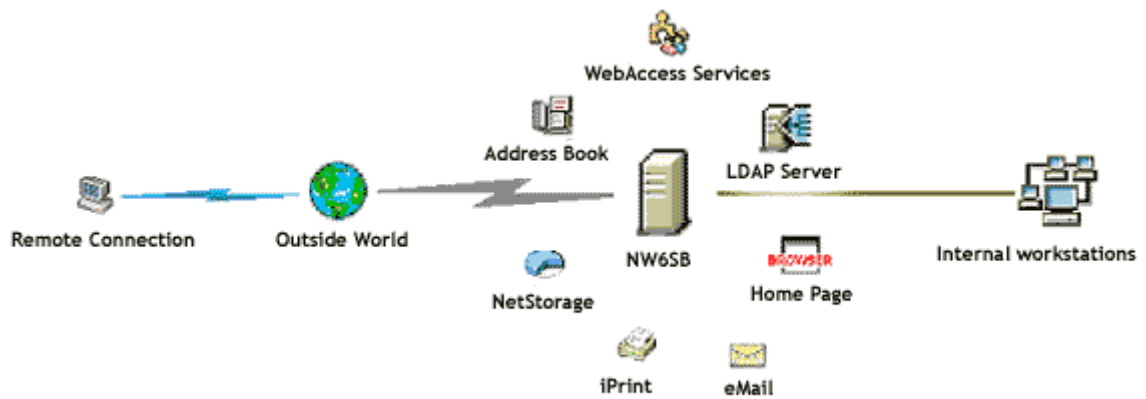
---

### Fresh Install vs. Upgrade and Migration

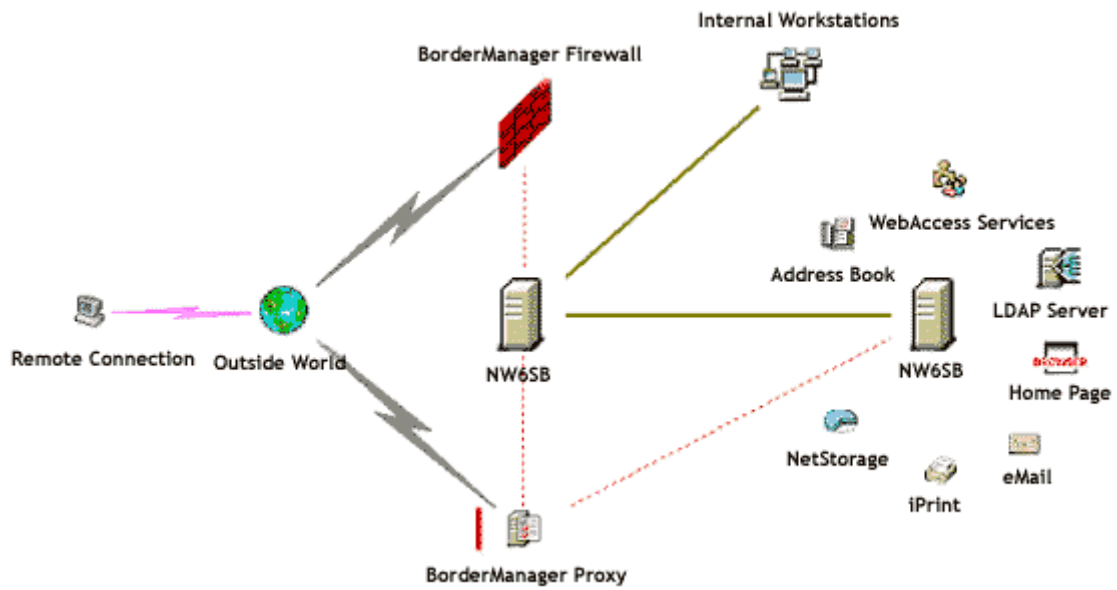
- Upgrade and Migration Considerations
  - Is SSL up and running?
  - Is the CA valid?
  - What is the version of DS and OS that I am currently running?
  - Have I performed all of the pre-migration steps before installing NetWare 6?
- Fresh Install Considerations
  - Am I going to install into a new or existing tree?
  - Am I going to use the multiple or single IP address option?

### Small Business Deployment (1-2 servers)

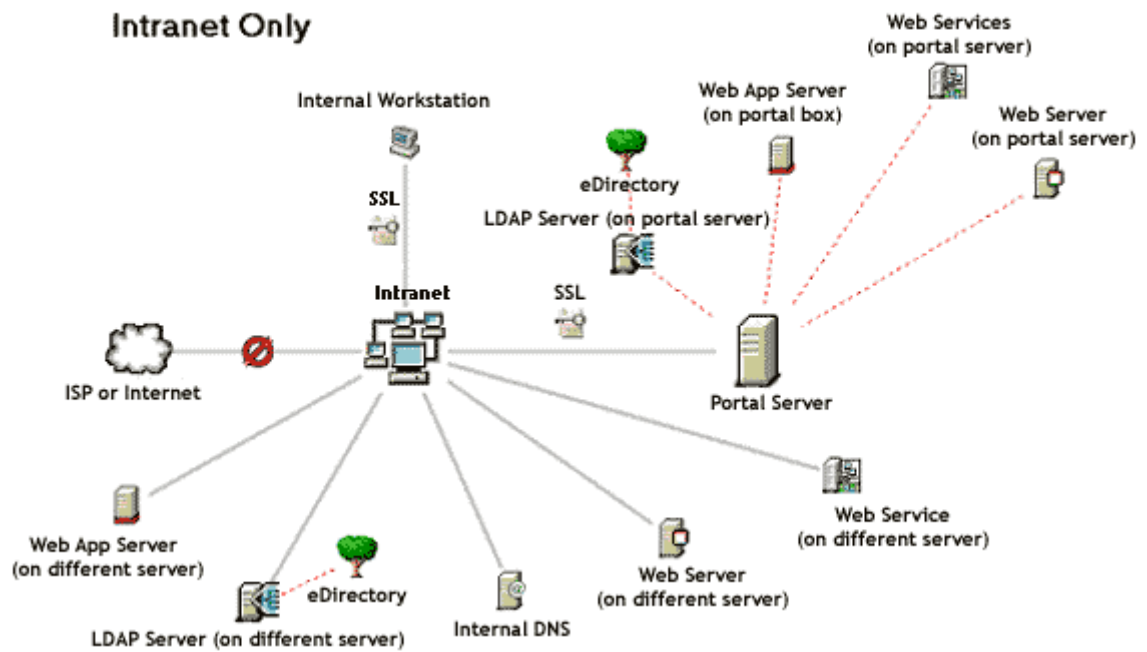
#### One Server Deployment



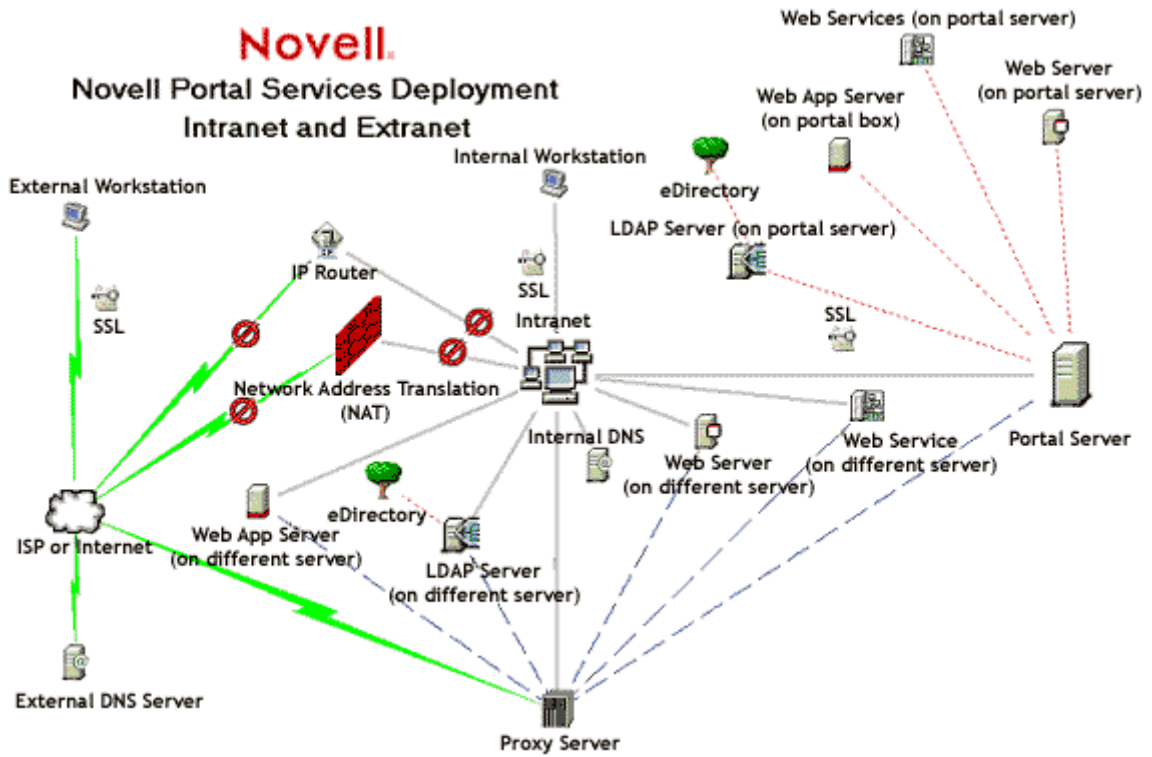
## Two Server Deployment



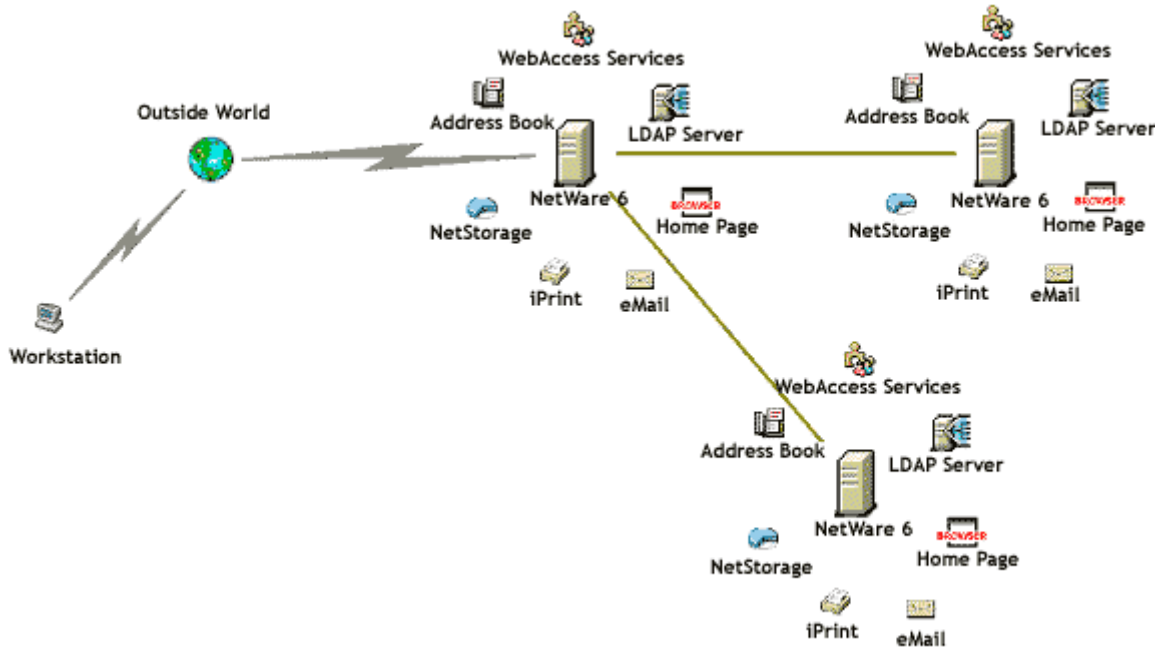
## Intranet Deployment



## Intranet and Extranet Deployment



# Extranet Deployment



## Part 3: Increasing the Amount of RAM that Tomcat Can Use

---

Here is how to increase the amount of RAM that Tomcat can use.

(1) Open your TOMCAT33.NCF file found in the SYS:/TOMCAT/33/BIN directory. Here we will need to add two parameters to the file. One for the minimum heap size and one for the maximum heap size. The parameters are as follows:

```
-Xms128m -Xmx256m
```

(2) Now we will insert these parameters into the TOMCAT33.NCF file. At the bottom of the file you will find the following line:

```
java -envCWD=$TOMCAT_HOME -classpath $TOMCAT_CLASSPATH -Dtomcat.home=SYS:\tomcat\33  
org.apache.tomcat.startup.Main -f sys:/tomcat/33/conf/nwserver.xml %1
```

(3) We will add the heap size parameters to this line, right after the "java" directive. The new line will look as follows.

```
java -Xms128m -Xmx256m -envCWD=$TOMCAT_HOME -classpath $TOMCAT_CLASSPATH -  
Dtomcat.home=SYS:\tomcat\33 org.apache.tomcat.startup.Main -f sys:/tomcat/33/conf/nwserver.xml %1
```

**Important:** Each java process is allocated 512 meg of user space within the RAM that it can use. Some of that is used for java threads, socket communication, etc. So what is really available is not much more than 400 meg which can be allocated to a java process. This information can be seen at the server by typing in the command JAVA -SHOW and then finding out what the ID is for the Tomcat java process. Then at the server console you can type JAVA -SHOWMEMORY# with the # representing the Tomcat process ID. Then switch over to the logger screen and you should see how much memory is being allocated within the Tomcat process. An example is shown below:

```
JAVA -SHOW
```

```
Classname                               ID  
=====                               =====  
org.apache.tomcat.startup.main..... 390
```

```
JAVA -SHOWMEMORY390
```

```
Memory Statistics For Class: org.apache.tomcat.startup.Main
```

```
-----  
Reserved Heap: 68161536  
Committed Heap: 11124732  
Reserved Virtual Memory Pool: 67108864  
Committed Virtual Memory Pool: 11272192  
NLM Data Memory: 225280  
Per Thread Data And OS Stacks: 3403776  
Virtual Memory Pool Overflow: 0  
JVM Tracking Memory: 23381  
Socket Communication Memory: 101616
```

```
-----  
Total Committed Virtual Memory: 22622204  
Total Physical Memory: 3528773  
Total Committed JVM Memory: 26150977
```

**Note:** The main statistics to look at here are the reserved heap and the committed heap. The reserved heap represents our -Xmx switch. This is the maximum memory that is allocated to the individual java process. What we show here by default is 68161536 which equates to 65 meg. The committed heap represents the -Xms switch. This is the minimum amount of memory that will be committed to the individual java process. In this instance we have 11124732 which equated to 8 meg of committed memory.

(4) To verify that our switches have changed our committed and reserved heap sizes, we will need to take down java and then restart tomcat. See example below:

## JAVA -EXIT

**Important:** If the console comes back with MODULE JAVA.NLM UNLOADED, then you can startup tomcat by typing in TOMCAT33 at the console prompts. If it comes back that it is still cleaning up resources in the background, and you have a console prompt, then you can type in JAVA -EXIT again to force java down.

(5) Once tomcat is started again then you can check the memory again and it should now show up.

Memory Statistics For Class: org.apache.tomcat.startup.Main

-----  
Reserved Heap: 272633856  
Committed Heap: 136314876

**Troubleshooting:** If you do not show the new memory size then one of the following probably happened.

- (A) You reloaded Tomcat before JAVA was completely unloaded.
- (B) You specified a maximum heap (-Xmx) that was too large. The most that this can be is around -Xmx386m without addition configuration.
- (C) Your -Xms parameter is larger than your -Xmx parameter.

On NetWare 6 we have the option of loading the AUTOEXEC.BAT file with a -u switch to increase the amount of memory that can be used by the JVM. This switch can be used in conjunction with java heap parameters to utilize dedicate more memory to memory intensive Web Applications. Examples of memory intensive Web application on NetWare 6 are Novell Portal Services 1.5 and eGuide 2.0.

There are a few things that should be noted before using the -u parameter. The -u parameter allows you to specify a higher amount of memory than exists on your server. This could create problems if you are trying to specify memory that you don't have. So what should be done to avoid this? Well, don't exceed the amount of RAM that you have on your server with the -u parameter. In fact it would be well advised for you to not go above three quarters of the amount of RAM that you have. Remember that the JVM is not the only application on the server that will be using the RAM. Take care to leave some RAM available for other processes.

(6) Edit the AUTOEXEC.BAT file found at the root of the C:\ drive. It should look similar to the example below:

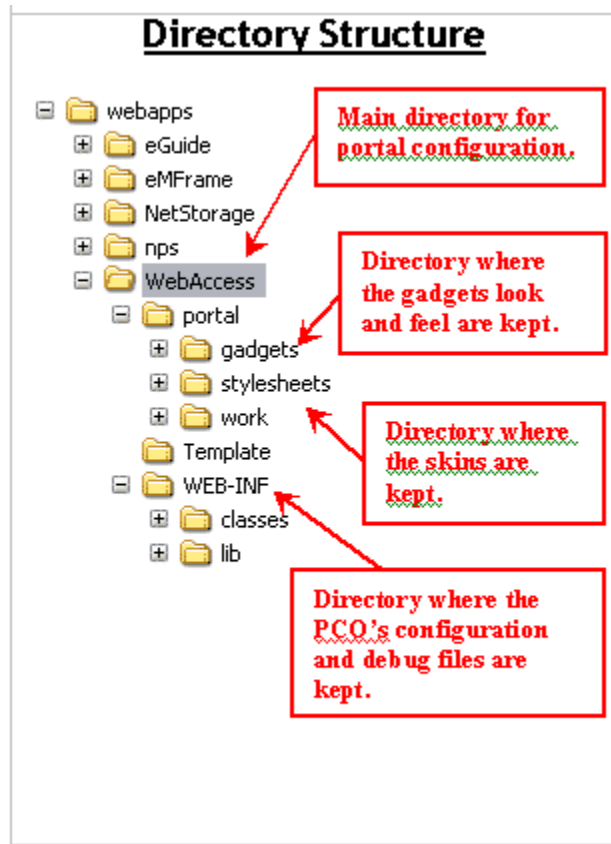
```
C:  
CD \NWSERVER  
SERVER
```

(7) After editing the file to add the -u switch, it should look similar to the example below, with the exception that the number value may be different depending on the desired amount of RAM needed for the JAVA process.

```
C:  
CD \NWSERVER  
SERVER -u1000000000
```

**Note:** In this example we are specifying 1 GIG of RAM to be used. The parameter is measured in bytes. This will get past the limit specified in step (4). You should now be able to increase your heap size above 400 MB.

## Part 4: NetWare WebAccess Configuration



### Commands

JAVA -SHOW = Allow you to see what java processes are running  
TOMCAT33 -STOP = Brings down Tomcat process within java. (not recommended)  
JAVA -EXIT = Brings down all java processes. (recommended)  
TOMCAT33 = Brings up Tomcat  
NVXADMDN = Brings down Apache  
NVXADMUP = Brings up Apache

### Authentication

#### DEFAULT

LDAP  
CONTEXTLESS

#### AVAILABLE

CONTEXT = LDAP FORMAT

### Files

PORTALSERVLET.PROPERTIES = Found in the SYS:/WEBAPPS/NPS/WEB-INF directory. This configuration file holds configuration LDAP, logging, rendering, etc., information for the Portal Configuration Object (PCO).  
DEBUG.XML = Found in the SYS:/WEBAPPS/NPS/WEB-INF directory. This is the main debug log for NPS.  
ADMINSERV.CONF = Found in the SYS:/APACHE/CONF directory. This is the main Apache configuration file that holds the rights and access to Apache directories.  
NWWAAPACHE.CONF = Found in the SYS:/WEBAPPS/WEBACCESS/WEB-INF directory. This is an appendage to the ADMINSERV.CONF file through an include statement. It is a configuration file that holds the rights and access to

directories.

NPS.PDT = Found in the SYS:/SYSTEM directory. This is the file that is used for holding previously installed Portal information.

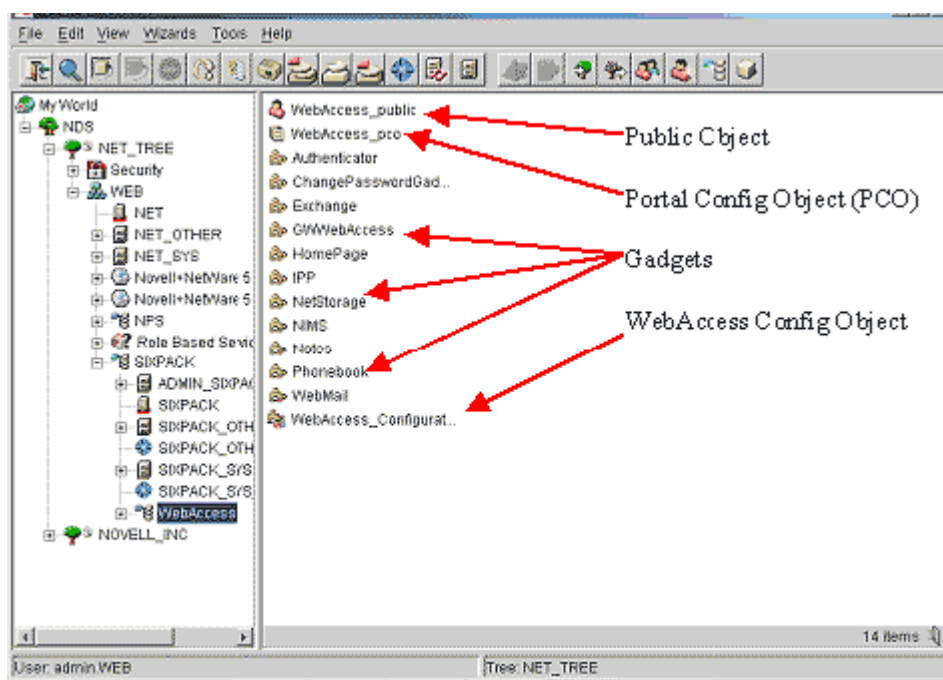
NWAPPS-WEBACCESS.XML = Found in the SYS:/TOMCAT/33/CONF directory. This file defines how Tomcat is to access NPS.

NWWA.JAR = Snapins for NetWare Web Access.

## NETWARE WEB ACCESS GADGET CONFIGURATION

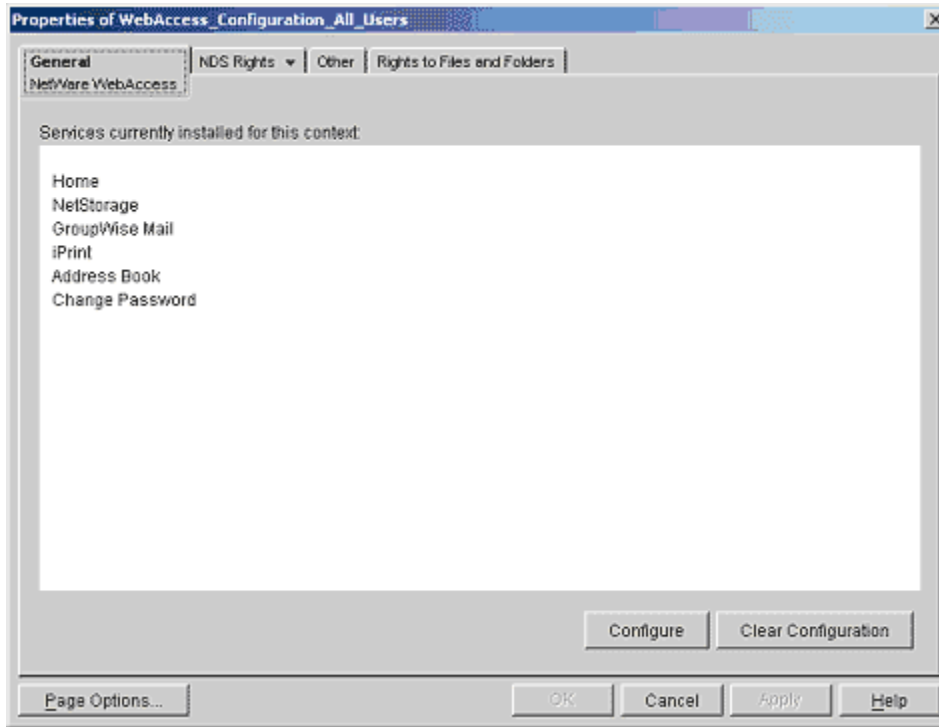
NetWare Web Access creates several NDS objects under the WEBACCESS container. They are as follows:

- Public object to present the authentication screen
- Portal Configuration Object
- Gadget
- Web Access Configuration Object

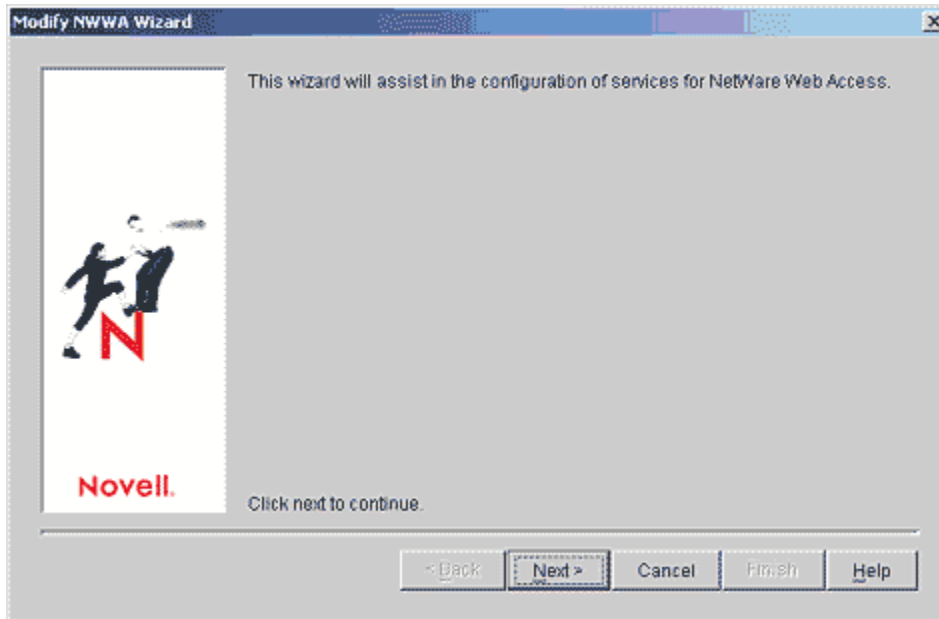


### Web Access Configuration Object

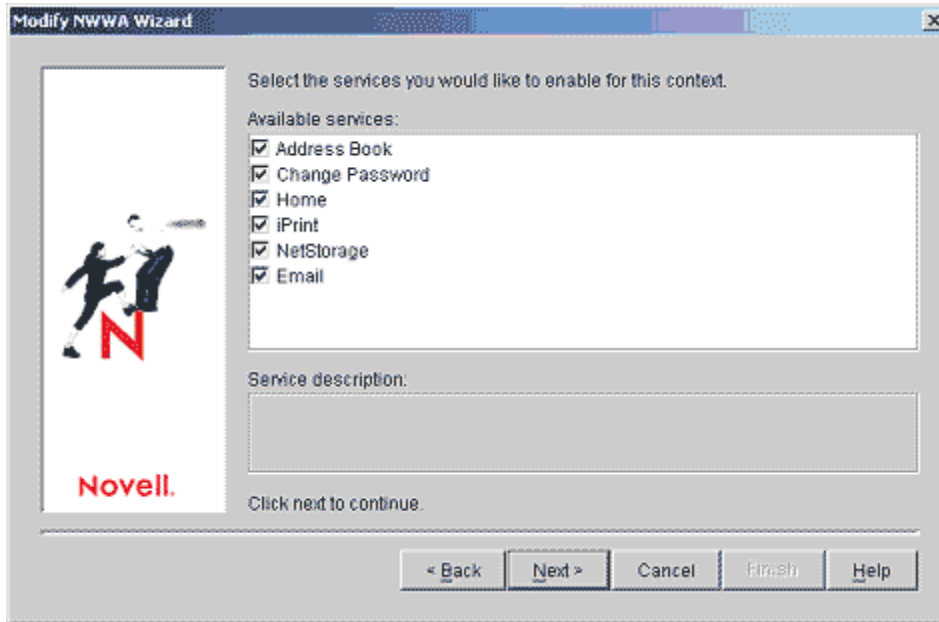
(A) Below is a list under the General Tab for the Web Access Configuration Object. Here are listed the Gadgets that are already installed and configured. To clear the configuration select the CLEAR CONFIGURATION button. To configure each object select the CONFIGURE button and it will take you to the next screen.



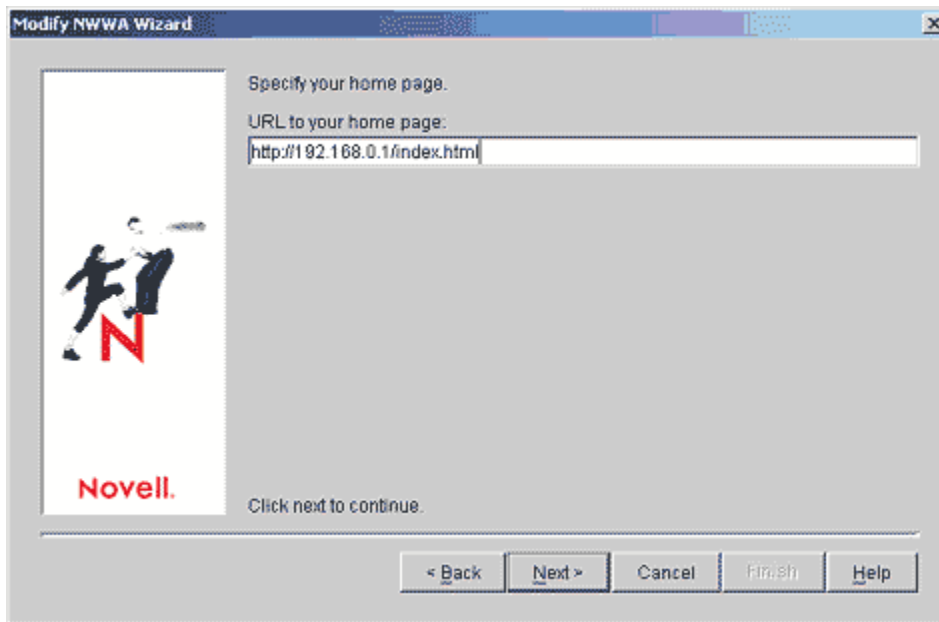
(B) Next you will be presented with a dialog box explaining that you are in the configuration wizard for NetWare Web Access. Click NEXT to continue.



(C) Here you can select the services you would like to enable for Web Access. If there was a services that was not listed in step (A) then you can select it here. After selecting the desired services click NEXT to continue.

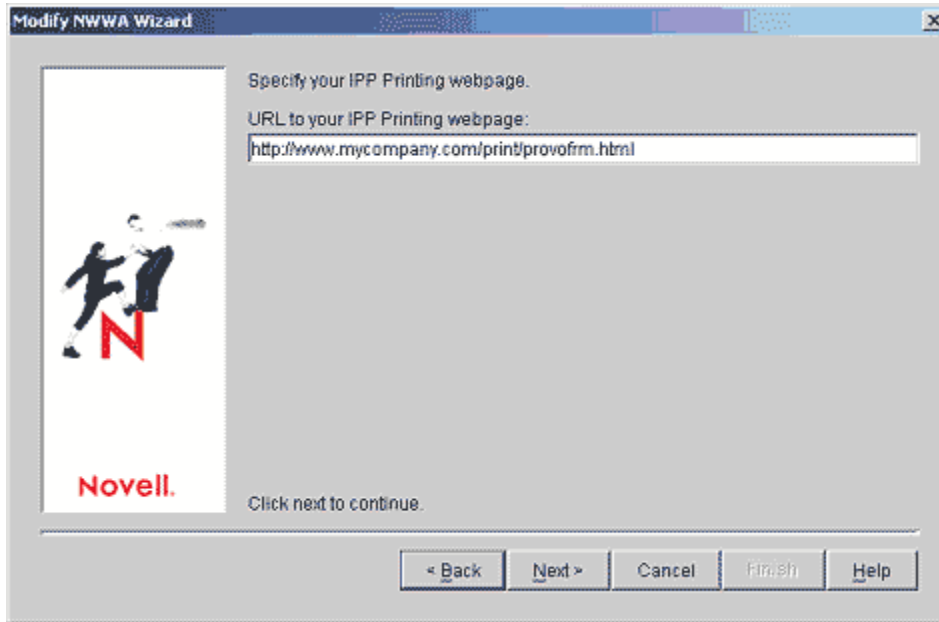


(D) If HOME was selected then the first screen you will come to is your home page configuration. Put in the proper URL to configure your home page.



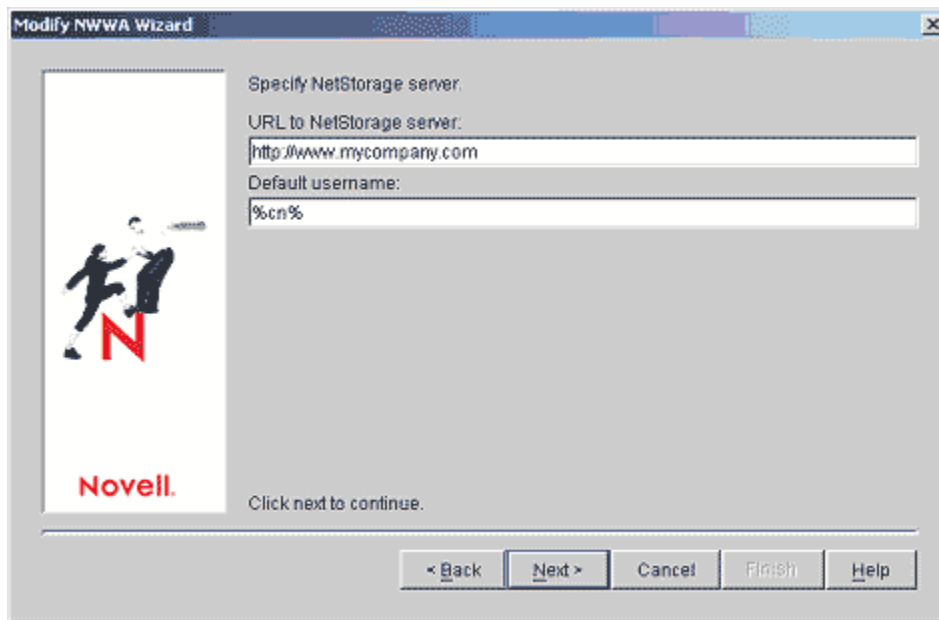
(E) If you selected iPrint then you will be prompted to enter in you IPP Printing Web Page. Click NEXT to continue.

**\*\*NOTE\*\*** You will need to have iPrint already setup at this point.

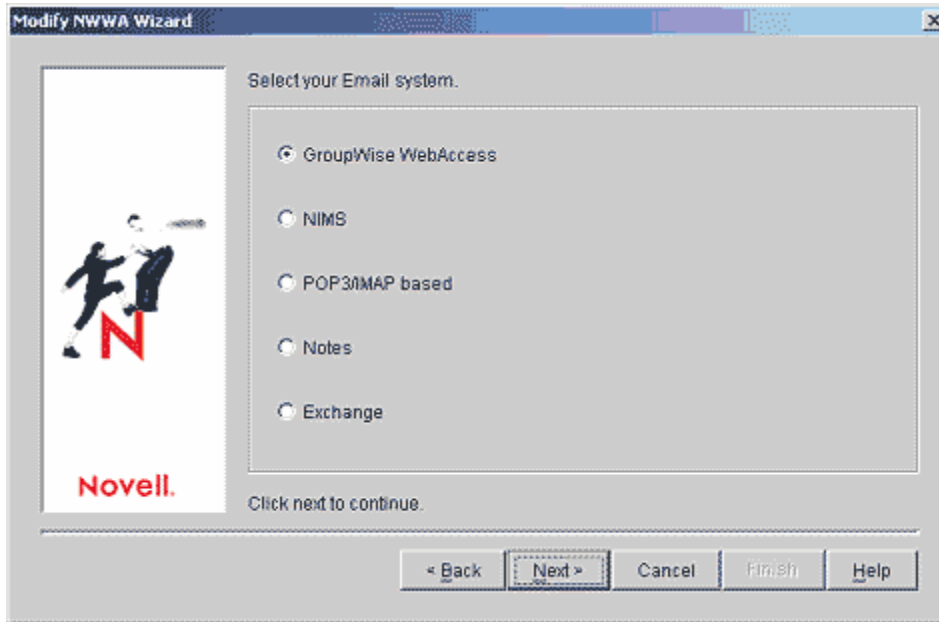


(F) If you selected NetStorage you will be presented with the NetStorage dialog box. You will need to put in the NetStorage URL to access that account.

\*\* NOTE \*\* You will need to have NetStorage already setup at this point.

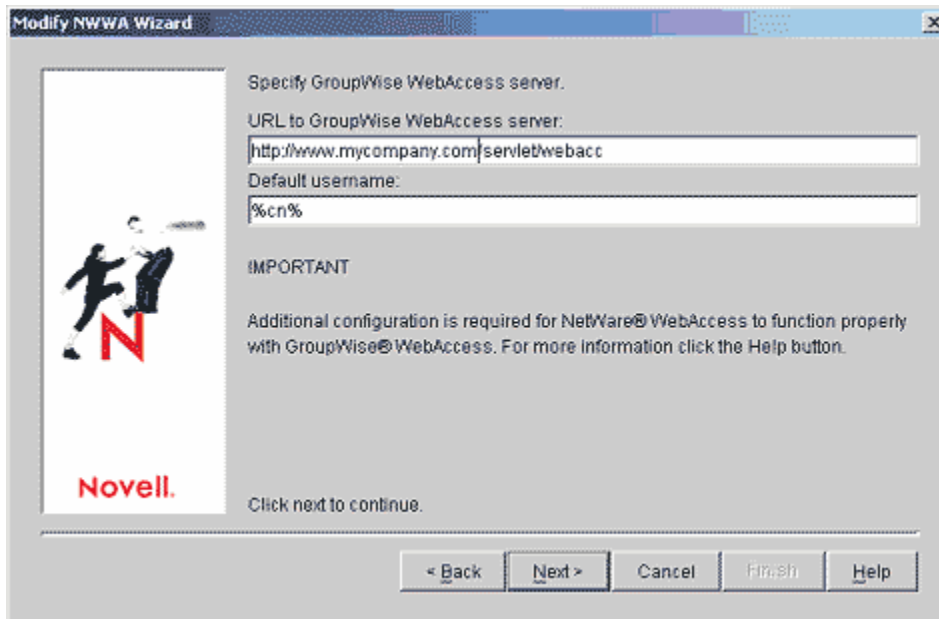


(G) Next will be the type of email system that you have. Here you can select GroupWise WebAccess, NIMS, POP3/IMAP, Lotus Notes, or Microsoft Exchange email system. For GroupWise WebAccess you will select the GroupWise WebAccess option and then select NEXT.

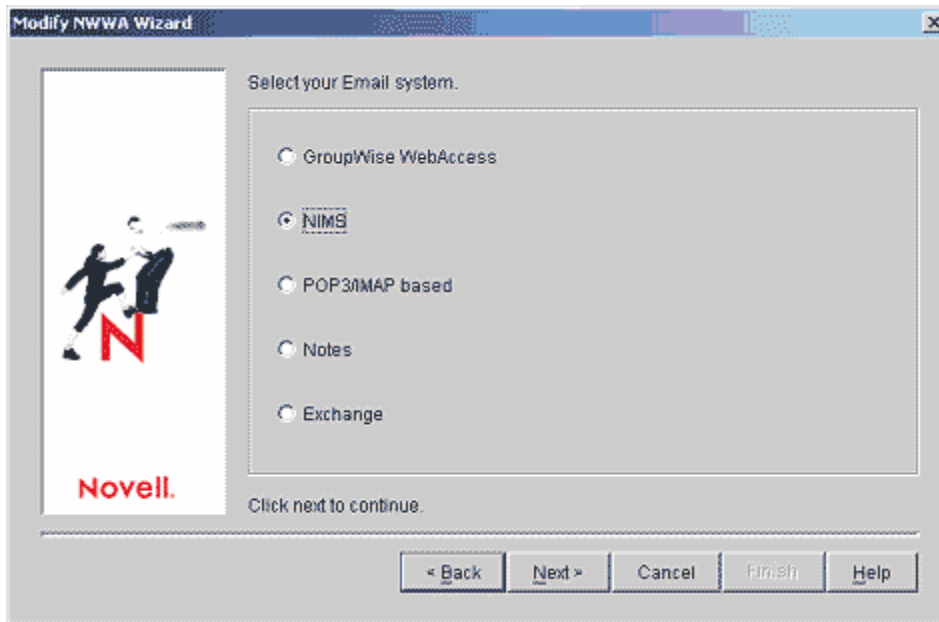


(H) Here you will be presented with the GroupWise WebAccess dialog box. You will need to put in the URL to you GroupWise WebAccess system. You can point to the index page or to the GO page if you use the /servlets/webacc extension as seen in the dialog box below.

**\*\*NOTE\*\*** You will need to have GroupWise WebAccess already setup at this point.

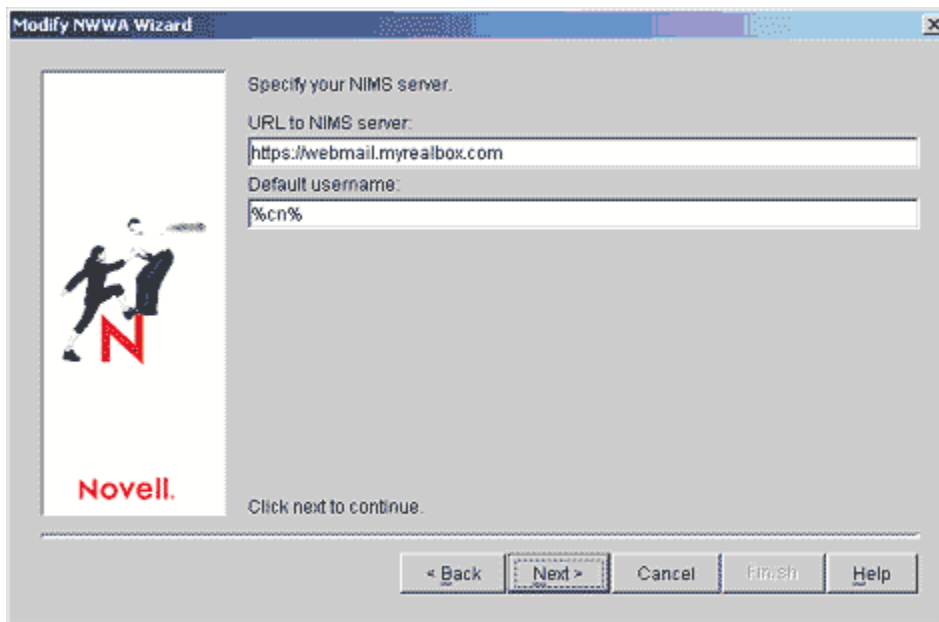


(I) If you desire to use NIMS then you will need to select the NIMS option and click NEXT.

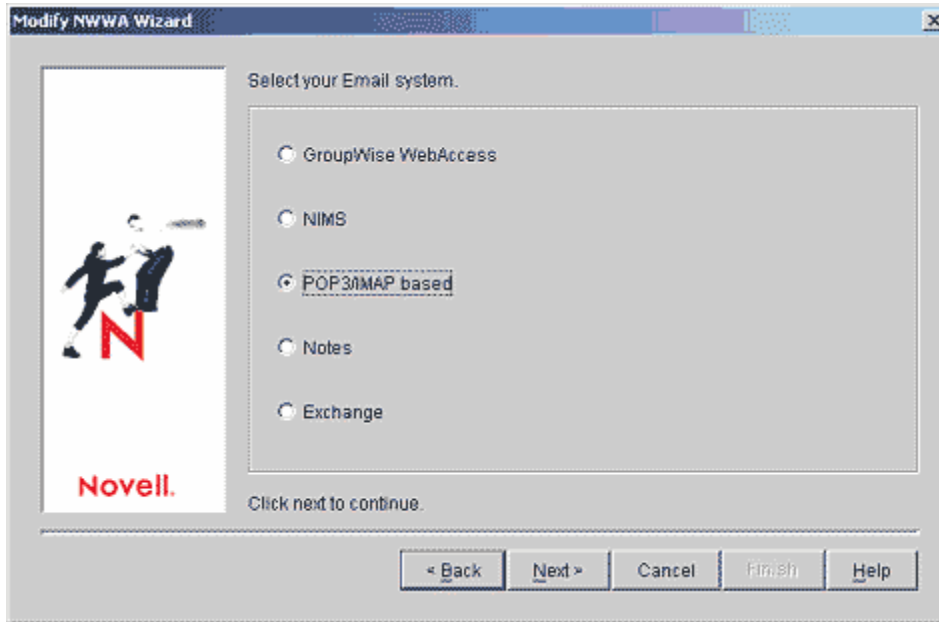


(J) Here you are presented with the NIMS dialog box. You will need to configure the URL to point to the WebMail page of your NIMS server.

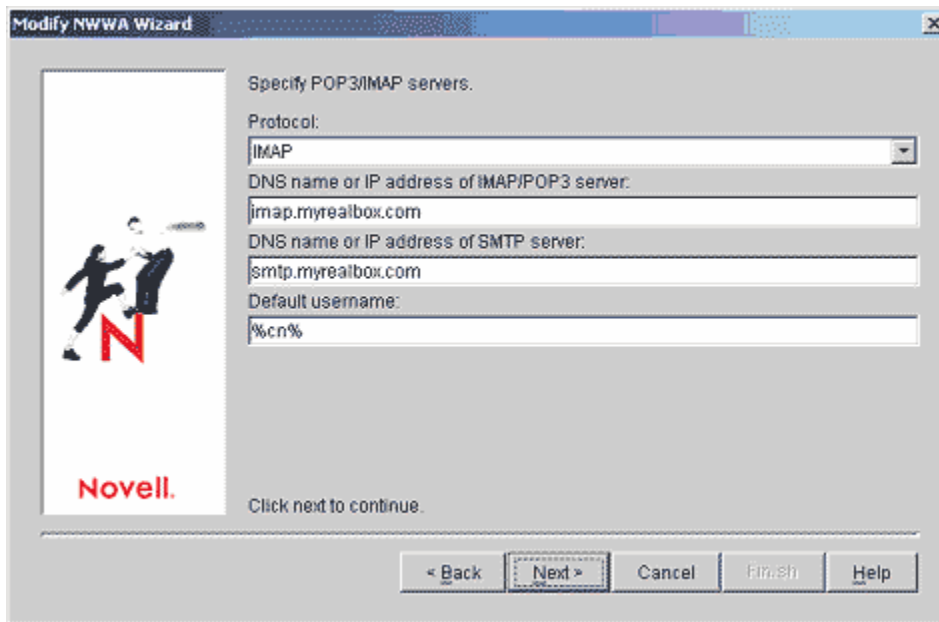
**\*\*NOTE\*\*** You will need to have NIMS already setup at this point.



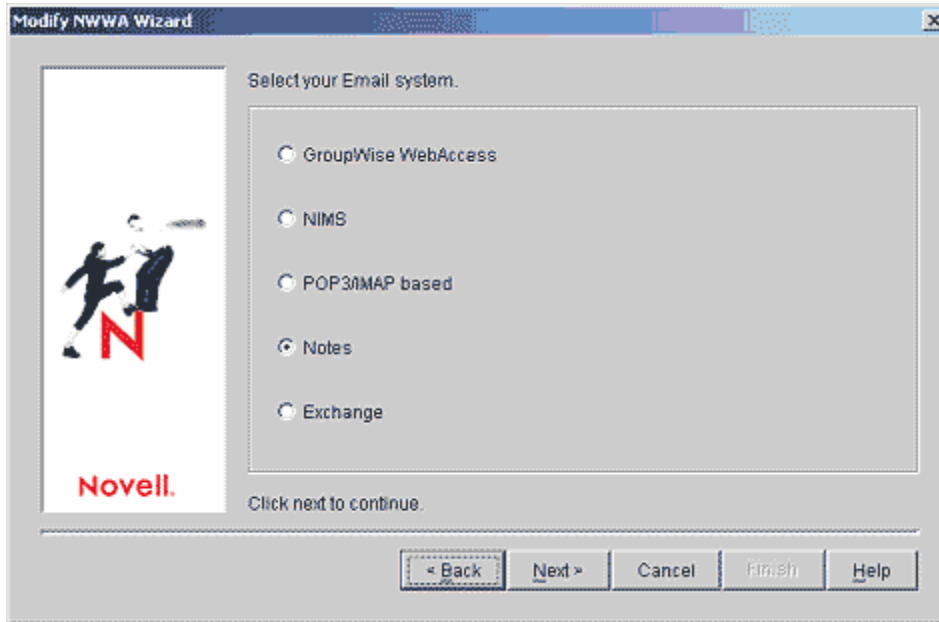
(K) If you have POP3 or IMAP accounts then choose the POP3/IMAP BASED option and click NEXT.



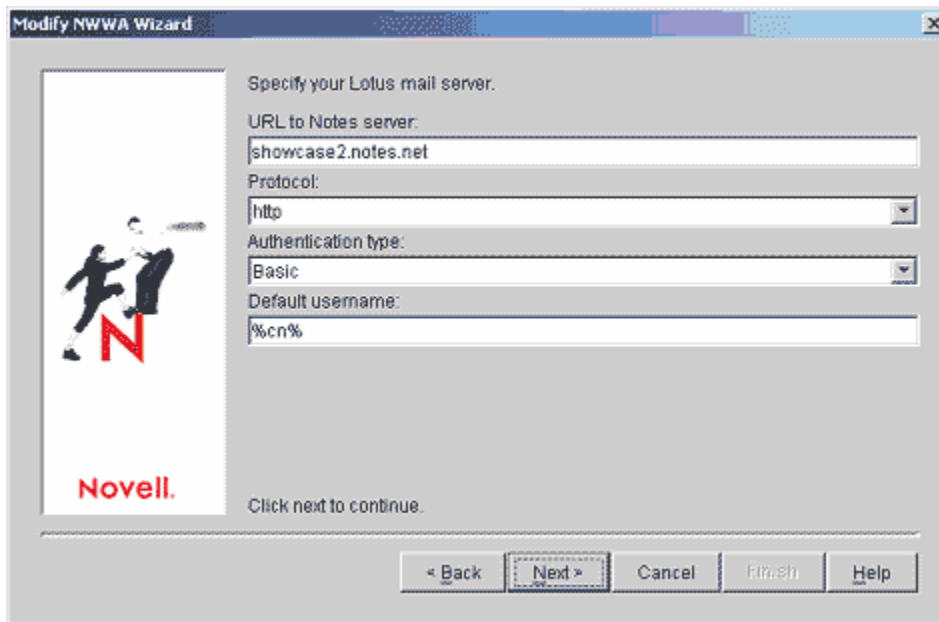
(L) Here you would select the Protocol (IMAP or POP). Then you would put in the DNS name or IP address of the IMAP or POP3 server. You would then put in the IP or DNS name of the SMTP server and click NEXT.



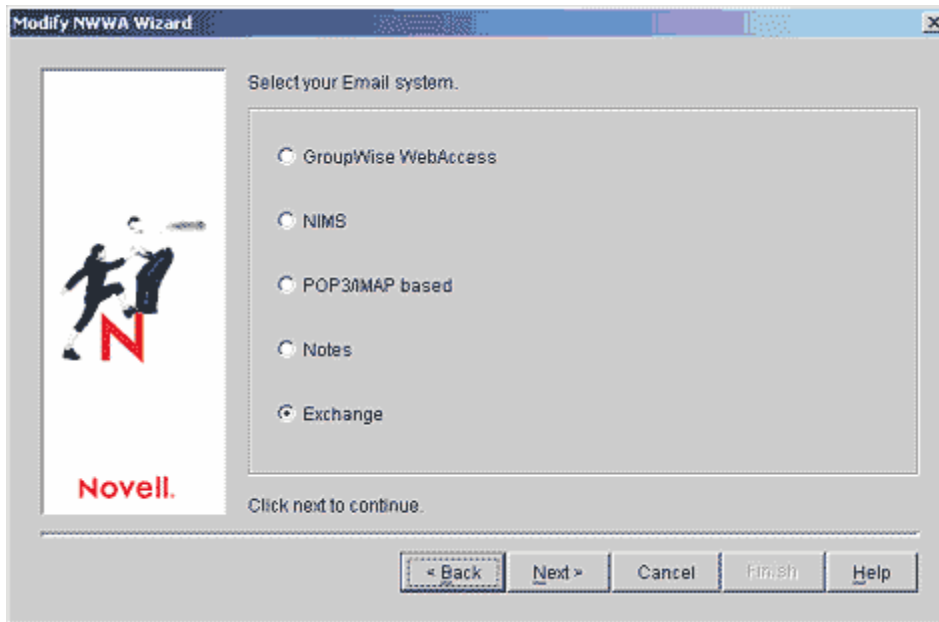
(M) If you have Lotus Notes as your email, you can select NOTES and click NEXT.



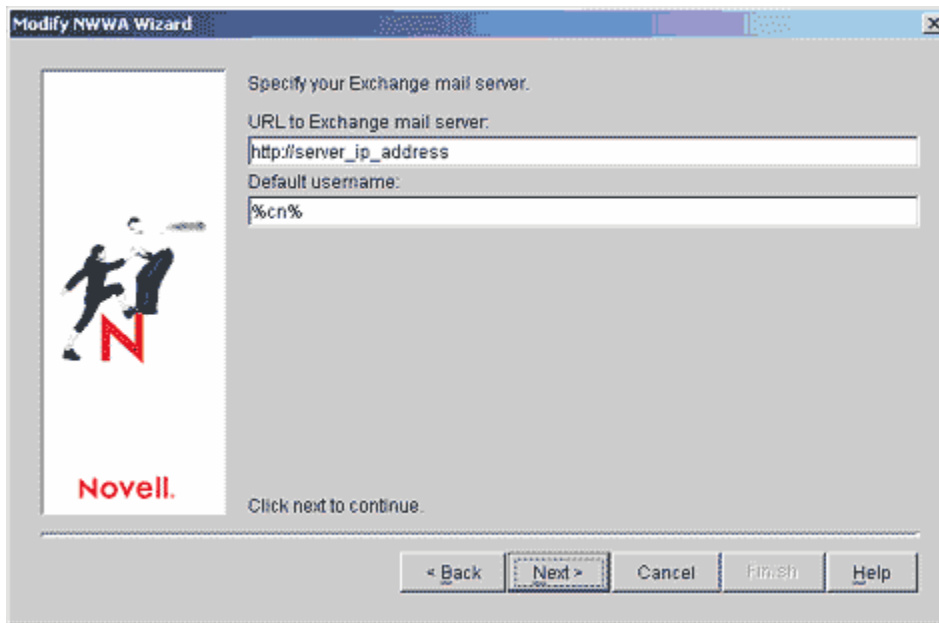
(N) Here you will need to put in the URL of the mail server. The protocol that you are using and the authentication type.



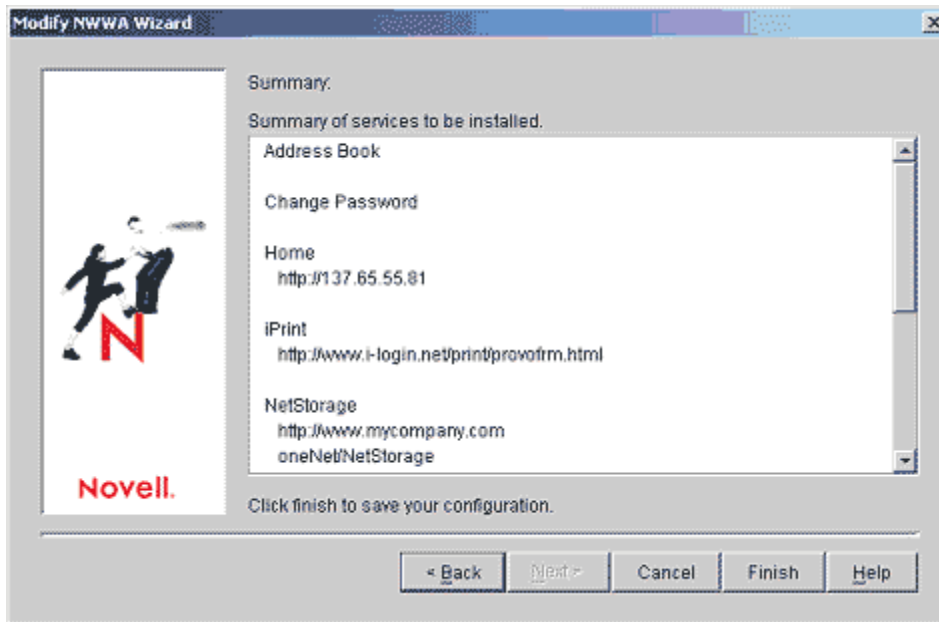
(O) If you are using a Microsoft Exchange server then you will need to select EXCHANGE and then NEXT.



(P) Here you will need to put in the URL that you use to access the Exchange server and then hit NEXT.

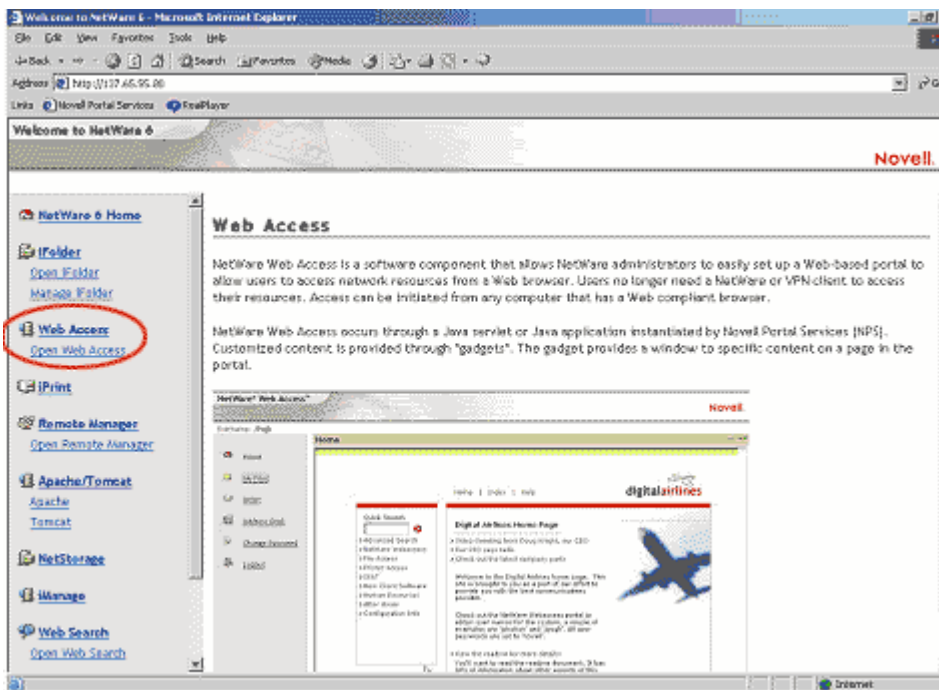


(Q) The last dialog that you will be presented with is the Summary of Services dialog. This will give you another chance to make sure that everything is correct before clicking on finish. If you are satisfied then go ahead and click on FINISH.

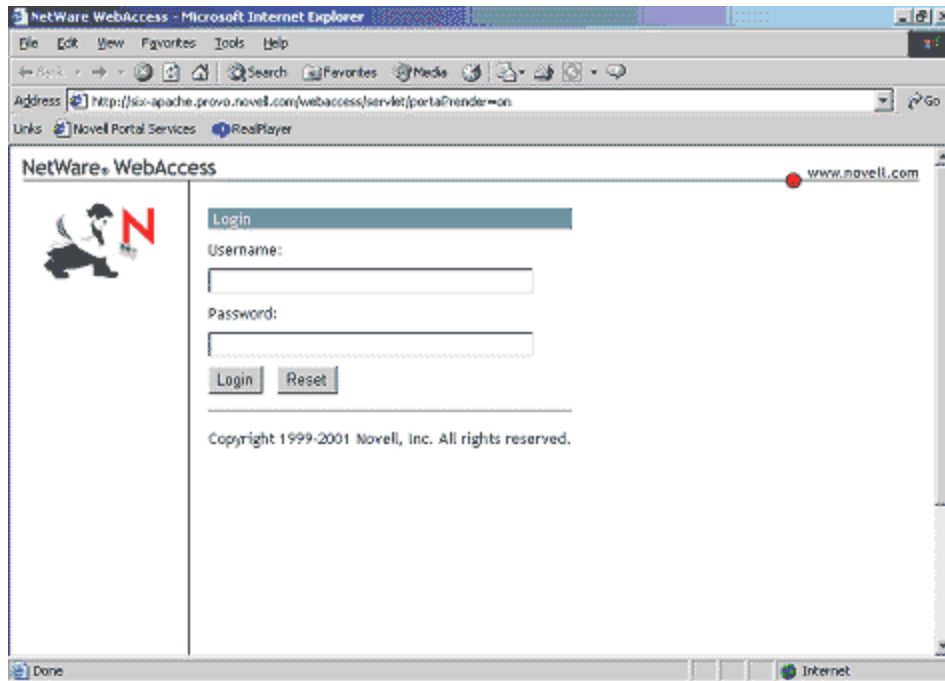


## ACCESSING NETWARE WEB ACCESS

(1) Accessing NetWare Web Access via the NetWare 6 home page.



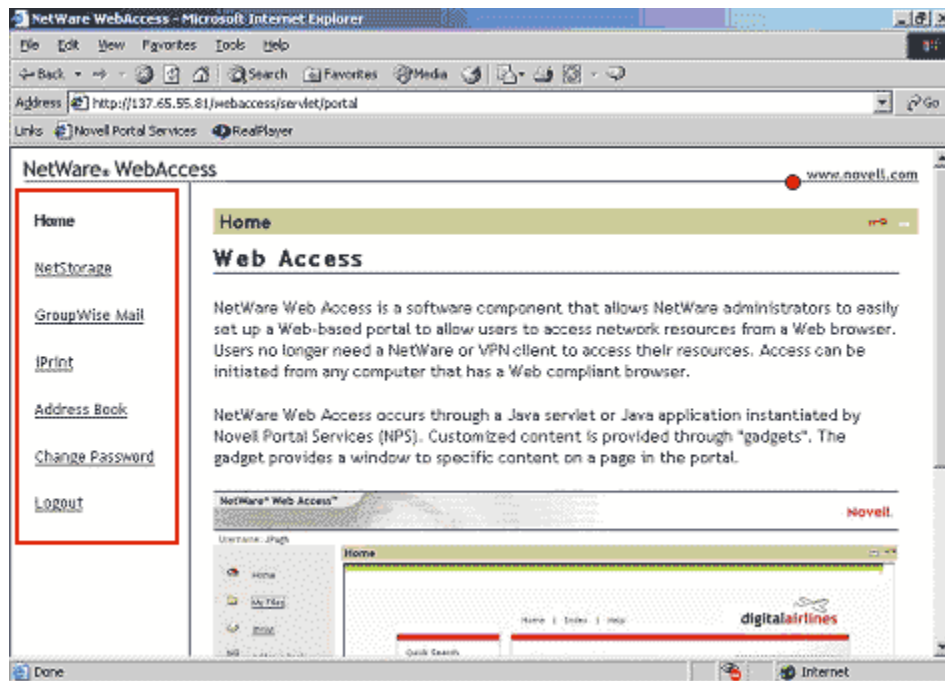
\*\* NOTE \*\*The OPEN WEBACCESS link in this page will send you to the JSP that is run by Apache.



(2) The other option is to access it via URL. An example is shown below.

**https://IPorDomainName/webaccess**

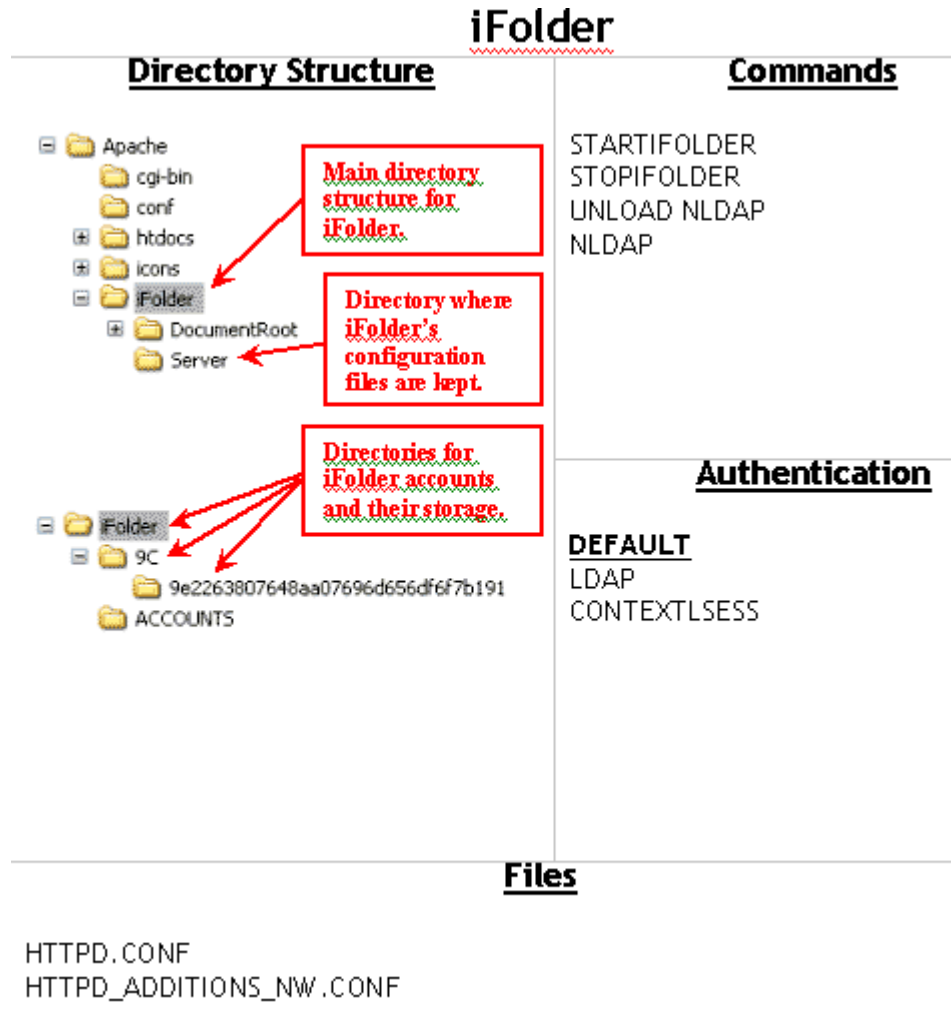
(3) After logging in you will see the Web Access gadgets. Here you can see the individual gadgets listed on the left hand side.



**\*\* IMPORTANT \*\*** If you desire to implement more gadgets than those that ship with NetWare Web Access you can upgrade to Novell Portal Services. As of the shipping of NetWare 6 the following gadgets are available with Novell Portal Services:

- Applet\_Launcher
- Authenticator
- Citrix
- Customizing Exchange
- Groupwise\_Calendar
- Groupwise\_Mail
- HTML
- iFrame
- MoreOverNewsGadget
- NetworkFileGadget
- NNTP
- PhoneBook
- PortalAdmin
- PortalStats
- Registration
- RSS\_News
- ShortCuts
- Topics
- User\_Admin
- WebContent
- WebMailGadget
- XML\_Remote
- Change\_Password
- Notes
- SelfAdmin
- NIMS
- StockTicker
- Survey
- Query

## Part 5: iFolder Administration

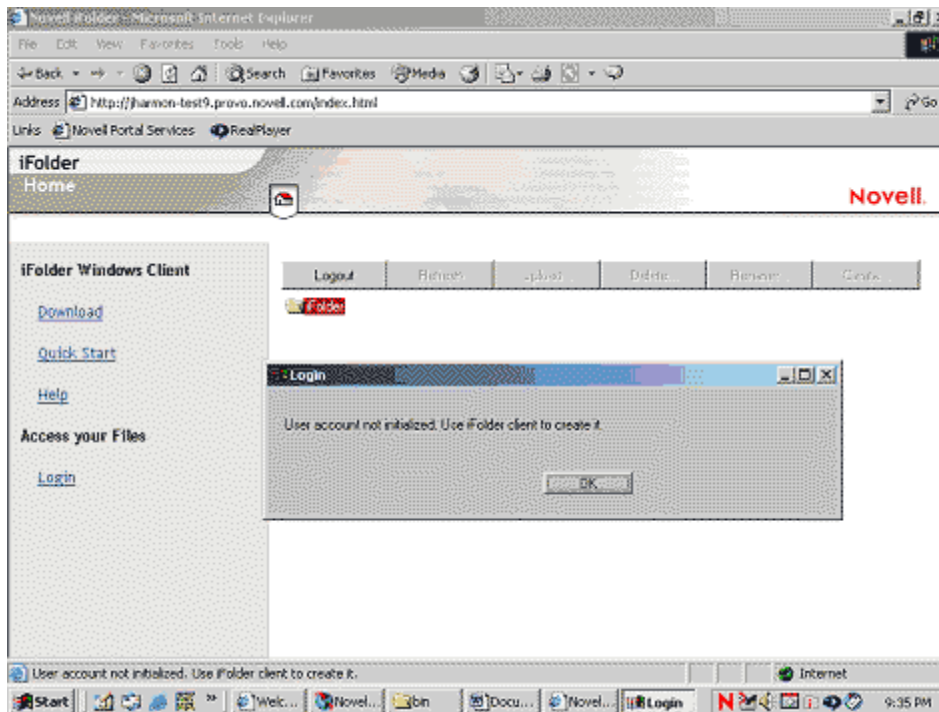


### Creating the iFolder Account

**\*\* NOTE \*\*** After the iFolder installation, the first thing you will need to do is download the iFolder client. You must first login via the client first in order for iFolder to create your account on the server.

**\*\*IMPORTANT\*\*** If you attempt to log in to iFolder via the Web before you have logged in via the client you will receive the following error:

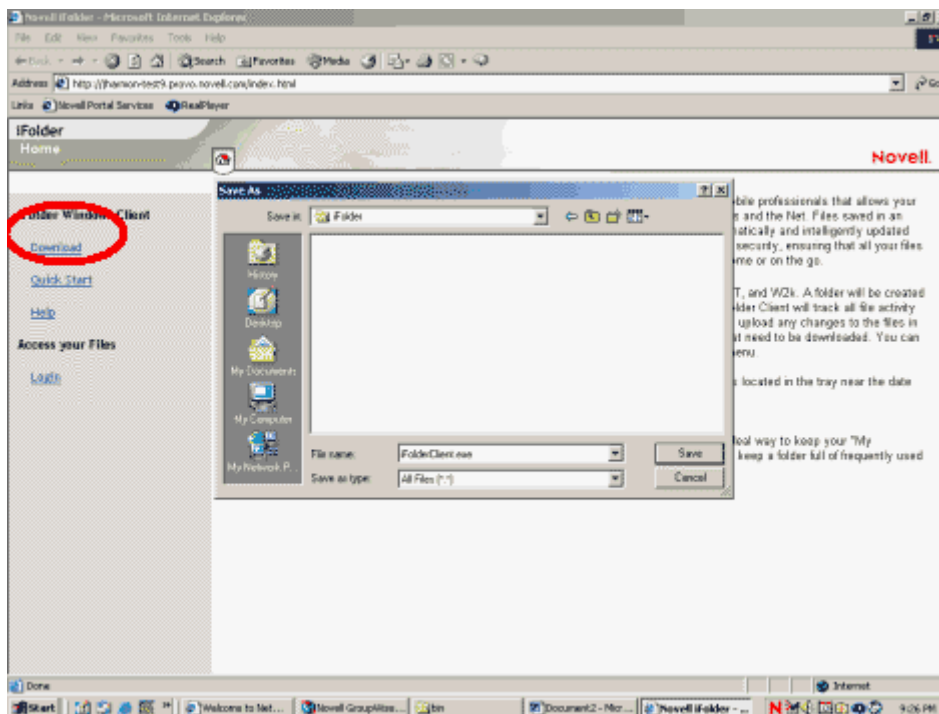
Error: "User account not initialized. Use iFolder client to create it." See graphic below.



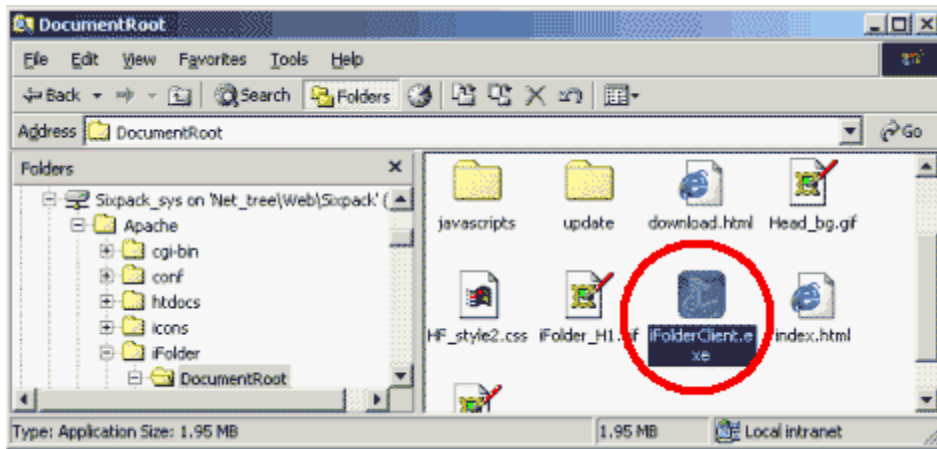
## Obtaining the iFolder Client

You can obtain the iFolderClient.exe one of two ways:

(1) Access iFolder via the Web and click on the download option by going to the IP address or domain name where you installed iFolder. If you chose the single IP address option you will need to add a port number on the end of the URL.

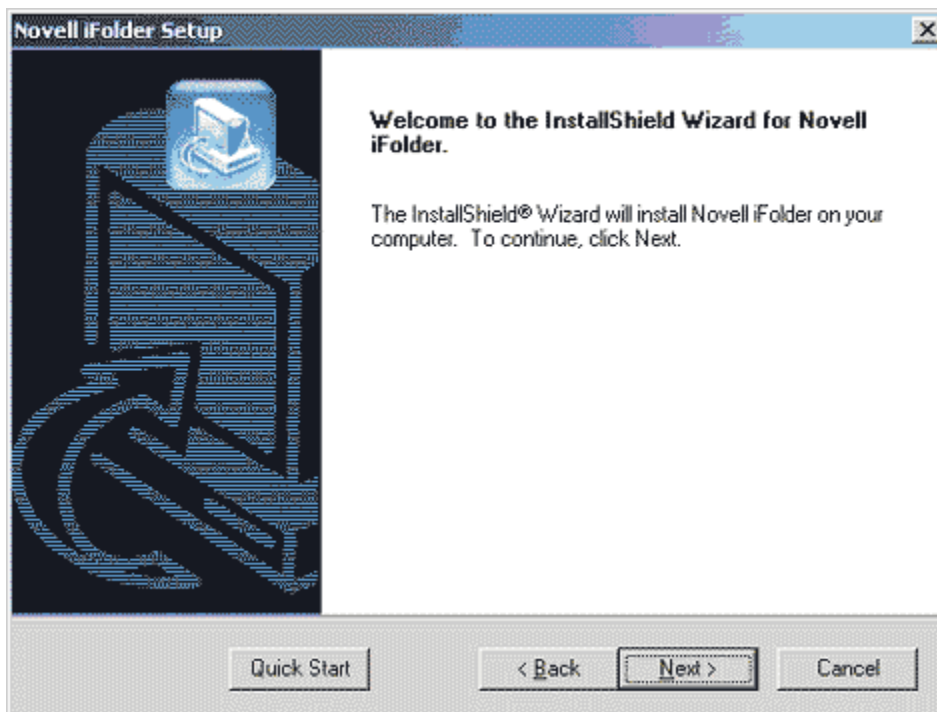


(2) The second option is to access the IFOLDERCLIENT.EXE on the server. Go to SYS:/APACHE/IFOLDER/DOCUMENTROOT/IFOLDERCLIENT.EXE

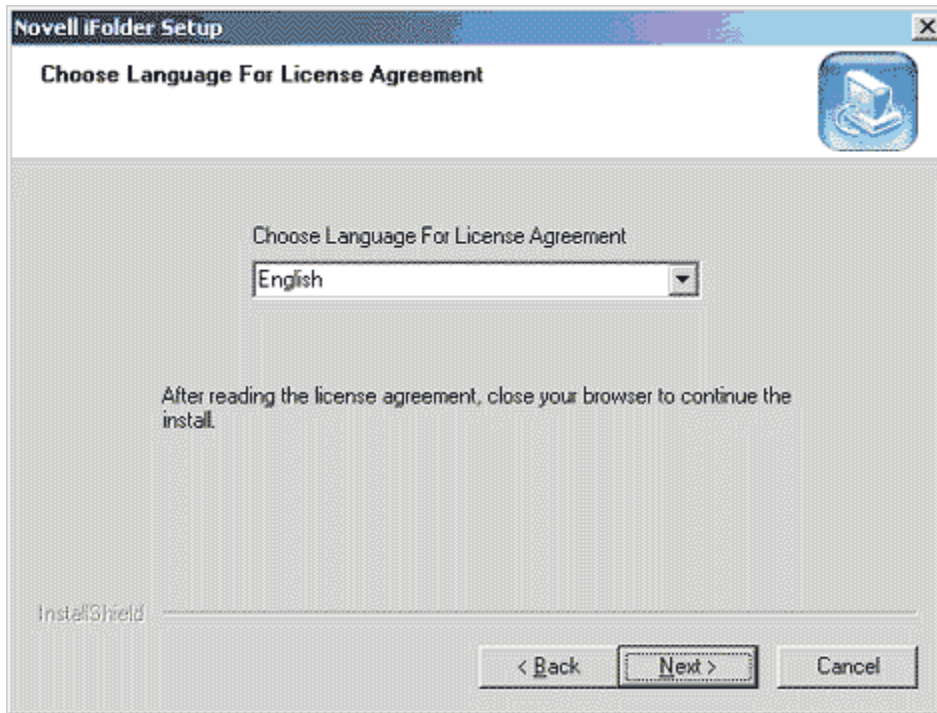


## Installing the iFolder Client

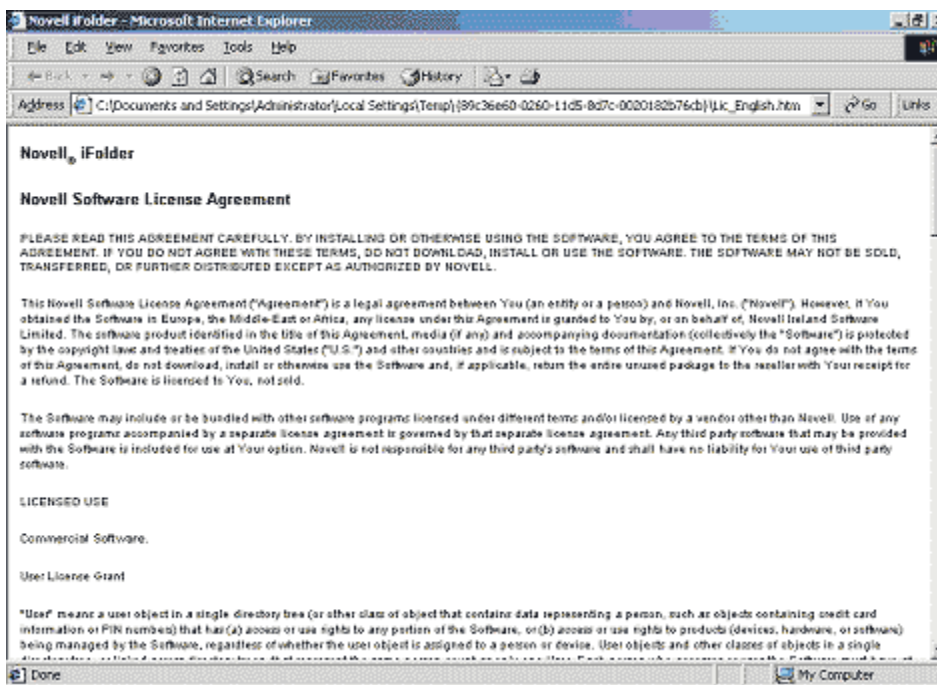
(A) The first screen will be the Welcome screen for the iFolder Client. You get this screen when you double click on the IFOLDERCLIENT.EXE. Press NEXT to continue.



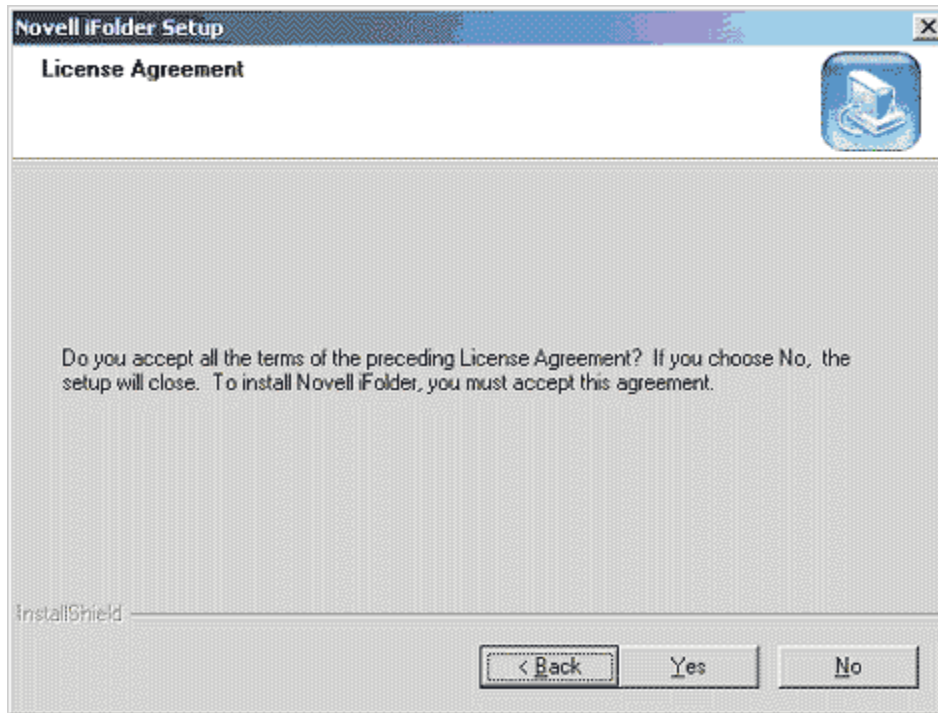
(B) The next screen will ask you what language you want to view the license agreement in. Choose the desired language and then press NEXT.



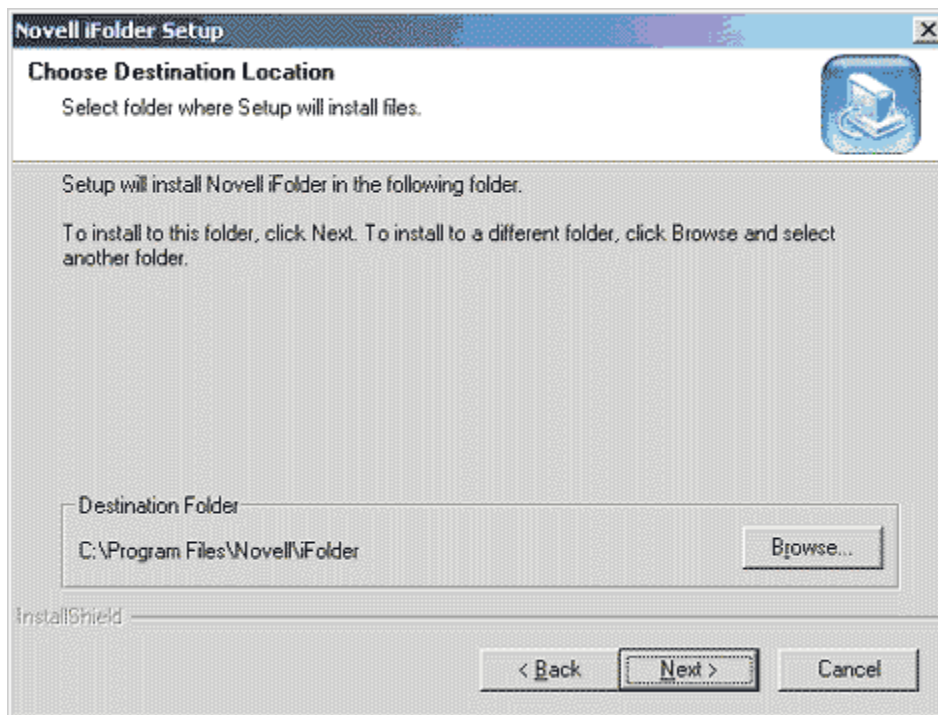
(C) The license agreement will pop up in your browser. After you have read the agreement, close the browser to continue.



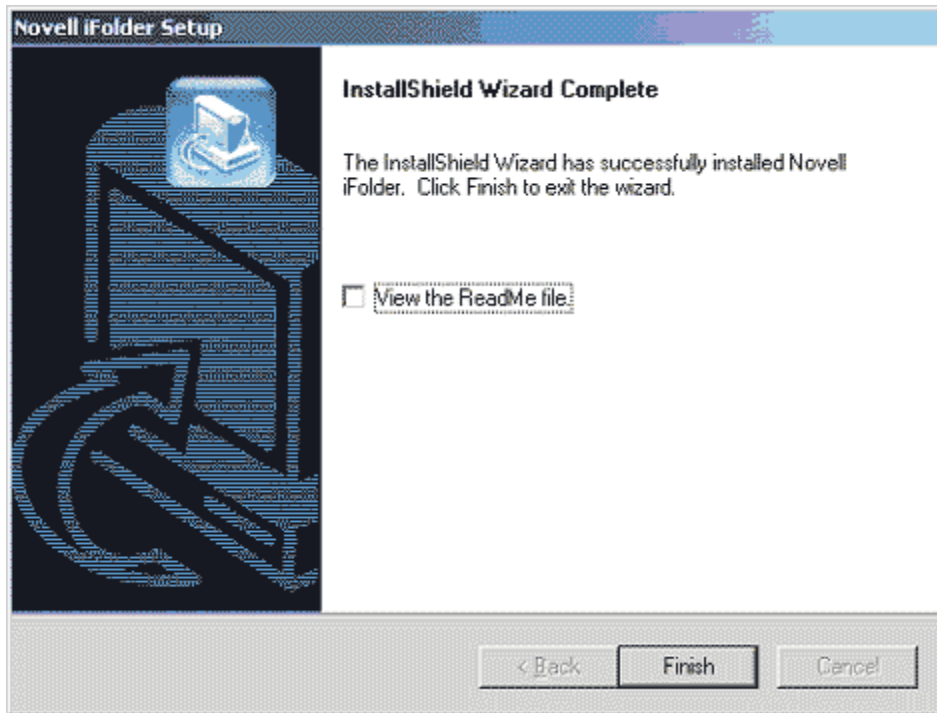
(D) The next screen will ask you whether or not you agree to the agreement. Select YES to continue.



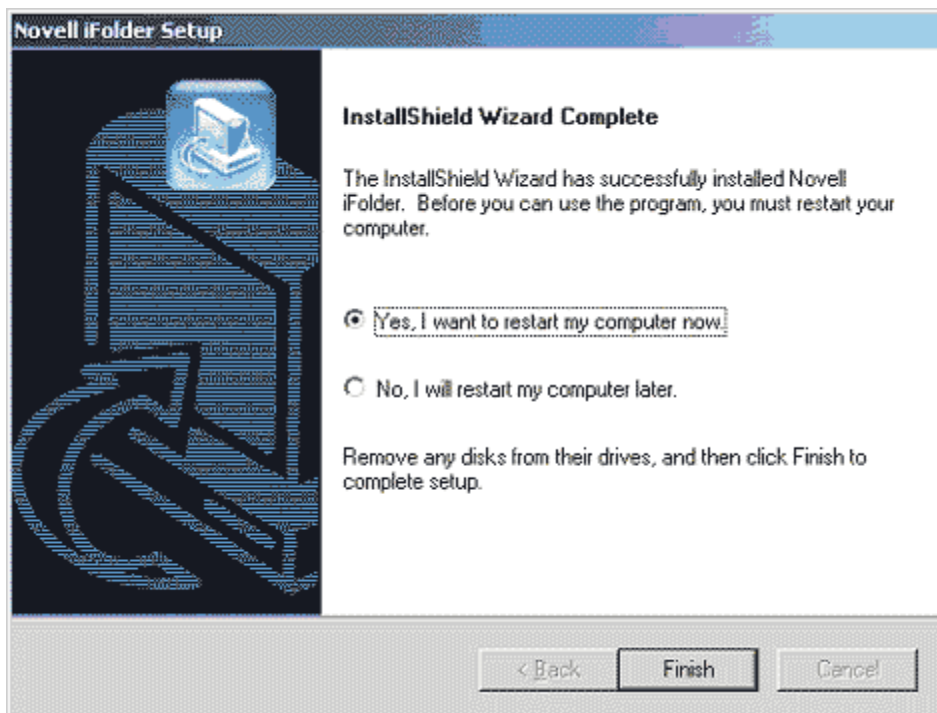
(E) The next screen will ask you where you wish to install iFolder. If you desire to change this option the default location, click on the BROWSE button and choose the desired directory. Click NEXT to continue.



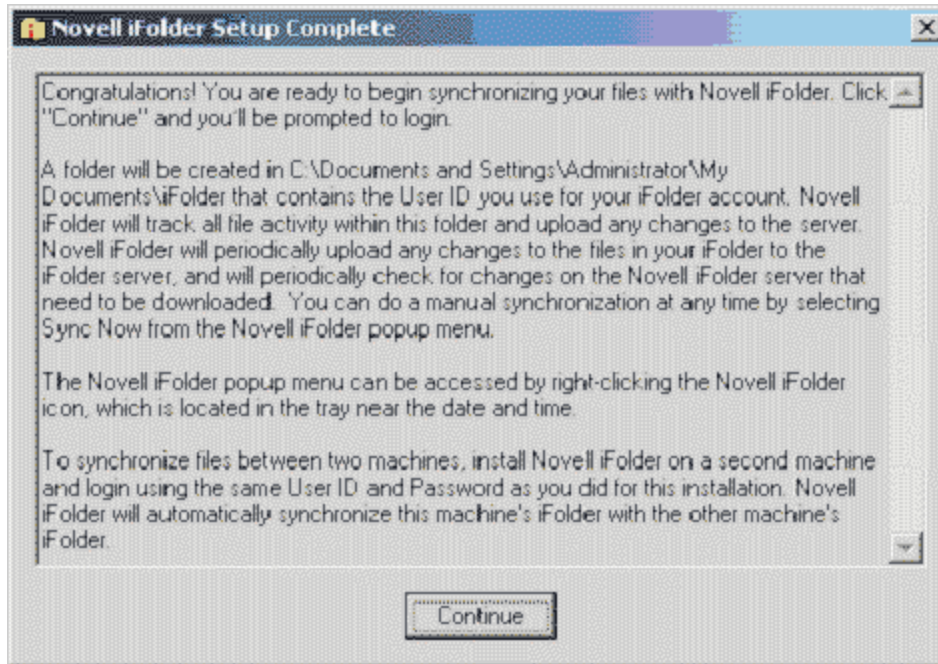
(F) The next screen shows that iFolder has been successfully installed. Click FINISH to exit the wizard.



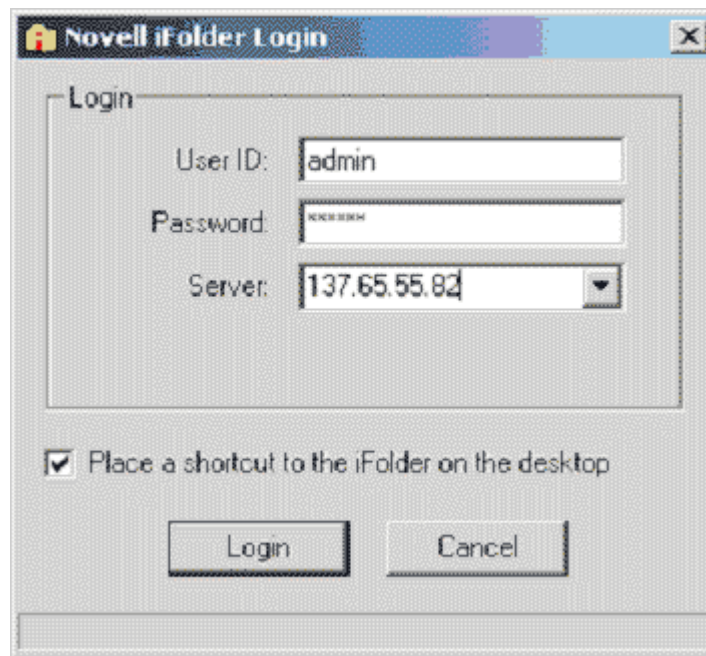
(G) You will now be asked to reboot your workstation. You will need to reboot before you can use iFolder for the first time. Click FINISH to continue.



(H) After your workstation reboots you will receive the Setup Complete Screen. This explains that an account has been created for you under C:\DOCUMENTS AND SETTINGS\ADMINISTRATOR\MY DOCUMENTS\FOLDER\USERACCOUNT\HOME. Click CONTINUE.



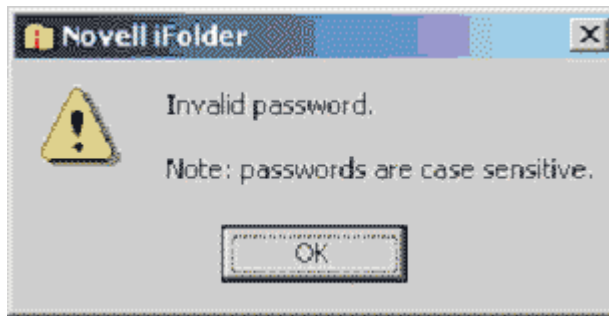
(I) Now you will need to login to iFolder. Enter your NDS user name and password. Put in the server IP address or Domain Name. You will also need to put in the port number if you installed with the single IP address option. If you want to place a shortcut on your workstation for easy access then make sure that option is checked. Click LOGIN to continue.



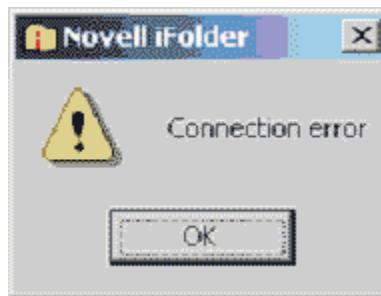
Shortcut Placed on Desktop:



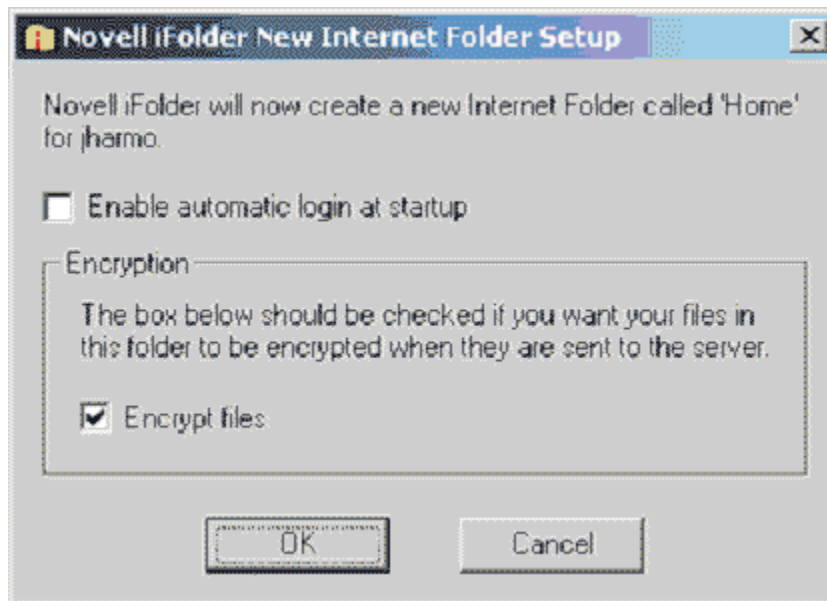
**\*\* NOTE \*\*** If you enter in the wrong password you will receive an invalid password error as seen below.



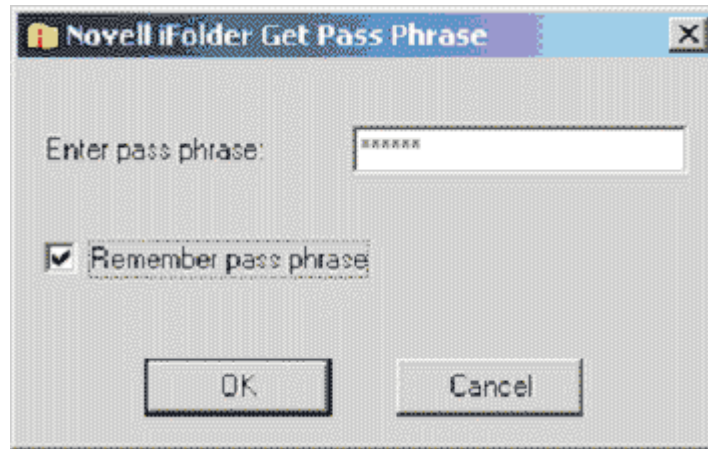
**\*\* NOTE \*\*** If you enter in the wrong server information you will receive a connection error as seen below.



(J) When creating your account for the first time you will be prompted with the following dialog box. Here you have the options of ENABLE AUTOMATIC LOGIN AT STARTUP and ENCRYPT FILES. If you choose the option to ENCRYPT FILES, you will be prompted to enter in a pass phrase as seen in step (K).



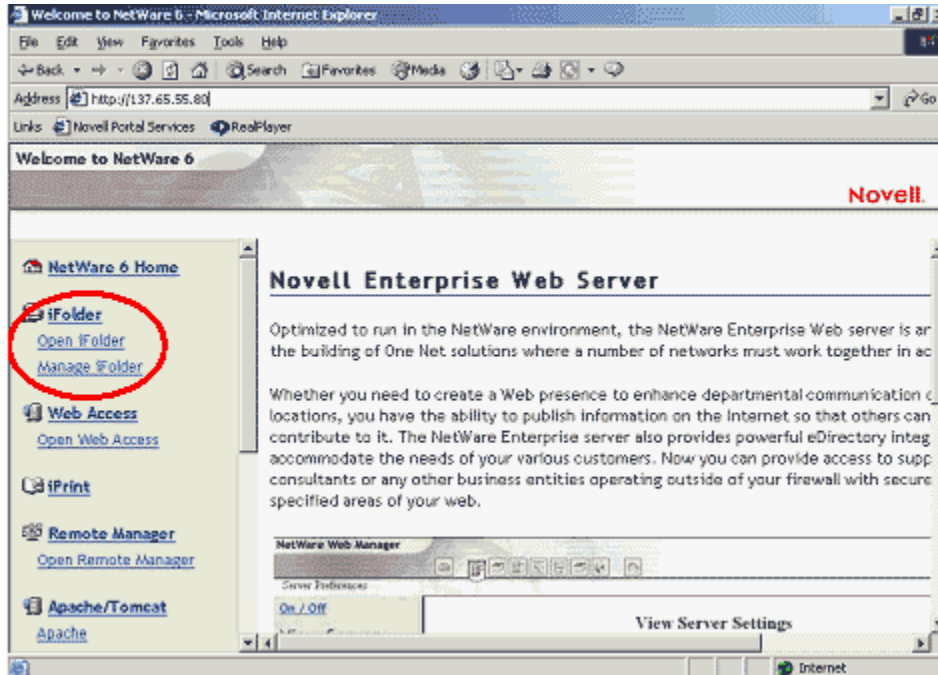
(K) You will then be asked to enter a pass phrase. You can leave it blank if you desire or you can enter one in. If you don't want to be prompted for the pass phrase the next time that you login then you will need to check the REMEMBER PASS PHRASE option.



## iFolder Access and Administration

You will be able to access iFolder in several ways on NetWare 6.

(A) Accessing iFolder via the NetWare 6 Home Page. [iFolder WebApp and iFolder Administration]

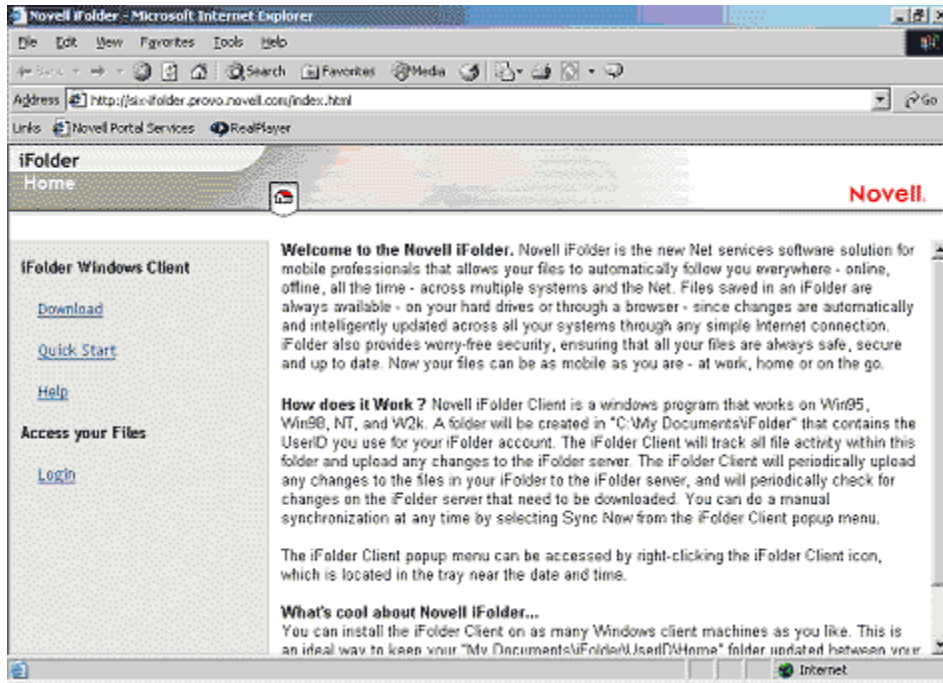


**\*\* NOTE \*\*** From the NetWare 6 home page there are 2 options under iFolder; OPEN IFOLDER and MANAGE IFOLDER. The Open iFolder and Manage iFolder options work the same way, as if you were to access iFolder via an IP address or Domain Name.

(B) Open iFolder (or) Accessing via IP/Domain Name.

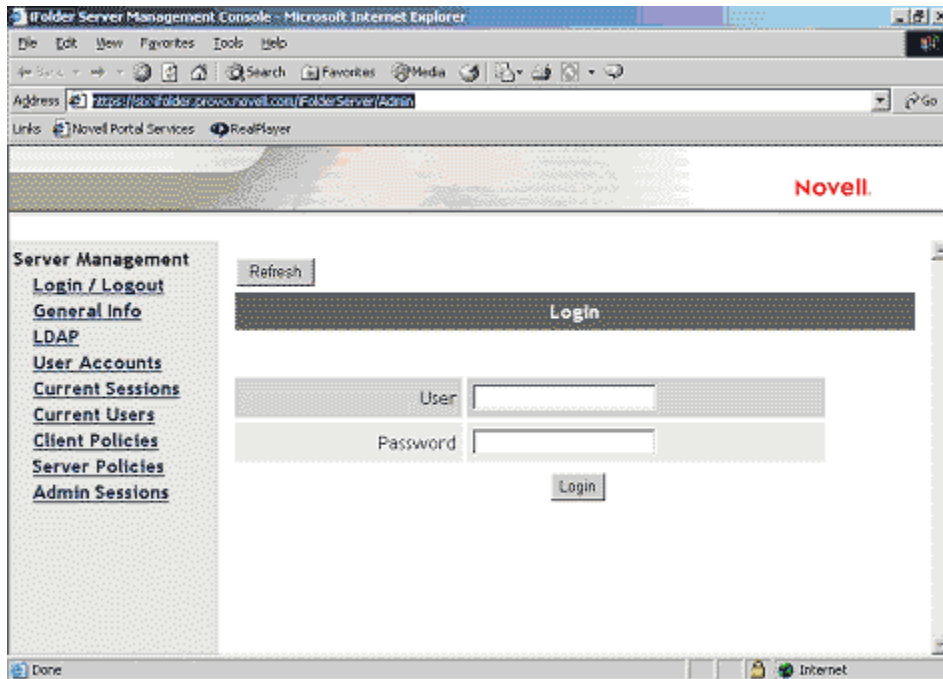
**http://IPorDomainName** - Multiple IP address option.

**http://IPorDomainName:Port** - Single IP address option.



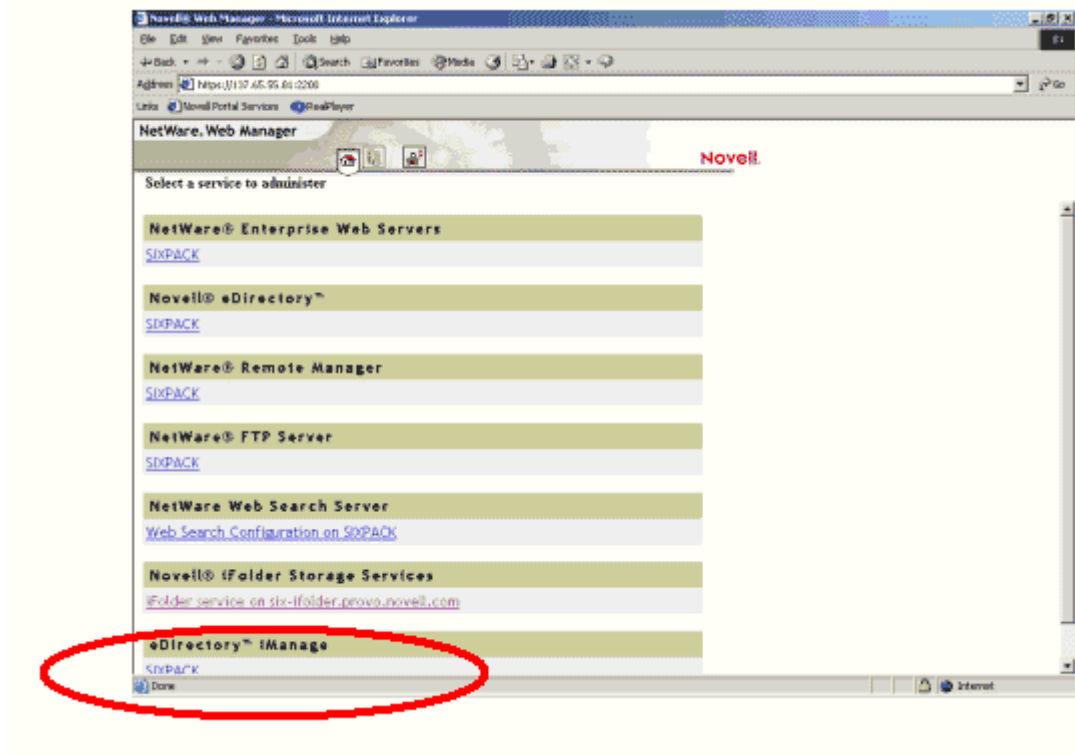
(C) Open iFolder (or) Accessing via IP/Domain Name.

<http://IPorDomainName/iFolderServer/Admin> - Multiple IP address option.  
<http://IPorDomainName:Port /iFolderServer/Admin> - Single IP address option.



(D) Accessing iFolder via the Web Manager

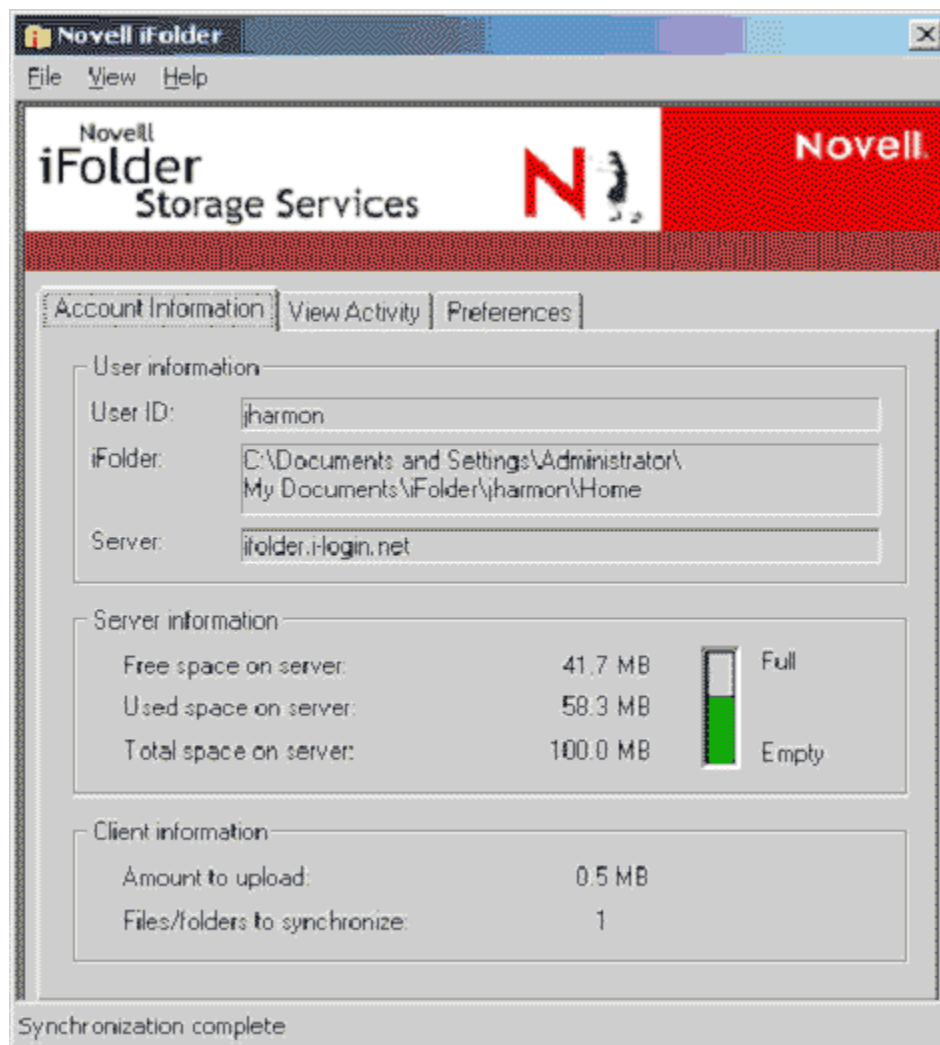
https://IPorDomainName:2200



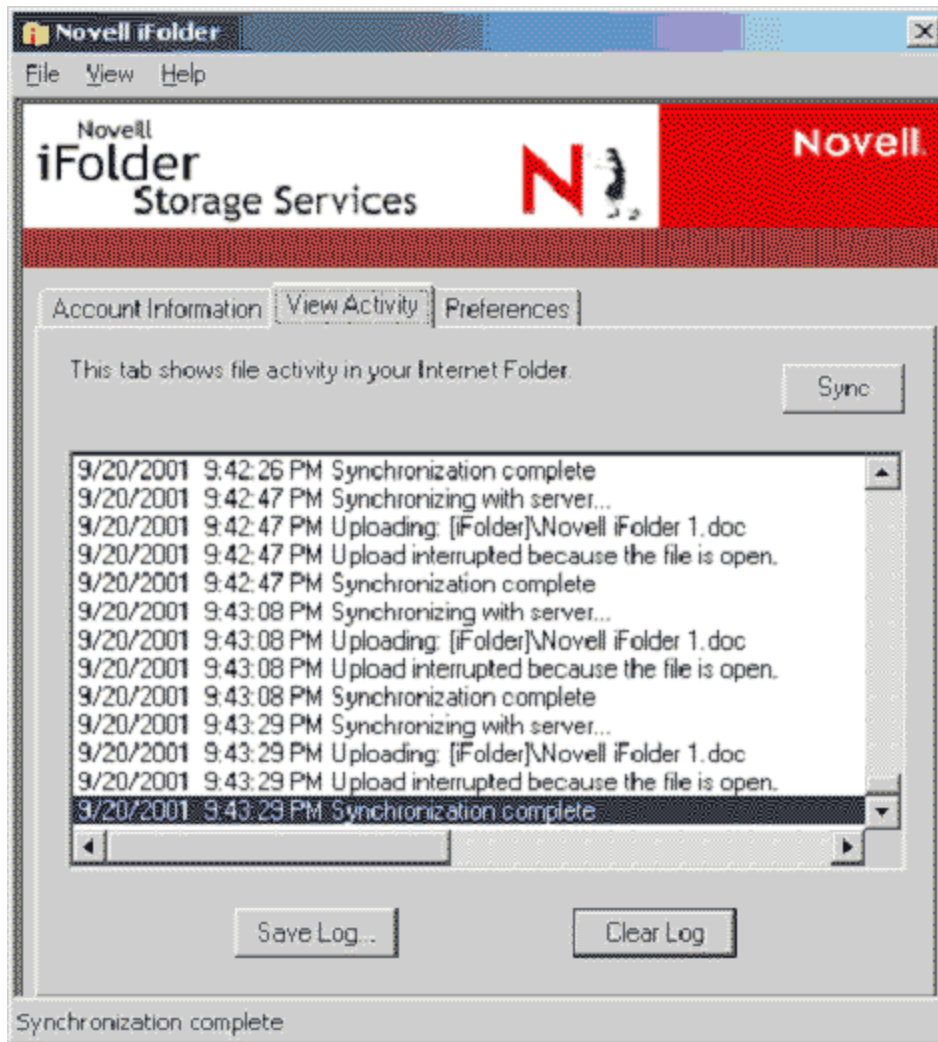
## iFolder Statistics

\*\* NOTE \*\* iFolder statistics can be accessed by double-clicking on the iFolder icon in your system tray.

(1) The Account information tab allows you to see connection, user, and server information.



(2) The View Activity tab shows the activity (such as folder syncs) as they occur on your workstation.



(3) This preferences tab allows you to adjust the information on synchronization, password, and pass phrase information.



## Part 6: Apache

<b>Apache</b>	
<p style="text-align: center;"><b>Directory Structure</b></p>	<p style="text-align: center;"><b>Commands</b></p> <p>NVXADMIN = Brings down Apache NVXADMUP = Brings up Apache</p> <hr/> <p style="text-align: center;"><b>Authentication</b></p> <p><b>DEFAULT</b> NDS CONTEXTLESS</p> <p><b>AVAILABLE</b> LDAP LOCAL DATABASE CONTEXT = NDS OR LDAP FORMAT</p>
<p><b>Files</b></p> <p>ADMINSERV.CONF - Found in the SYS:/APACHE/CONF directory. This file controls the configuration for the Apache Web Server and Web Manager.            ERROR_LOG - Found in SYS:/APACHE/LOGS. Error log for Apache.            ACCESS_LOG - Found in SYS:/APACHE/LOGS. Access log for Apache. (not enabled by default)            MOD_JK.LOG - Found in the SYS:/TOMCAT/33/LOGS directory. Logs communication between Tomcat and Apache.            LOGGER.TXT - Found in the C:/NWSERVER directory on the server. Dumps the Logger Screen to a file if F2 is pressed while at the server on the Logger Screen. This cannot be done from a Remote Session.</p>	

Note: Web Manager installs Tomcat and Apache.

<b>Web Manager</b>	
<p style="text-align: center;"><b><u>Directory Structure</u></b></p> <p style="text-align: center;"><b><u>Commands</u></b></p> <p>JAVA -SHOW = Allow you to see what java processes are running  TOMCAT33 -STOP = Brings down Tomcat process within java. (not recommended)  JAVA -EXT = Brings down all java processes. (recommended)  TOMCAT33 = Brings up Tomcat  NWXADMN = Brings down Apache  NWXADMUP = Brings up Apache</p>	<p style="text-align: center;"><b><u>Authentication</u></b></p> <p><b><u>DEFAULT</u></b> NDS</p> <p><b><u>AVAILABLE IF CONFIGURED</u></b> LDAP</p>
<p><b><u>Files</u></b></p> <p>ADMINSEV.CONF - Found in the SYS:/APACHE/CONF directory. This file controls the configuration for the Apache Web Server and Web Manager.  TOMCAT33.NCF - Found in SYS:/TOMCAT/33/BIN. Startup file for Tomcat on NetWare 6.0.  NWAPPS-WEBADMIN.XML  SERVERS.XML</p>	

## Apache

Please note: Joe's comments are in **Red**.

ServerType standalone

ServerRoot "sys:/apache"

**This is the server root. Therefore, this path is assumed by Apache.**

PidFile logs/httpd.pid

ScoreBoardFile logs/apache\_status

Timeout 300

KeepAlive On

MaxKeepAliveRequests 100

KeepAliveTimeout 15

ThreadsPerChild 50

ThreadStackSize 81920

MaxRequestsPerChild 0

**This section allows us to tune the performance of the Apache Web Server.**

LoadModule lcgimodule modules/mod\_lcgimodule.nlm

LoadModule nds\_auth\_module modules/mod\_nds.nlm

LoadModule tls\_module modules/mod\_tls.nlm

**Novell has added in their own modules. If a third party module has been added to Apache then the module must be manually loaded. Here we have LCGI, NDS, and TLS.**

ServerAdmin someone@somewhere.com

**This directive allows us to display the admin's email address within an error message.**

ServerName jharmon-test4.provo.novell.com

**The ServerName can be the domain name or the IP address. If it is the domain name then it will need to be registered.**

```
# Start Apache Services Port Configuration
Port 80
Listen 137.65.55.77:80
```

**This listen statement allows us to listen on more than one IP address or port. The criteria here is that the IP address needs to be bound to the server.**

```
<IfModule mod_tls.c>
```

```
    SecureListen 137.65.55.77:443 "SSL CertificateDNS"
```

**SecureListen is for our SSL. If SSL is not running then mod\_tls will not load.**

```
</IfModule>
```

```
# End Apache Services Port Configuration
```

```
DocumentRoot "sys:/apache/nwdocs"
```

**SYS:/APACHE/NWDOC is the main document root for publishing our web content on NetWare 6.**

```
<Directory "/">
```

```
    Options FollowSymLinks
```

```
    AllowOverride None
```

```
    Order deny,allow
```

```
    deny from all
```

**There is an issue with leaving these two directives. They were left over to fix an existing Apache problem. If left this way the customer can get a forbidden under certain circumstances.**

```
</Directory>
```

```
<Directory "sys:/apache/nwdocs">
```

```
    Options Indexes FollowSymLinks MultiViews
```

```
    AllowOverride None
```

```
    Order allow,deny
```

```
    allow from all
```

```
</Directory>
```

**This directive is allowing public access to the document root.**

```
DirectoryIndex index.html
```

**This directive tells us what file we will try and pick up by default. If it can't find this file then it will show a directory listing instead.**

```
AccessFileName .htaccess
```

```
<Files ~ "^.ht">
```

```
    Order allow,deny
```

```
    Deny from all
```

```
</Files>
```

```
UseCanonicalName Off
```

```
TypesConfig conf/mime.types
```

```
DefaultType text/plain
```

```
HostnameLookups Off
```

```
ErrorLog logs/error_log
```

```
LogLevel warn
```

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" combined
```

```
LogFormat "%h %l %u %t \"%r\" %>s %b common
```

```
LogFormat "%{Referer}i -> %U" referer
```

```
LogFormat "%{User-agent}i" agent
```

**This section is defining our error logs for the main Apache server.**

```
AccessFileName .htaccess
```

```
# Uncomment the following line to turn on access logging
```

```
# CustomLog logs/access_log common
```

**This section is for our access log on the Apache server.**

ServerSignature On

Alias /icons/ "sys:/apache/icons/"

Alias /icons "sys:/apache/icons/"

**This section shows an example of an Alias or Virtual Directory. This option is used if we need to access information outside of the document root.**

```
<Directory "sys:/apache/icons">
```

```
Options Indexes MultiViews
```

```
AllowOverride None
```

```
Order allow,deny
```

```
allow from all
```

```
</Directory>
```

**Here we are granting public access to the icons directory structure. The two main portions of this are:**

**Order allow,deny**

**Allow from all**

IndexOptions FancyIndexing

**The IndexOptions directive defines how we are going to display our index of files and folders if we are unable to pickup an index.html file.**

```
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
```

```
AddIconByType (TXT,/icons/text.gif) text/*
```

```
AddIconByType (IMG,/icons/image2.gif) image/*
```

```
AddIconByType (SND,/icons/sound2.gif) audio/*
```

```
AddIconByType (VID,/icons/movie.gif) video/*
```

```
AddIcon /icons/binary.gif .bin .exe
```

```
AddIcon /icons/binhex.gif .hqx
```

```
AddIcon /icons/tar.gif .tar
```

```
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
```

```
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
```

```
AddIcon /icons/a.gif .ps .ai .eps
```

```
AddIcon /icons/layout.gif .html .shtml .htm .pdf
```

```
AddIcon /icons/text.gif .txt
```

```
AddIcon /icons/c.gif .c
```

```
AddIcon /icons/p.gif .pl .py
```

```
AddIcon /icons/f.gif .for
```

```
AddIcon /icons/dvi.gif .dvi
```

```
AddIcon /icons/uuencoded.gif .uu
```

```
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
```

```
AddIcon /icons/tex.gif .tex
```

```
AddIcon /icons/bomb.gif core
```

```
AddIcon /icons/back.gif ..
```

```
AddIcon /icons/hand.right.gif README
```

```
AddIcon /icons/folder.gif ^^DIRECTORY^^
```

```
AddIcon /icons/blank.gif ^^BLANKICON^^
```

```
DefaultIcon /icons/unknown.gif
```

```
ReadmeName README
```

```
HeaderName HEADER
```

```
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
```

```
AddEncoding x-compress Z
```

```
AddEncoding x-gzip gz tgz
```

```
AddLanguage da .dk
```

```
AddLanguage nl .nl
```

```
AddLanguage en .en
```

```
AddLanguage et .ee
```

```
AddLanguage fr .fr
```

```
AddLanguage de .de
```

```
AddLanguage el .el
```

```
AddLanguage he .he
```

```
AddCharset ISO-8859-8 .iso8859-8
```

```
AddLanguage it .it
```

```
AddLanguage ja .ja
```

```
AddCharset ISO-2022-JP .jis
```

```
AddLanguage kr .kr
```

```
AddCharset ISO-2022-KR .iso-kr
```

```

AddLanguage no .no
AddLanguage pl .po
AddCharset ISO-8859-2 .iso-pl
AddLanguage pt .pt
AddLanguage pt-br .pt-br
AddLanguage ltz .lu
AddLanguage ca .ca
AddLanguage es .es
AddLanguage sv .se
AddLanguage cz .cz
AddLanguage ru .ru
AddLanguage tw .tw
AddCharset Big5 .Big5 .big5
AddCharset WINDOWS-1251 .cp-1251
AddCharset CP866 .cp866
AddCharset ISO-8859-5 .iso-ru
AddCharset KOI8-R .koi8-r
AddCharset UCS-2 .ucs2
AddCharset UCS-4 .ucs4
AddCharset UTF-8 .utf8
LanguagePriority en da nl et fr de el it ja kr no pl pt pt-br ru ltz ca es sv tw
AddType application/x-tar .tgz
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\..0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4\..0" force-response-1.0
BrowserMatch "Java/1\..0" force-response-1.0
BrowserMatch "JDK/1\..0" force-response-1.0

```

**This whole section defines what icons will show up for certain extensions. It also defines how the web server is to handle this information.**

```

#
# The following line instructs Apache to load the jk module
#
LoadModule jk_module modules/mod_jk.nlm
Mod_JK is our module that integrates Apache and Tomcat.

```

```

JkWorkersFile "SYS:/tomcat/33/conf/jk/nwworkers.properties"
Configuration file for the MOD_JK module. JkLogFile "SYS:/tomcat/33/logs/mod_jk.log"
Log file that records the information between Apache and Tomcat.

```

```

#
# Log level to be used by mod_jk
#
JkLogLevel error
JKLogLevel and LogLevel are directives for what level of debugging that we want to capture. There are three main levels.

```

- (1) warn**
- (2) error**
- (3) debug**

```

#
# Root context mounts for Tomcat
#
JkMount /*.jsp ajp13
JkMount /servlet/* ajp13
jkMount directives allow us to define Tomcat information within Apache.

```

```

# NetWare Web Manager config starts

Listen 137.65.55.77:2211
<IfModule mod_tls.c>
    SecureListen 137.65.55.77:2200 "SSL CertificateDNS"
# Listen 137.65.55.77:2200
</IfModule>

```

```
<VirtualHost jharmon-test4.provo.novell.com:2200>
```

**The virtual host section allows us to apply specific directives to a specific host rather than the whole web server. There are two different types of Virtual Host. IP Based Virtual Hosts and Domain Based Virtual Hosts.**

```
ServerAdmin mymailname@mail.server
DocumentRoot sys:/webapps/WebAdmin
ServerName jharmon-test4.provo.novell.com
ErrorLog sys:/novonyx/suitespot/admin-serv/logs/errors.txt
CustomLog sys:/novonyx/suitespot/admin-serv/logs/access.txt common
DefaultType text/html
BrowserMatch "MSIE" nokeepalive downgrade-1.0 force-response-1.0
```

```
<Directory "/">
Options FollowSymLinks
AllowOverride None
Order deny,allow
deny from all
</Directory>
```

```
<Directory "sys:/webapps">
Options FollowSymLinks
AllowOverride None
Order deny,allow
Allow from all
</Directory>
```

```
<Directory "sys:/webapps/WebMan">
Options Indexes FollowSymLinks
AllowOverride None
Order deny,allow
Allow from all
AuthName "NetWare Web Manager"
AuthType Basic
AuthNDSTree NW6_TESTING_TREE
AuthNDSContext O=WEB
AuthNDSRequireSSL On
require valid-user
</Directory>
```

**This section is a good example of how to implement NDS authentication within an Apache Web Server.**

```
DirectoryIndex index.html
```

```
Alias /icons/ "sys:/novonyx/suitespot/bin/admin/admin/icons/"
Alias /admin-serv/icons/ "sys:/novonyx/suitespot/bin/admin/admin/icons/"
Alias /Novonyx/suitespot/manual/ "sys:/novonyx/suitespot/manual/"
```

```
<Directory "sys:/novonyx/suitespot/bin/">
Options None
AllowOverride None
Order deny,allow
Allow from all
AuthName "NetWare Web Manager"
AuthType Basic
AuthNDSTree NW6_TESTING_TREE
AuthNDSContext O=WEB
AuthNDSRequireSSL On
require valid-user
</Directory>
```

```
<Directory "sys:/novonyx/suitespot/bin/admin/admin/icons">
Options Indexes MultiViews
AllowOverride None
Order allow,deny
Allow from all
</Directory>
```

```

<Directory "sys:/novonyx/suitespot/manual">
  Options Indexes MultiViews
  AllowOverride None
  Order allow,deny
  Allow from all
</Directory>

AddHandler lcgi-script nlm pl
LCGIModuleMap sys:\nsn\lcgi\scrtptgs.nlm .asp .nsp /sp
LCGIModuleMap sys:\nsn\lcgi\cgi2ucs.nlm .bas /nsn
LCGIModuleMap sys:\perl\lcgi\cgi2perl.nlm .pl /perl

AddEnvVar NETSITE_ROOT /Novonyx/suitespot
AddEnvVar GLOBAL_ROOT /Novonyx/suitespot
AddEnvVar ADMSERV_ROOT /Novonyx/suitespot/admin-serv/config
AddEnvVar ADMIN_LOG /Novonyx/suitespot/admin-serv/logs/admin.log
AddEnvVar COMMIT_LOG /Novonyx/suitespot/admin-serv/logs/commit.log
AddEnvVar PATH_ROOT /https-NW6_TRAINING/bin/
AddEnvVar PERL_ROOT /Novonyx/suitespot/bin/https/admin
  Here we are adding environment variables.

ScriptAliasMatch ^/([\^-\]*-[\^]/(.*) "sys:/novonyx/suitespot/bin/$1/admin/$2"

<LocationMatch "/.*/(bin|admin)">
  SetHandler lcgi-script
  Options None
  AllowOverride None
  Order deny,allow
  Allow from all
</LocationMatch>

#####
# Apache configuration for the /WebAdmin context starts.
#####

#
# The following line makes apache aware of the location of the /WebAdmin context
#
Alias /WebAdmin "SYS:/webapps/WebAdmin"
Alias /WebMan "sys:/webapps/WebMan"

#
# The following line tells the plugin to inherit the mount points (/servlet
# and *.jsp) from the base server
#
JkMountCopy On

# The following lines mount all the servlets and jspes for the WebAdmin
# context
#
JkMount /WebAdmin/ServersAdmin ajp13
# JkMount /WebMan/servlet/* ajp13
JkMount /WebMan/*.jsp ajp13
JkMount /WebMan/WebMan ajp13
JkMount /WebMan/WebManTop ajp13
JkMount /WebMan/Options ajp13
JkMount /WebMan/AdminSettings ajp13
JkMount /WebMan/AdminLogSettings ajp13
JkMount /WebMan/ReadErrorLog ajp13
JkMount /WebMan/ReadAccessLog ajp13

#
# The following line prohibits users from directly accessing WEB-INF
#
<Location "/WebAdmin/WEB-INF/">

```

```

    AllowOverride None
    deny from all
</Location>
#
# Use Directory too. Location doesn't work unless case matches
#
<Directory "SYS:/webapps/WebAdmin/WEB-INF/">
    AllowOverride None
    deny from all
</Directory>

#
# The following line prohibits users from directly accessing WEB-INF
#
<Location "/WebMan/WEB-INF/">
    AllowOverride None
    deny from all
</Location>
#
# Use Directory too. Location doesn't work unless case matches
#
<Directory "SYS:/webapps/WebMan/WEB-INF/">
    AllowOverride None
    deny from all
</Directory>

#
# The following line prohibits users from directly accessing META-INF
#
<Location "/WebAdmin/META-INF/">
    AllowOverride None
    deny from all
</Location>
#
# Use Directory too. Location doesn't work unless case matches
#
<Directory "SYS:/webapps/WebAdmin/META-INF/">
    AllowOverride None
    deny from all
</Directory>

#####
# configuration for the /WebAdmin context ends.
#####

include "SYS:/NSearch/WEB-INF/NSAdmApache.conf"
Redirect /iManage/ https://137.65.55.77:2200/eMFrame/iManage.html
Redirect /iManage https://137.65.55.77:2200/eMFrame/iManage.html
Include "SYS:/webapps/eMFrame/WEB-INF/eMFrame-apache.conf"
</VirtualHost>

# NetWare Web Manager config ends

# This is to work around a bug in IE that doesn't handle the server shutting
# down keepalive requests on secure sockets
<VirtualHost _default_:443>
    BrowserMatch "MSIE" nokeepalive downgrade-1.0 force-response-1.0
</VirtualHost>

include "SYS:/NSearch/WEB-INF/NSSrchApache.conf"
Include statements allow us to append to the ADMINSEV.CONF file.
# NetWare WebAccess Config
Include sys:/webapps/webaccess/web-inf/nwwaApache.conf
# Include XTier configuration file
Include sys:/netstorage/xsrv.conf

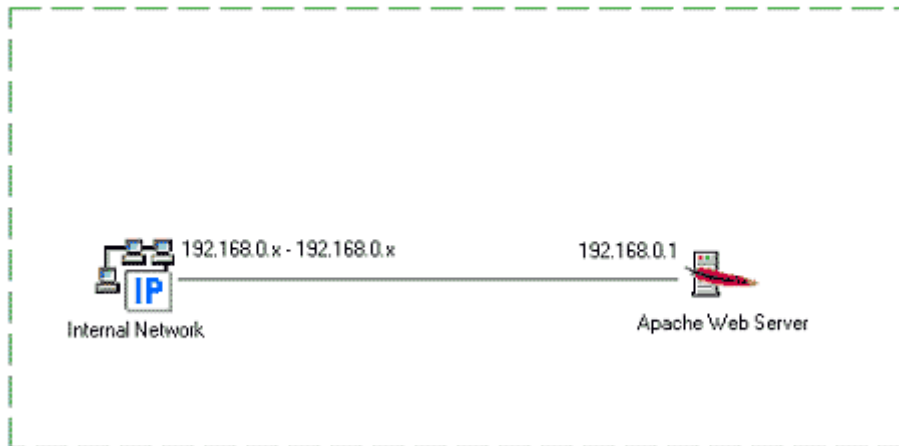
```

## Part 7: Apache Web Server Installation and Deployment

### Intranet Only

This is the easiest of solutions, but also the rarest. This is the easiest deployment because you do not need to worry about outside access to your web server. All of the access will be coming from the inside only. The reason this is the rarest deployment is due to the fact that a web server is usually for publishing content to the outside world. This simply consists of configuring your web server to an internal IP address, which will most likely be the default configuration of the server itself.

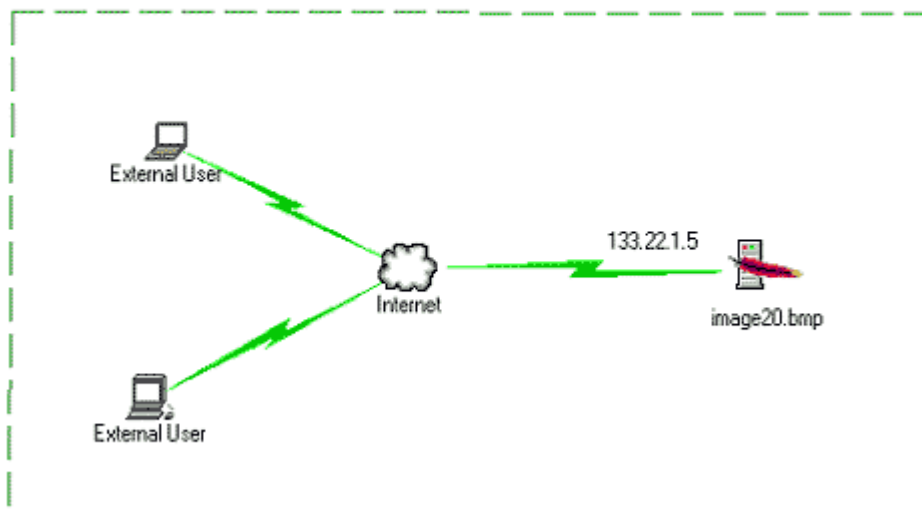
#### Internal Network - No Outside Access



### Extranet Only

An example of an extranet deployment would be an ISP. This is a deployment to strictly provide content to the public. In this instance NO content is provided to an internal network. This is also a rare instance. Again, the default installation of Apache should be sufficient for this deployment assuming that the server is installed with only a public IP address.

#### External Network - No Internal Access



## Intranet and Extranet

The Intranet / Extranet deployment is the most common and the most difficult. Most companies have internal and external content to be deployed. Several different factors need to be taken into consideration before deploying an Apache server. If Apache has already been installed and the deployment needs to be changed then there are other issues we will need to look at. Consider the following questions:

(1) How many Network Interface Cards (NIC's) does the server have? **\*\* NOTE \*\*** Remember that every NIC is going to represent a different network. In most instances, a person would have two NIC's if they are trying to bridge an Intranet and an Extranet together.

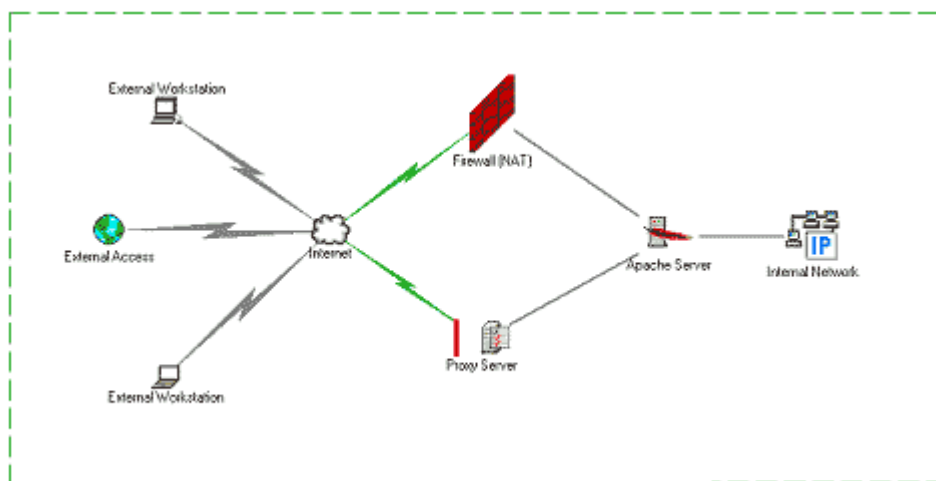
- If the server has one NIC, is it on a public or private IP address?
  - If a public IP address, do you want to provide content from this web server to your internal users?
  - If a private IP address, do you want to provide content from this web server to your external users?
- If the server has more than one NIC, are you bridging an internal and external network, bridging more than one internal network, or bridging more than one external network?
  - If bridging an internal and external network, how are you providing access to users on the outside?
    - NAT? (BorderManager, Router, Firewall, etc.)
    - Reverse Proxy? (BorderManager, Router, Firewall, etc.)
    - Forward Proxy? (iChain, etc)
  - If bridging more than one internal network, how are you providing access to users on the outside?
  - If bridging more than one external network, how are you providing access to users in the inside?

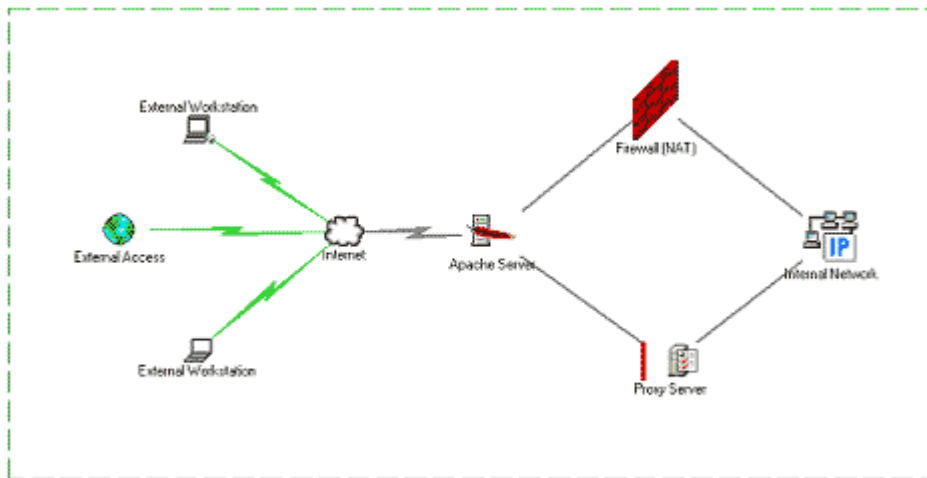
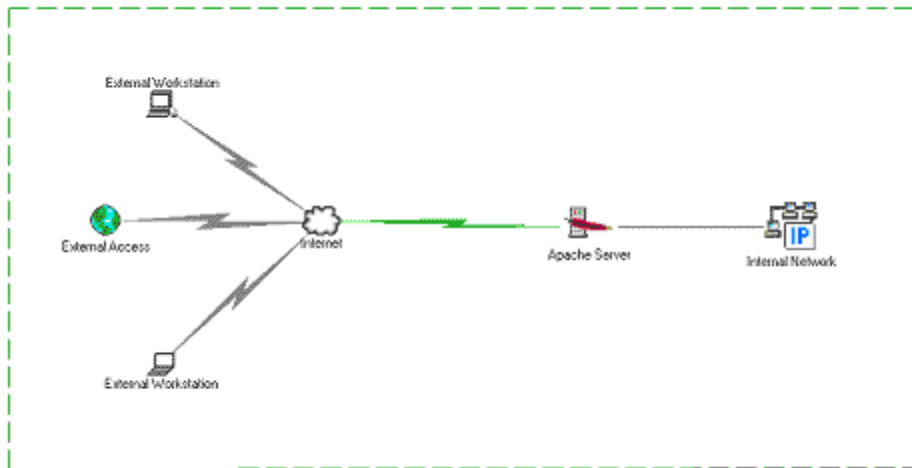
(2) Is the content that you are going to provide to internal and external users the same content, different content, or both?

(3) What is the type of content that you are going to provide?

(4) Is your site going to have public access, SSL, restrictions to content, or a combination?

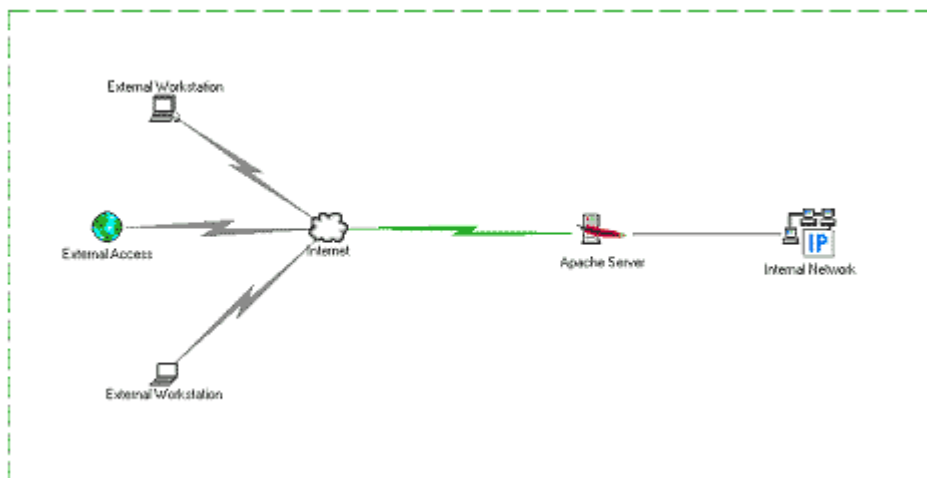
(5) Are there any programs that are going to be using Apache as their web server?





## Small Business

The small business suite has the option of doing an express or a custom installation. If an express installation is chosen then the server will install everything, as far as the web is concerned, onto a private IP address. This causes an issue if the customer is trying to have public access to his services. The most common deployment for small business is as seen in the diagram below:



## PortResolverInstalled.Properties

### Single IP Address VS. Multiple IP Address.

**\*\* NOTE \*\*** NetWare 6 has the ability to use Multiple IP Addresses or a Single IP Address to load handle its web servers. The reason for this is simple. Both Apache and the Enterprise Server use the same ports (80 and 443). This brings us to the reason for the choice between using a single IP address or multiple IP addresses.

If you have a single IP address you will need to assign ports other than 80 and 443 to one of the services (if both are being installed). The issue here is that port 80 represents HTTP and port 443 represents HTTPS. So long as you are specifying HTTP or HTTPS there is no need to place the port number at the end of the URL. The ports are assumed. If you specify the single IP address option you will need to specify the port at the end of the URL. Example: `http://192.168.0.1:1000`. If you specify the multiple IP address option, you can assign port 80 and 443 to another IP address. Thus eliminating the need to place the port at the end of the URL.

**\*\*IMPORTANT\*\*** You can allow the same IP address to listen on multiple ports. You can allow the same port to listen on multiple IP addresses. You CANNOT allow the same IP address to listen on the same port multiple times.

Since there are several services within NetWare 6 that want to use the same ports, you are given the option to have one IP address with multiple ports or multiple IP addresses with the same ports. The following services on NetWare 6 try to use port 80 or 443:

- NetWare Enterprise Server
- Apache
- iFolder
- iPrint

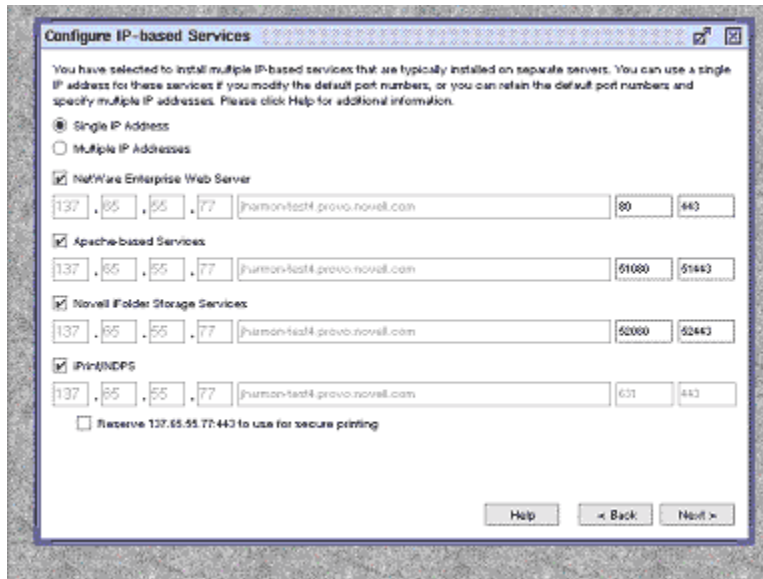
**\*\* IMPORTANT \*\*** If you choose the option for SINGLE IP ADDRESS, you will need to decide which ports you want to use. Keep in mind that the ports may already be taken by another product. For a list of common ports you can go to [NetWare 6 Port Assignments](#). By default, if the NetWare Enterprise Web Server is installed, it will take over port 80 and 443. If desired you can change this over to the Apache-based Services. However, one thing to keep in mind is that you will have to configure Apache through a configuration file vs. the Enterprise Server being configured through a GUI interface.

**\*\* IMPORTANT \*\*** Whichever service owns port 80 and 443 will also receive the NetWare 6 home page. If assigned to the Enterprise Server, the NetWare 6 home page will be copied out to the `SYS:/NOVONYX/SUITESPOT/DOCS` directory. If assigned to Apache-based Services then the NetWare 6 home page will be copied out to the `SYS:/APACHE/NWDOCS` directory. If you desire to keep the NetWare 6 home page and host your own home page on port 80 and 443, then it is suggested that you choose the multiple IP address option. You can still choose the single IP address option and host both the NetWare 6 page and your own home page, but it will require additional configuration after the installation.

**\*\* IMPORTANT \*\*** Be sure that you engrave this next point into your mind. There is a section for the IP address, DNS name, and ports to be used for each service. During the install the section for DNS name is used to configure most of the web services configuration files. This is why the services will fail if the DNS name does not resolve. I spoke about this in step 18. If you are running a test box and do not want to worry about DNS resolution, then you can place the IP address in the host field as well as in the IP field. This will then configure your files with the IP address and not the DNS name. The only time where you will run into a problem is if you need to setup reverse proxy for outside access. Reverse proxy will need information within certain products to be the DNS name and not the IP address. But for testing purposes this should be fine.

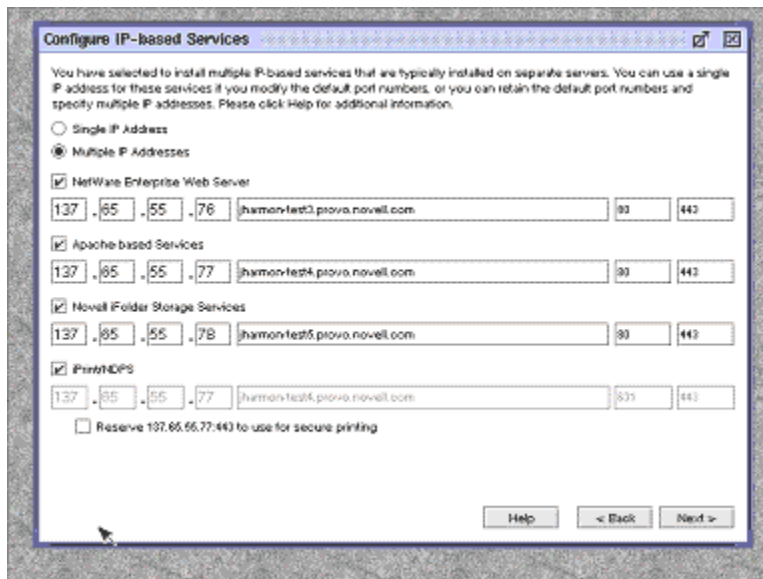
### Single IP address option

**\*\* NOTE \*\*** In this instance port 80 and 443 are defaulting to the Enterprise Server. If left this way the NetWare 6 home page will be given to the Enterprise Server. If this is an upgrade then your `INDEX.HTML` file will be renamed. If you had an existing Enterprise Server installed then it is suggested that you give port 80 and 443 to Apache-based Services. That way Apache will be hosting the NetWare 6 home page. The last thing I want you to note here is the secure port for iPrint. iPrint will take 443 for its secure port. This is required by the RFC. If you load a Web Service and iPrint on the same box you will need either need to change port 443 for the web service. iPrint is grayed out and does not allow you to change the port.



### Multiple IP address option

**\*\* REMEMBER \*\*** Back when the server was being installed I gave the server 137.65.55.77 for its IP address. By default the main IP address will be given to the first one on the list. If the Enterprise Server is installed it will default to that service. In this instance I will change that to the Apache-based Services. This is not necessary but will keep cause a lot less confusion. Remember that if you already have the Enterprise Server installed and this is an upgrade, the NetWare 6 home page (which is an INDEX.HTML) will rename any existing INDEX.HTML that exists. This is another reason to give the main IP address to Apache. See a more detailed explanation below.



If you choose the multiple IP address option then there are a few things you should note to avoid confusion. The main IP address that you give to the server will be picked up by the NetWare Enterprise Web Server by default. The main IP address will also host the NetWare 6 home page.

**Option 1** - If the main server IP address is given to the NETWARE ENTERPRISE WEB SERVER, the following will occur:

1. The NetWare 6 home page will be run by the NetWare Enterprise Web Server under the SYS:/NOVONYX/SUITESPOT/DOCS directory.

2. Apache Services will be available on the secondary IP address. However, the Web Manager (which runs through Apache) will be listening on the main IP address (same one the Enterprise Server is using) on port 2200.
3. Apache will be available to run your home page under the SYS:/APACHE/NWDOCS directory since it is not using the NetWare 6 home page. However, if you plan on using the NetWare Enterprise Server to host your pages you will lose the NetWare 6.0 home page.

**Option 2** - If the main server IP address is given to APACHE-BASED SERVICES, the following will occur:

1. The NetWare 6 home page will be run by Apache Services under the SYS:/APACHE/NWDOCS directory.
2. Apache Services will be available on the main IP address along with the Web Manager running on port 2200 on that IP address.
3. Apache will not be able to run your home page by default because it will be hosting the NetWare 6 page. The Enterprise Server will be available to host your home page at this point under the SYS:/NOVONYX/SUITESPOT/DOCS directory.

Therefore, if you want Apache to host your pages on port 80 and 443, then give the Enterprise Server the main server IP address so that Apache will be open on port 80 and 443 (Option 1). If you want the Enterprise Server to host your pages on port 80 and 443 then give Apache-Based Services the main Server IP address (Option 2).

### **HTTP vs. HTTPS (Clear Text vs. SSL) Public Access vs. Restricted Access**

A common misconception is that Clear Text and Public Access or Authentication and SSL are one entity. They do work in conjunction with each other, but they also have the capability to work as separate entities. There are times that you may want them as separate entities and times when you would not. These are discussed below:

**HTTP (Clear Text)** - HTTP is a Hypertext protocol that is transferred over clear text. Clear text means that if someone was running a trace, then it is possible to read and interpret the information.

**HTTPS (SSL)** - HTTPS is a Hypertext protocol that is transferred over SSL or encryption. This means that if a trace was run on your system the information would be encrypted and could not be read without first decrypting the information.

**Public Access** - Public Access indicates that there is no restriction on the information you are publishing. In essence, public access means that it is available to the general public and is not restricted. Anyone can access it.

**Restricted Access** - Restricted Access means that you have privileged information with which only a selected group or individual are allowed to access. There are two types of restrictions that can occur:

- **Authenticated Access** - Authenticated Restriction requires you to authenticate in order for Apache to understand who you are and what rights you have.
- **Blocked Access** - Blocked Access allows access to only selective groups according to IP address or domain. This is more at the physical machine level rather than the user level.

All of these services can be used separately or in conjunction depending on what is being accomplished. Below are some of the common combinations and whether or not they are suggested as proper implementations.

- **HTTP and Public Access** - This is suggested if you have a site that requires no restrictions to the content you are trying to provide.
- **HTTP and Authentication** - This is NOT suggested but can be done by turning off the parameter "AuthNDSRequireSSL" found in the NDS Authentication deployment. The reason this is not suggested is because you will be sending a user name and password over clear text.
- **HTTP and Blocked Access** - This is suggested if you have a certain range of IP address or certain domains that you do not want to allow access to your content.

- HTTPS and Public Access - This is suggested if you are providing your own login in the form of a public page. An example of this would be GroupWise WebAccess. You want the login page to be public, but you want to encrypt the information that is being passed.
- HTTPS and Authenticated Access - This is suggested if you need to check what rights an individual user has before they access information. In this situation the web server is providing the login.
- HTTP to HTTPS - This is suggested if you are going to be using SSL (HTTPS) and you don't want your users to have to remember to put https into the browser.

## HTTP VS. HTTPS LAB

Plan the directory structure and implementation for the scenarios listed below. The Authentication section can wait until the next section.

- HTTP and Public
- HTTP and Blocked Access
- HTTPS and Public Access
- HTTPS and Authenticated
- HTTP to HTTPS

## Authentication

### NDS Authentication - MOD\_NDS

#### AuthNDSUserFile

**Definition:** Sets the name of a text file containing a list of usernames that are allowed to authenticate (obsolete on NetWare).

**Syntax:** AuthNDSUserFile <File-Name>

**Context:** directory, .htaccess

#### AuthNDSAuthoritative

**Definition:** Determines whether the request is allowed to be passed on to lower level modules for further authentication

**Syntax:** AuthNDSAuthoritative <On (default) | Off>

**Context:** directory, .htaccess

#### AuthNDSTree

**Definition:** Sets the NDS tree that will be used for user authentication. This is a mandatory directive.

**Syntax:** AuthNDSTree <Tree-Name>

**Context:** directory, .htaccess

#### AuthNDSRequirePW

**Definition:** Determines if a user name with an empty password will be allowed to access the site.

**Syntax:** AuthNDSRequirePW <On | Off (default)>

**Context:** directory, .htaccess

#### AuthNDSExpiredURI

**Definition:** Provides redirection to an alternate page if an expired password is detected.

**Syntax:** AuthNDSExpiredURI</Path/To/Expired-Notice.html>

**Context:** directory, .htaccess

#### AuthNDSCacheTimeout

**Definition:** Sets the time-to-live value for entries in the cache (in seconds), or disables the cache entirely (by setting it to zero).

**Syntax:** AuthNDSCacheTimeout <Number> (Default is 300)

**Context:** directory, .htaccess

#### AuthNDSUniqueCNS

**Definition:** Enables the caching of name->FDN mappings, which prevents the module from having to search for the user's FDN on every request

**Syntax:** AuthNDSUniqueCNS <On | Off (default) >

**Context:** server config

#### AuthNDSContext

**Definition:** Sets a search list of contexts for contextless logins.

**Syntax:** AuthNDSContext <.Context.To.Search.Context.To.Search ...>

**Context:** directory, .htaccess

#### **AuthNDSContextOverride**

**Definition:** This directive only applies to AuthNDSContext. If set to 'ON' for a given directory, it causes all search contexts defined in higher-level directories to be ignored.

**Syntax:** AuthNDSContextOverride <On | Off (Default)>

**Context:** directory, .htaccess

#### **The following 'require' directives are supported: require user**

**Definition:** Defines a list of valid users.

**Syntax:** require user <.user1.full.context .user2.full.context .user3.full.context ...>

#### **require valid-user**

**Definition:** Allows access for any valid user name and password.

**Syntax:** require valid-user

#### **require context**

**Definition:** Allows access for any valid user name and password with a matching context.

**Syntax:** require context <.exact.matching.context1 .exact.matching.context2 ...>

#### **require context**

**Definition:** Allows access for any valid user name and password with a partial matching context.

**Syntax:** require context </.partially.matching.context1 ...>

## **NDS Authentication LAB**

You would have to add an alias here if you were accessing anything before the Apache root directory.

```
LoadModule nds_auth_module modules/mod_nds.nlm
```

```
LoadModule tls_module modules/mod_tls.nlm
```

```
<Directory "sys:/novonyx/suitespot/bin/">
    Options None
    AllowOverride None
    Order deny,allow
    Allow from all
    AuthName "NetWare Web Manager"
    AuthType Basic
    AuthNDSTree JOE60_TREE
    AuthNDSContext O=WEB
    AuthNDSRequireSSL On
    require valid-user
</Directory>
<IfModule mod_tls.c>
    SecureListen 137.65.215.72:443 "SSL CertificateDNS"
</IfModule>
```

**allow from** address or expression  
**allow from env**=environment variable  
**deny from** address or expression  
**deny from env**=environment variable

**\*\* NOTE \*\*** With the NetWare Enterprise Server (NES) we were not able to have public and private directories under each other. They had to be at the same level. With Apache and NDS Authentication we can have public and private directories under each other as well restricting with the following directives:

- By User
- By Container
- By Domain
- By IP address
- By ENV
- Using wild cards

## LDAP Authentication - MOD\_LDAP

### AuthLDAPBindDN

**Definition:** An optional DN used to bind to the server when searching for entries. If not provided, AUTH\_LDAP will use an anonymous bind.

**Syntax:** AuthLDAPBindDN <Distinguished-Name>

**Context:** directory, .htaccess

### AuthLDAPBindPassword

**Definition:** A bind password to use in conjunction with the bind DN

**Syntax:** AuthLDAPBindPassword <Password>

**Context:** directory, .htaccess

### AuthLDAPAuthoritative

**Definition:** Set to 'OFF' if this module should let other authentication modules attempt to authenticate the user, should authentication with this module fail

**Syntax:** AuthLDAPAuthoritative <On (Default) | Off>

**Context:** directory, .htaccess

### AuthLDAPURL

**Definition:** A URL which specifies the LDAP search parameters to use.

**Syntax:** AuthLDAPURL <url>

**Context:** directory, .htaccess

### AuthLDAPRemoteUserIsDN

**Definition:** If this directive is set to 'ON', the value of the REMOTE\_USER environment variable will be set to the full distinguished name of the authenticated user, rather than just the username that was passed by the client

**Syntax:** AuthLDAPRemoteUserIsDN < Off (Default) | On>

**Context:** directory, .htaccess

### AuthLDAPCertDBPath

**Definition:** Specifies in which directory AUTH\_LDAP should look for the certificate authorities database. There should be a file named cert7.db in that directory.

**Syntax:** AuthLDAPCertDBPath </Path/To/Cert7.db/Directory>

**Context:** server config

### AuthLDAPCacheSize

**Definition:** Specifies the maximum size of the LDAP search cache

**Syntax:** AuthLDAPCacheSize <Size>

**Context:** server config

### AuthLDAPCacheTTL

**Definition:** Specifies the time (in seconds) that an item in the search cache remains valid.

**Syntax:** AuthLDAPCacheTTL <Time>

**Context:** server config

### AuthLDAPOpCacheSize

**Definition:** Specifies the size of the cache AUTH\_LDAP uses to cache LDAP operations.

**Syntax:** AuthLDAPOpCacheSize <Size>

**Context:** server config

### AuthLDAPOpCacheTTL

**Definition:** Specifies the time (in seconds) that entries in the operation cache remain valid. The default is 600 seconds.

**Syntax:** AuthLDAPOpCacheTTL <Time>

**Context:** server config

### AuthLDAPCacheCompareOps

**Definition:** If this directive is set to 'ON', AUTH\_LDAP will cache any compare operations (these are used to satisfy require user directives).

**Syntax:** AuthLDAPCacheCompareOps <On (Default) | Off>

**Context:** server config

## LDAP Authentication LAB

```
LoadModule auth_ldap_module modules/authldap.nlm
LoadModule tls_module modules/mod_tls.nlm

<IfModule auth_ldap.c>
    #Alias /secure "sys:/Apache/htdocs/secure"
    <Directory "sys:/Apache/htdocs/secure">
        UseCanonicalName Off
        Options Indexes MultiViews
        Order deny,allow
        Allow from all

        #Directives to allow from specific IP addresses
        #Deny from all
        #Allow from 137.65.53.134

        AuthType Basic
        AuthName Secure_Docs
        AuthLDAPURL ldap://137.65.55.71/o=web
        require valid-user
    </Directory>
</IfModule>
<IfModule mod_tls.c>
    SecureListen 137.65.215.72:443 "SSL CertificateDNS"
</IfModule>
```

1. **allow from** address or expression
2. **allow from env**=environment variable
3. **deny from** address or expression
4. **deny from env**=environment variable

**\*\* NOTE \*\*** With the NetWare Enterprise Server (NES) we were not able to have public and private directories under each other. They had to be at the same level. With Apache and NDS Authentication we can have public and private directories under each other as well restricting with the following directives:

- By User
- By Container
- By Domain
- By IP address
- By ENV
- Using wild cards

## Applications and their Authentication

Novell iFolder - LDAP Authentication to Apache  
Novell NetStorage - NDS Authentication to Apache  
Jakarta-Tomcat - NDS Authentication passed by Apache  
NSearch - NDS Authentication  
Novell Portal Services - LDAP Authentication  
NetWare Web Manager - NDS Authentication  
iManager - LDAP Authentication  
eGuide - LDAP Authentication

### iFolder Example

```
LoadModule ifolderserver_module "iFolder/Server/iFolder.nlm"

<VirtualHost ifolder6.provo.novell.com:80>
    ServerName ifolder6.provo.novell.com
```

```
DocumentRoot "SYS:\apache\iFolder\DocumentRoot"
```

```
<Directory "SYS:\apache\iFolder\DocumentRoot">  
Options Indexes FollowSymLinks MultiViews  
AllowOverride None  
Order allow,deny  
Allow from all  
</Directory>
```

```
<location /iFolderServer>  
SetHandler ifolderserver-form-handler  
</location>  
LdapHost apache6.provo.novell.com  
LdapPort 636  
LdapLoginDnContext "O=WEB"  
LdapRootCert "SYS:\apache\iFolder\server\RootCert.der"  
iFolderServerRoot SYS:\iFolder  
iFolderAdminName admin  
ServerSecurePort 443
```

```
</VirtualHost>
```

## NSearch Example

```
<?xml version="1.0" encoding="ISO-8859-1" ?>  
<webapps> <!-- Setting special properties for NSearch context -->  
<Context path="/NSearch" docBase="SYS:/NSearch" debug="0" reloadable="true">  
<NDSAuth path="SYS:/NSearch/nsrchjw.nlm" />  
</Context>  
</webapps>
```

MODULES.XML

```
<module name="NDSAuth" javaClass="com.novell.tomcat.modules.aaa.NDSAuth" />
```

MODULES.PROPERTIES

```
NDSAuth=com.novell.tomcat.modules.aaa.NDSAuth
```

## NetWare Web Manager Example

```
<Directory "sys:/webapps/WebMan">  
Options Indexes FollowSymLinks  
AllowOverride None  
Order deny,allow  
Allow from all  
AuthName "NetWare Web Manager"  
AuthType Basic  
AuthNDSTree JOE60_TREE  
AuthNDSContext O=WEB  
AuthNDSRequireSSL On  
require valid-user  
</Directory>
```

## Virtual Hosting

### IP Based Virtual Hosts

There are several components that need to be understood about an IP based virtual host:

- They must load with either a different IP address or a different port.
- They can contain most apache directives.

- If a non-critical directive in the virtual host fails, then it will revert back to that same directive outside of the virtual host.
- If the virtual hosts document root is outside of the main Apache document root then access will need to be defined to that directory.

```
<VirtualHost 137.65.55.74>
    DocumentRoot sys:/Apache/htdocs/virtual
</VirtualHost>
```

## IP Based Virtual Hosting LAB

Build two IP based virtual hosts, one under the main document root and the other under a directory outside of the Apache document root. You will need to allow access to the virtual server that is outside of the Apache document root.

### Domain Based Virtual Hosts

Domain Based Virtual hosts allow the same IP address to be used with each virtual host being defined with a different DNS name. However, there are several rules that apply.

- Since you are using the same IP address then the only way to distinguish between the virtual hosts is by the DNS name. Therefore DNS resolution must first exist.
- ServerName is a required field.

```
NameVirtualHost 137.65.55.81
<VirtualHost 137.65.55.81>
    ServerName www.joeserver.com
    ServerAlias joeserver.com *.joeserver.* joeserver
    DocumentRoot sys:/Apache/htdocs/virtual/joeserver
</VirtualHost>
<VirtualHost 137.65.55.81>
    ServerName www.otherserver.com
    DocumentRoot sys:/Apache/htdocs/virtual/otherserver
</VirtualHost>
```

## Domain Based Virtual Hosting LAB

Create 3 Virtual Hosts off of your main IP address.

### Indexing

#### Indexing Types

```
<IfModule mod_autoindex.c>
    IndexOptions FancyIndexing
    AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
    AddIconByType (TXT,/icons/text.gif) text/*
    AddIconByType (IMG,/icons/image2.gif) image/*
    AddIconByType (SND,/icons/sound2.gif) audio/*
    AddIconByType (VID,/icons/movie.gif) video/*
    AddIcon /icons/binary.gif .bin .exe
    AddIcon /icons/binhex.gif .hqx
    AddIcon /icons/tar.gif .tar
    AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
    AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
    AddIcon /icons/a.gif .ps .ai .eps
    AddIcon /icons/layout.gif .html .shtml .htm .pdf
    AddIcon /icons/text.gif .txt
    AddIcon /icons/c.gif .c
    AddIcon /icons/p.gif .pl .py
    AddIcon /icons/f.gif .for
```

```

AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core
AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^
DefaultIcon /icons/unknown.gif
ReadmeName README
HeaderName HEADER
IndexIgnore .??* *- *# HEADER* README* RCS CVS *,v *,t
</IfModule>

```

## Indexing LAB

Create an indexing option that will associate a file extension to a graphic.

### Aliasing

#### Alias

```

<IfModule mod_alias.c>
    Alias /add "sys:/apache/add"
    <Directory "sys:/apache/add">
        Options Indexes MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>
</IfModule>

```

## Aliasing LAB

Create two aliases that will allow access to directories outside of the Apache document root. After the aliases have been created, allow public access to one and restrict the other either with NDS or LDAP authentication.

Remember, if it doesn't exist under the Apache Document Root then you will need to provide access to the directory structure, otherwise Apache will not know how to access the directory.

### ScriptAlias

```

<IfModule mod_lcgi.c>
    AddHandler lcgi-script nlm pl nsn bas
    LCGIModuleMap sys:\nsn\lcgi\scriptpgs.nlm .asp .nsp /sp
    LCGIModuleMap sys:\nsn\lcgi\cgi2ucs.nlm .bas /nsn
    LCGIModuleMap sys:\perl\lcgi\cgi2perl.nlm .pl /perl
    AddEnvVar PERL_ROOT sys:\Novonyx\suitespot\docs\perlroot
    ScriptAlias /perl sys:/Novonyx/suitespot/docs/perlroot
    ScriptAlias /nsn sys:/nsn/web
</IfModule>

```

## ScriptAlias LAB

Change the location of the PERL\_ROOT and verify that it is running properly by running some sample scripts.

## Listen Statements and Bindings

```
LoadModule tls_module modules/mod_tls.nlm
```

```
Listen 137.65.215.72:80
```

```
<IfModule mod_tls.c>  
    SecureListen 137.65.215.72:443 "SSL CertificateDNS"  
</IfModule>
```

## Scripting

\* This module has only been tested with Apache 1.3.14 and later. It should work for all 1.3.x versions; whether or not it will work with previous releases is unknown.

\* This module requires a release of NSL CGI.NLM dated later than 7/25/2000. A new version can be downloaded from <http://developer.novell.com/ndk/modapach.htm> or <http://support.novell.com>.

Installing Apache w/ mod\_lcgi

-----  
Including the following lines to your configuration file will enable the scripting languages and access to other NLMs written to the LCGI API for use with Apache for NetWare:

```
LoadModule lcgi_module modules/mod_lcgi.nlm  
<IfModule mod_lcgi.c>  
    AddHandler lcgi-script nlm pl nsn bas  
    LCGIModuleMap sys:\nsn\lcgi\scriptpgs.nlm .asp .nsp /sp  
    LCGIModuleMap sys:\nsn\lcgi\cgi2ucs.nlm .bas /nsn  
    LCGIModuleMap sys:\perl\lcgi\cgi2perl.nlm .pl /perl  
    AddEnvVar PERL_ROOT sys:\Novonyx\suitespot\docs\perlroot  
    ScriptAlias /perl sys:/Novonyx/suitespot/docs/perlroot  
    ScriptAlias /nsn sys:/nsn/web  
</IfModule>
```

Configuration Directives:

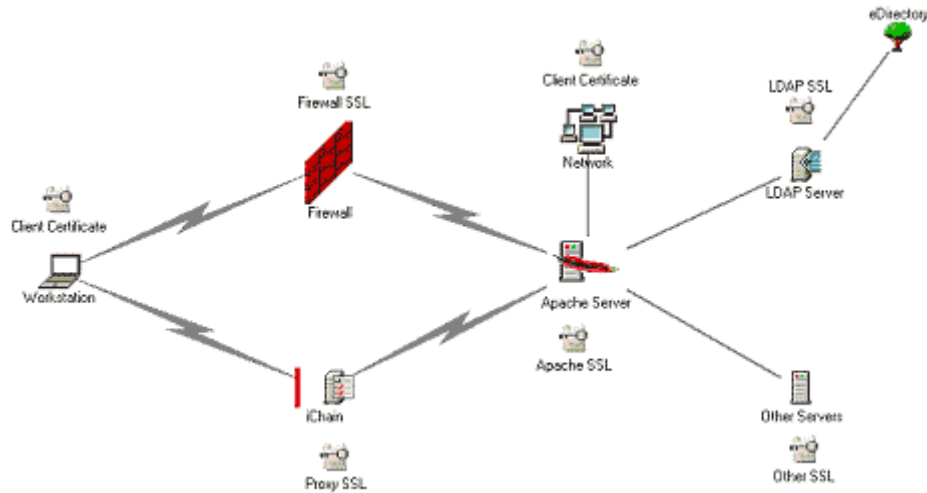
-----  
--  
LCGIModuleMap  
Syntax: LCGIModuleMap <LCGI\_NLM\_Path> <.ext .ext ...>  
Context: server config  
Associates one or more file extensions with an LCGI module.  
--  
--  
AddEnvVar  
Syntax: AddEnvVar <path>  
Context: server config  
Set additional environment variables for use within an LCGI script or module  
--  
Perl Samples  
ASP Samples

## Scripting LAB

Change the PERL\_ROOT

## SSL with Web Products

- Browser to Server
- Server to Server
- Server to LDAP
- RSA
- Proxy to Server
- NAT



## Novell Functionality

### User Home Directories

#### Prerequisites:

1. Due to its dependency on DSAPI, this module only works under NetWare.
2. These modules have only been tested with Apache 1.3.14 and later. They should work for all 1.3.x versions; whether or not it will work with previous releases is unknown.
3. Apache 1.3.19 or later is required for remote directory support.

Add the following lines to your httpd.conf:

```
LoadModule hdirs_module modules/modhdirs.nlm
LoadModule rdirs_module modules/modrdirs.nlm
```

#### Configuring Apache for mod\_rdirs

MOD\_RDIRS does not need any additional configuration. By default whenever a path similar to: SERVER/VOLUME:/PATH/FILE is requested, MOD\_RDIRS will make the appropriate connection to the remote server and convert the path to short filename. Once the Connection has been established and the filename converted, Apache treat the request as a normal request. Once the request has been satisfied, MOD\_RDIRS will cache the connection and clean up.

#### Configuring Apache for mod\_hdirs

Make sure that the Apache MOD\_USERDIR module is not enabled by removing any UserDir directives from the .CONF file. The default setting for MOD\_HDIRS will allow it to access home directories using the standard URI notation. No addition directives are required unless special considerations are needed.

#### Configuration Directives:

--

##### UserDirTag

Syntax: UserDirTag Context: server config

This sets the URI tag that indicates a user home directory. If omitted the default tag is '~'.

--

--

##### UserSubDir

Syntax: UserSubDir <Directory\_Name>

Context: server config

This set the sub-directory below the user home directory that will be accessed for all web page request. If omitted the default sub-directory is 'public\_html'.

The sub-directory name must not be preceded by a slash.

##### SearchNDSContext

Syntax: SearchNDSContext context.to.search1 .context.to.search2 ...

Context: server config

This creates a list of additional contexts to search for the specified NDS user ID. If omitted, only the root context will be searched.

--  
Notes

-----  
To get NDS home directory support, all that is required is to load MODHDIRS using the standard LoadModule statement in your HTTPD.CONF file. Once the module has been loaded, a request to ~USER should result in retrieving a web page from the USER's PUBLIC\_HTML sub-directory within their NDS home directory. There are several limitations when accessing files on a remote server.

- 1). All servers must be in the same tree. You can not access a remote server from your Apache server that is not in the same tree as the Apache server.
- 2). When defining a <Directory> or <File> block in your .CONF file for a remote server, the server name should not be included. Paths should be defined as <Directory VOLUME:/PATH>... or <File VOLUME:/PATH/FILE>....
- 3). Since all remote server paths and file names on NetWare are handled in short name format, all remote directory paths and file names must also be defined in short name format. For example:

```
Alias /foo/ remote/vol1:/my_remote_path/my_foo_directory
```

```
<Directory vol1:/my_foo_path/my_foo_directory>
```

```
...
```

```
</Directory>
```

should be defined as:

```
Alias /foo/ remote/vol1:/my_rem-1/my_foo-1
```

```
<Directory vol1:/my_rem-1/my_foo-1>
```

```
...
```

```
</Directory>
```

Special attention should be paid to server configuration files that begin with a '.'. For example, the following changes to the HTTPD.CONF file would allow .HTACCESS files to be accessed on a remote server:

```
AccessFileName .htaccess htacce~1
```

```
<Files ~ "\.ht">
```

```
Order allow,deny
```

```
Deny from all
```

```
</Files>
```

```
<Files ~ "\htacce~1">
```

```
Order allow,deny
```

```
Deny from all
```

```
</Files>
```

```
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t htacce*
```

The file name "htacce~1" needs to be added to the access file list to allow for the short file name format. An additional <File ...> block is also added. The short file name "htacce\*" pattern is added to the "IndexIgnore" directive to restrict users from seeing remote .htaccess files.

## User Home Directories LAB

Setup user home directories so that they can be accessed on a sys volume and another volume.

## WebDAV

November 21, 2002 ApacheCon US 20023

<http://www.webdav.org/papers/ApacheCon-2002-US-TH01.pdf>

What is WebDAV?

- Web-based Distributed Authoring and Versioning
  - "DAV" is the usual short form
- Goal: enable interoperability of tools for distributed web authoring
- Turns the Web into a writeable medium
- Applies to all kinds of content -not just HTML and images

- Based on extensions to HTTP
- Uses XML for properties, control, status
- RFC 2518

November 21, 2002 ApacheCon US 20023

<http://www.webdav.org/papers/ApacheCon-2002-US-TH01.pdf>

#### Technical Benefits

- Properties ("metadata")
- Overwrite protection
- Namespace management
- Versioning
- Infrastructure: old and new
- Replacement protocol

#### Novell's implementation of WebDAV

- NetStorage
- WebDAV with the Enterprise Server

### **Listen Statements and Bindings**

```
LoadModule tls_module modules/mod_tls.nlm
Listen 137.65.215.72:80
<IfModule mod_tls.c>
    SecureListen 137.65.215.72:443 "SSL CertificateDNS"
</IfModule>
```

## Part 8: Tomcat

---

- [Integration of Tomcat and Apache](#)
- [Servlets](#)
- [JSPs](#)
- [Separate Instance of Tomcat](#)
- [WAR files](#)

### Integration of Tomcat and Apache

#### MOD\_JK

MOD\_JK is a module that allows Apache and Tomcat to communicate. The MOD\_JK module is needed in order for Apache to be able to plug into Tomcat. This is accomplished through the JKMount statement. An example of the load statement you would need is as follows:

```
LoadModule jk_module modules/mod_jk.nlm
JkWorkersFile SYS:/tomcat/33/conf/jk/nwworkers.properties"
JkLogFile "SYS:/tomcat/33/logs/mod_jk.log"
JkLogLevel error
```

#### JKMount

The JKMount statement allows URL's within Apache to be assigned to Tomcat. An example of a JKMount would be as follows:

```
JkMount /*.jsp ajp13
JkMount /servlet/* ajp13
```

#### AJP12 and AJP13 protocols

[http://jakarta.apache.org/tomcat/tomcat-3.3-doc/mod\\_jk-howto.html#s82](http://jakarta.apache.org/tomcat/tomcat-3.3-doc/mod_jk-howto.html#s82)

MOD\_JK can use either the original Ajpv12 protocol or the newer Ajpv13 protocol. Both protocols are enabled by default. The "Ajpv13" Connection Handler in Tomcat will give you the benefit of a faster protocol and the ability to identify requests made via HTTPS.

The default installation of Tomcat 3.3 comes with connectors for both protocols in the `sys:/tomcat/33/conf/nwserver.xml`. The default choice by the ApacheConfig directive will be to use the "ajpv13" worker which uses the Ajpv13 protocol. Even though mod\_jk uses the Ajpv13 by default, you shouldn't remove the Ajpv12 connector present in the server.xml file. The Ajpv12 connector is also used for shutting down Tomcat.

#### MOD\_JK.LOG

This is the log file that records the communication between Tomcat and Apache.

### Servlets

**Servlet Definition** from the Servlet Specification:

"A servlet is a web component, managed by a container, that generates dynamic content. Servlets are small, platform independent Java classes compiled to an architecture-neutral byte code that can be loaded dynamically into and run by a web server. Servlets interact with web clients via a request response paradigm implemented by the servlet container. This request-response model is based on the behavior of the Hypertext Transfer Protocol (HTTP)."

Java Alternative - Servlets

- Faster than CGI
  - Single thread - separate process required for each CGI script
  - Scales - once loaded in memory can be called many times
- Vendor / platform neutral
  - Browser-based app need not support Java - output can be HTML, XML?
  - Runs on any servlet server
  - NSAPI & ISAPI (vendor specific CGI)
- All the advantages of Java
  - Uses generic API's - JNDI, JDBC or JMS to talk to existing enterprise resources.
  - Extensible - developers can easily extend functionality.

**\*\* NOTE \*\*** Java servlets are one of the most exciting new technologies. Servlets are efficient, persistent, portable, robust, extensible, secure, and they have received wide acceptance in the industry. Good for replacing CGI. Servlets solve many of the problems present in CGI.

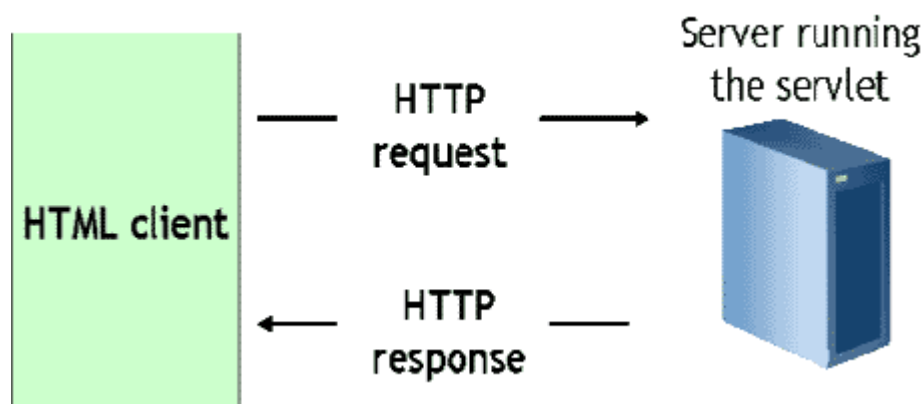
**Efficiency** - A servlet's initialization code is executed only the first time the Web server loads it. After the servlet is loaded, handling new requests is only a matter of calling a service method. This is a much more efficient technique than loading a completely new executable with every request.

**Persistence** - Servlets can maintain state between requests. When a servlet is loaded, it stays resident in memory while serving incoming requests. A simple example of this would be a Vector that holds a list of categories used in an online catalog. When the servlet is initialized, it queries the database for a list of categories and stores these categories in a Vector. As it services requests, the servlet accesses the Vector that holds the categories instead of querying the database again. Hence, performance is improved drastically.

**Portability** - Servlets are developed using Java; therefore, they are portable. This enables servlets to be moved to new operating system without changing the source. You can have code compiled in Win NT and move it to a Solaris box without making any changes.

**Browser >> Servlet >> Browser**

**\*\* NOTE \*\*** The browser issues a request to a particular URL that may also include parameters.



The servlet generates the content dynamically and returns in a response to the browser.

Common HTTP Requests and Servlet Methods:

- Get: retrieves resources identified by request URL.

- Post: sends data of unlimited length to the browser.
- Put: stores a resource under the request URL.
- Delete: removes resource identified by the request URL.
- init()
- service()    doGet() and doPost()
- destroy()
- getServletConfig()
- getServletInfo()

## Servlets LAB

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
//
//
// MyFirstServlet
//
//
class MyFirstServlet extends HttpServlet
{
public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException
{
// set content type res.setContentType("text/html"); // Hello world PrintWriter out = res.getWriter();
out.println("<H1>Hello World</H1>"); out.close(); }

public String getServletInfo() { return "My First Servlet"; }
}
```

## JSPs

**JSP Definition** from the JSP Specification:

"JavaServer Pages ? technology is the Java ? platform technology for building applications containing dynamic Web content such as HTML, DHTML, XHTML and XML. The JavaServer Pages technology enables the authoring of Web pages that create dynamic content easily but with maximum power and flexibility."

### JSP vs Servlets

- **Readability**
  - Java embedded into HTML as opposed to servlets with content (HTML) embedded into Java code
- **Ease of use**
  - Code-free bean access
  - Simple include/forward
- **Flexibility**
  - Custom tag libraries encapsulate complex functionality without Java coding - further reduce scriptlets thus cleaner markup.

**\*\* NOTE \*\*** As a general rule, JSPs are best for textual output. Servlets better for binary output, like images, or PDFs.

**Advantage** - The biggest strength of JSP is that it looks like HTML or XML. Hence, persons who are familiar with HTML or XML can easily modify JSP. People who are weak in Java can use JSP. There is no need to be a Java guru to use JSP. When JSPs are compiled they automatically become servlets. Hence, it is difficult to point to any technical disadvantages of a JSP that aren't in servlets. Probably the only disadvantage, but also a strength, is the capability to mix Java code with HTML. If you do not organize your code you might end up with a mess of huge JSP files with HTML interspersed between huge blocks of Java code.

## JSP Tag Syntax

<b>Comments</b> <code>&lt;%-- ... --%&gt;</code>	client or server side
<b>Directives</b> <code>&lt;%@ ... %&gt;</code>	messages to the JSP container
<b>Declarations</b> <code>&lt;%! ... %&gt;</code>	declare variables or methods
<b>Expressions</b> <code>&lt;%= ... %&gt;</code>	an expression to evaluate
<b>Scriptlets</b> <code>&lt;% ... %&gt;</code>	code fragments

JSPs consist of HTML (normally) plus the five syntax structures above. Sun offers a very handy "JSP Syntax Card" available at <http://java.sun.com/products/jsp>

## JSP LAB

Now we will create our first JSP (Java Server Page) to go into our Web Application. This will be used later on when we build our web application. We will name it DATEANDTIME.JSP. The contents are as follows:

```
<HTML>
  <BODY>
    Today's date and time are: <%= new java.util.Date() %>
  </BODY>
</HTML>
```

### Summary - Servlets vs JSPs

- JSPs
  - HTML with some embedded Java (but not too much!)
  - Better separation of concerns
  - Visual tools (evolving)
  - Single content type (per JSP)
- Servlets

- o Java embedded with HTML content
- o More flexibility (including content type)
- o Better with binary data (files or dynamic images)
- o Better as mediator or controller (in MVC)

The use of JSPs and Servlets usually follows the 80/20 rule. You will probably use JSPs for 80% of your web application. Servlets will only be used 20% of the time.

The biggest strength of JSP is that it looks like HTML (or XML). Because JSP eventually become servlets, it's difficult to point to any technical disadvantage that are present in servlets.

## Separate Instance of Tomcat

### TOMCAT33.NCF

How to run multiple instances of Tomcat on NetWare 6

Under the SYS:/TOMCAT/33/BIN directory you will find a TOMCAT33.NCF file. Make a copy of this file and call it what you want. In this example, MYTOMCAT.NCF will be used.

```
-----
load java

envset TOMCAT_HOME=SYS:\tomcat\33
envset TOMCAT_CLASSPATH=sys:\java\lib\classes.zip
envset TOMCAT_CLASSPATH=$TOMCAT_CLASSPATH;$TOMCAT_HOME\lib\tomcat.jar
envset TOMCAT_CLASSPATH=$TOMCAT_CLASSPATH;SYS:\java\njclv2\lib\jndi.jar
envset TOMCAT_CLASSPATH=$TOMCAT_CLASSPATH;SYS:\java\njclv2\lib\njclv2.jar
envset TOMCAT_CLASSPATH=$TOMCAT_CLASSPATH;SYS:\tomcat\33\lib\common\jnet.jar
envset TOMCAT_CLASSPATH=$TOMCAT_CLASSPATH;SYS:\tomcat\33\lib\common\jcert.jar
envset TOMCAT_CLASSPATH=$TOMCAT_CLASSPATH;SYS:\tomcat\33\lib\common\jsse.jar
envset TOMCAT_CLASSPATH=$TOMCAT_CLASSPATH;SYS:\java\lib\jclient.jar

java -envCWD=$TOMCAT_HOME -classpath $TOMCAT_CLASSPATH -
Dtomcat.home=SYS:\tomcat\33 org.apache.tomcat.startup.Main -f sys:/tomcat/33/conf/nwserver.xml %1
```

### MYTOMCAT.NCF LAB

Below you will find the MYTOMCAT.NCF file that we have created, along with an explanation of the changes that took place.

```
load java

envset TOMCAT_HOME=SYS:\tomcat\33
envset TOMCAT_CLASSPATH=sys:\java\lib\classes.zip
envset TOMCAT_CLASSPATH=$TOMCAT_CLASSPATH;$TOMCAT_HOME\lib\tomcat.jar
envset TOMCAT_CLASSPATH=$TOMCAT_CLASSPATH;SYS:\java\njclv2\lib\jndi.jar
envset TOMCAT_CLASSPATH=$TOMCAT_CLASSPATH;SYS:\java\njclv2\lib\njclv2.jar
envset TOMCAT_CLASSPATH=$TOMCAT_CLASSPATH;SYS:\tomcat\33\lib\common\jnet.jar
envset TOMCAT_CLASSPATH=$TOMCAT_CLASSPATH;SYS:\tomcat\33\lib\common\jcert.jar
envset TOMCAT_CLASSPATH=$TOMCAT_CLASSPATH;SYS:\tomcat\33\lib\common\jsse.jar
envset TOMCAT_CLASSPATH=$TOMCAT_CLASSPATH;SYS:\java\lib\jclient.jar

java -ns -sn"My Tomcat" -Xms128m -Xmx256m -envCWD=$TOMCAT_HOME -classpath $TOMCAT_CLASSPATH -
Dtomcat.home=SYS:\tomcat\33 org.apache.tomcat.startup.Main -f sys:/tomcat/33/conf/myserver.xml %1
```

The -ns parameter is if you want the new version of tomcat to appear in its own screen. If this parameter is left out then it will be sent to the logger screen by default.

**\*\*Note\*\*** The logger screen has the ability to scroll back up the screen. The -Xms128m and -Xmx256m parameter specifies the maximum and minimum amount of memory for Tomcat to use. If you are using applications that do not ship with NetWare 6 by default, then you may want to bump up this parameter.

**\*\*Note\*\*** These parameters cannot go over the amount of RAM that you have on your server. We renamed the NWSERVER.XML to be MYSERVER.XML. This will make more sense as we move on to the next step.

## NWSERVER.XML

The next file that we are going to create is the MYSERVER.XML file. To do this we need to go to the SYS:/TOMCAT/33/CONF directory and copy the NWSERVER.XML file. Once that is copied we will give it the name of MYSERVER.XML file.

Here we will need to make several changes so that we can properly manage our web applications and so we don't run into any port conflicts.

```
-->
<ContextManager debug="0" workDir="work" home="SYS:/tomcat/33">
  <!-- ===== Global modules ===== -->
  <LogSetter name="tc_log" timestamps="true"
    verbosityLevel="INFORMATION" />
  <LogEvents enabled="false" />

  <!-- Backward compat: read the Context declarations from server.xml-->
  <ContextXmlReader config="conf/myserver.xml" />

  <!-- Separated Context -->
  <ContextXmlReader config="conf/myapps.xml" />
  <AutoDeploy source="webapps" target="webapps" />
  <AutoWebApp dir="webapps" host="DEFAULT" />
  <PolicyLoader securityManagerClass="java.lang.SecurityManager"
    policyFile="conf/tomcat.policy" />
  <SimpleMapper1 debug="0" />
  <SessionExpirer checkInterval="60" />
  <SessionIdGenerator randomClass="java.util.Random"
    randomFile="/dev/urandom" />
  <!-- ===== context processing modules ===== -->
  <!-- This will be the "default" profile
    ( all except the "global" modules can be set per context )
  -->
  <LogSetter name="servlet_log"
    timestamps="true"
    verbosityLevel = "INFORMATION"
    path="logs/nwservlet-${yyyyMMdd}.log"
  />
  <LogSetter name="JASPER_LOG"
    timestamps="true"
    path="logs/nwjasper-${yyyyMMdd}.log"
    verbosityLevel = "INFORMATION" />
  <LoaderInterceptor11 useApplicationLoader="true" />
  <WebXmlReader validate="true" />
  <ErrorHandler showDebugInfo="true" />
  <WorkDirSetup cleanWorkDir="false" />
  <Jdk12Interceptor />
  <!-- Non-standard invoker, for backward compat. ( /servlet/* ) -->
  <InvokerInterceptor />
  <!-- you can add javaCompiler="jikes" -->
  <JspInterceptor keepGenerated="true"
    largeFile="false"
    sendErrToClient="true"
    useJspServlet="false"
    javaCompiler="javac"
  />

  <StaticInterceptor debug="0" listings="false" />

  <ReloadInterceptor fullReload="true" />
  <SimpleSessionStore maxActiveSessions="-1" />
  <AccessInterceptor />
```

```

<CredentialsInterceptor />
<LoadOnStartupInterceptor />
<Servlet22Interceptor />
<DecodeInterceptor debug="0" />
<SessionId cookiesFirst="true" noCookies="false" />
<!-- ===== Connectors ===== -->
  <!-- new http adapter. Attributes:
    secure - use SSL ( https )
    keystore, keypass - certs for SSL
    port -->
  <!-- Uncomment this for Tomcat Standalone support -->
  <!--
  <Http10Interceptor port="8080"
    secure="false"
    maxThreads="100"
    maxSpareThreads="50"
    minSpareThreads="10" />
  -->
  <!-- Apache AJP12 support. This is also used to shut down tomcat.
    Parameter "address" defines network interface this Interceptor
    "binds" to. Delete it if you want to "bind" to all interfaces.
    9007 is non-standard so reference jk/nwworkers.properties in
    plugins.
  -->
<RequestInterceptor
  className="org.apache.tomcat.modules.server.Ajp12Interceptor"
  address="127.0.0.1"
  port="9007" />
  <!-- Apache AJP13 support (mod_jk)
    Parameter "address" defines network interface this Interceptor
    "binds" to. Delete it if you want to "bind" to all interfaces.
    9009 is non-standard so reference jk/nwworkers.properties in
    plugins.
  -->
  <RequestInterceptor
  className="org.apache.tomcat.modules.server.Ajp13Interceptor"
  address="127.0.0.1"
  port="9009" />
  <!--
  Context definitions can be placed here ( not recommended ) or
  in separate files. The ContextXmlReader will read all context
  definitions ( you can customize the "base" filename ).

  The default for this file is conf/nwapps-[name].xml.

  See conf/apps-examples.xml and conf/apps-admin.xml
  -->
</ContextManager>
</Server>

```

## MYSERVER.XML LAB

- Replace the ports 8080, 9007, and 9009 with ports 8090, 9002, 9003 respectively.
- Change the NWAPPS and NWSERVER to MYAPPS and MYSERVER.

## NWAPPS-WEBAPP.XML

Now we will need to go take a look at the SYS:/TOMCAT/33/CONF directory. Here is where you will define your web applications directory structure. By default, NetWare 6 uses applications that are defined by nwapps-NameOfWebApp.xml. In our example we defined this as myapps-NameOfWebApp.xml. **\*\*NOTE\*\*** By default this is apps-NameOfWebApp.xml on other platforms.

Below is an example of the APPS-EXAMPLES.XML that is installed with Tomcat by default on NetWare 6. Since by default APPS-EXAMPLES.XML is not being used by NetWare 6, we will use it for our new web application. If you desire to create your own web application, you can follow this example.

### NWAPPS-MYAPP.XML LAB

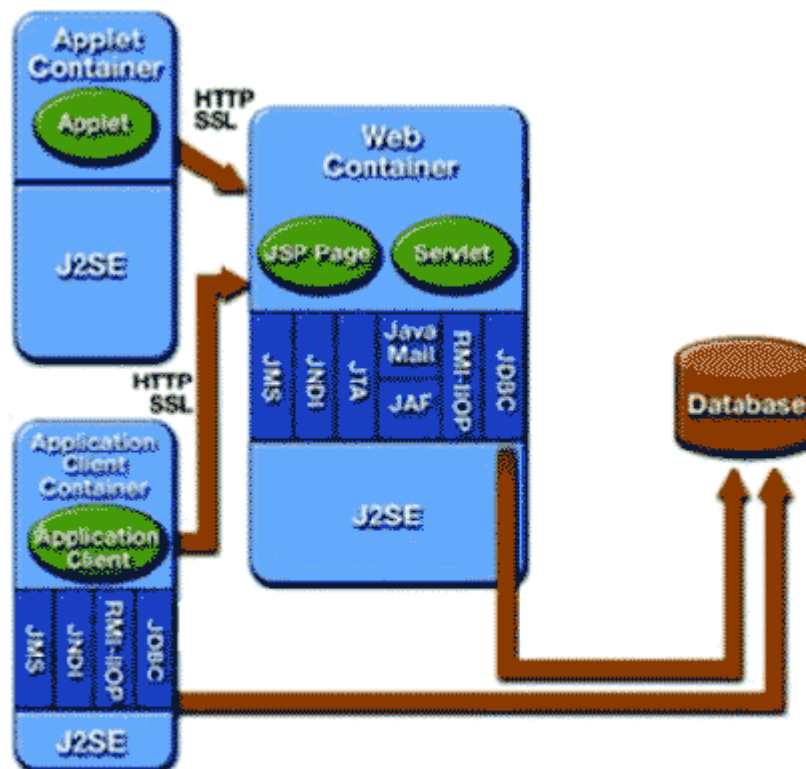
In order for us to use this we will need to create a copy of it and rename it as MYAPPS-EXAMPLES.XML.

1. The context path is listed as /examples. This is the URL that we will be using to access this web application.
2. The docBase is where all of the servlets, JSPs, html, class files, etc., will be placed for the web application.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<webapps>
  <!-- Setting special properties for /examples ( as an example of overriding the defaults )
  -->
  <Context path="/examples" docBase="webapps/examples" debug="0" reloadable="true" >
    <SimpleRealm filename="conf/users/example-users.xml" />
    <LogSetter name="example_tc.log" path="logs/examples.log" />
    <LogSetter name="example_servlet_log" path="logs/servlet_examples.log"
      servletLogger="true"/>
  </Context>
</webapps>
```

### Web Application

#### STRUCTURE AND LAYOUT



**J2EE BluePrint from SUN**

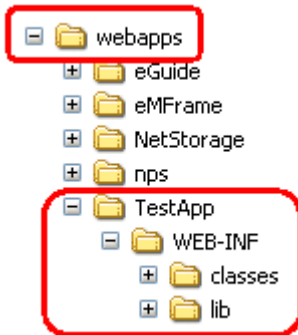
## CREATING A WEB APPLICATION

How to create a web application for Tomcat on NetWare 6.0

(1) The first thing that we need to do is to create the directory structure that will hold our web application.

Under the SYS:/WEBAPPS directory we will create a directory called TESTAPP. Under that directory we will need to create a WEB-INF directory. Then under the WEB-INF directory we will create a CLASSES and LIB directory. (see example below)

**\*\*NOTE\*\*** The CLASSES and LIB directory are not necessary for a Web Application, but we will be creating them for this example.



(2) Now we need to create a deployment descriptor (WEB.XML) file. The default code for this is listed below. Once the WEB.XML file is created, place it under the SYS:/WEBAPPS/TESTAPP/WEB-INF directory.

---

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
    "http://java.sun.com/j2ee/dtds/web-app_2.2.dtd">
<web-app>
</web-app>
```

---

(3) Now we need to add a ServletContext to our web application. The ServletContext will define the methods that the Web Applications components will use. In this example our ServletContext will be called MYAPPS-TESTAPP.XML. This will need to go to the SYS:/TOMCAT/33/CONF directory.

**\*\*NOTE\*\*** NWAPPS is the default descriptor that is defined for the default instance of Tomcat on NetWare 6. If you are running your own instance of Tomcat you could change this descriptor accordingly.

---

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<webapps>
    <!-- Setting special properties for TestApp context -->

    <Context path="/TestApp"
        docBase="SYS:/webapps/TestApp"
        debug="0"
        reloadable="false" >
    </Context>

</webapps>
```

---

(4) The next step is to create an INDEX.HTML file so we can test the context path and make sure it is working. We will be first testing this with Tomcat, since we have not yet integrated it with Apache. Tomcat has an HTTP listener that loads on 8080 if it is enabled. It is disabled by default. We will go and enable it.

**\*\*NOTE\*\*** One thing to keep in mind is that 8080 is the default port for BorderManager Proxy. If you are running anything on port 8080 then you will need to change the port that the Tomcat HTTP listener is running on. If you go

to SYS:/TOMCAT/33/CONF you will find a file called NWSERVER.XML. Open this file in a text editor and do a search for 8080. You will find the following section listed below.

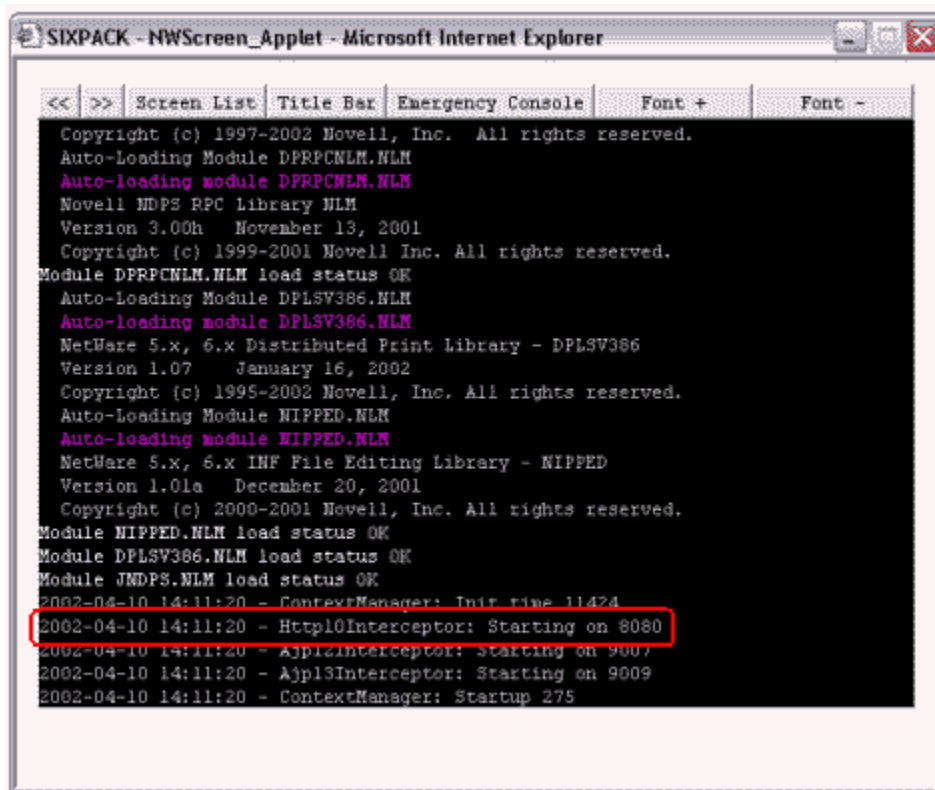
```
<!--  
<Http10Interceptor port="8080"  
    secure="false"  
    maxThreads="100"  
    maxSpareThreads="50"  
    minSpareThreads="10" />  
-->
```

Remove the beginning and end comment marks.

**\*\*NOTE\*\*** After removing the comment marks it will be necessary to restart Tomcat by doing the following:

```
JAVA -EXIT  
TOMCAT33
```

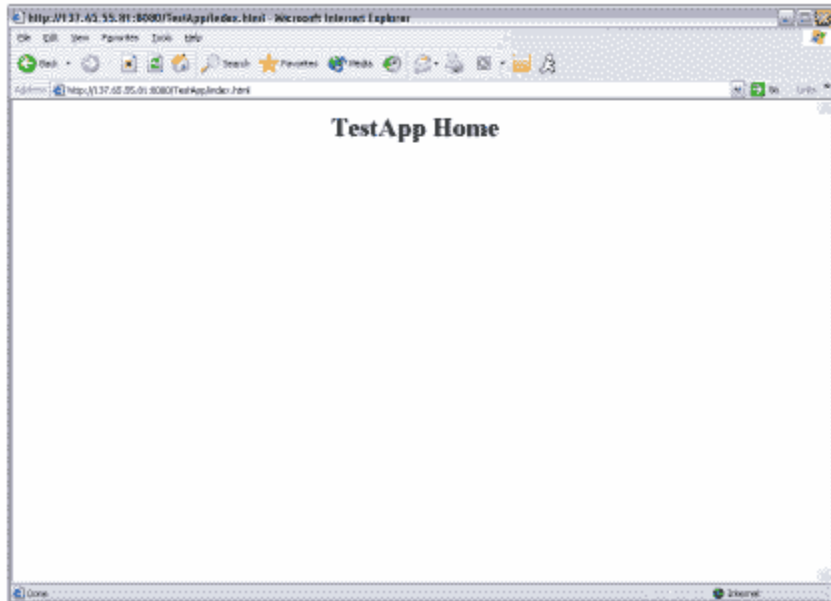
If everything loaded correctly you should be able to see that the port loaded in the logger screen:



(5) Now we will move on to creating a simple INDEX.HTML file as seen below:

```
<p align="center"><b><font size="6">TestApp Home</font></b></p>
```

We will save this out to the SYS:/WEBAPPS/TESTAPP directory as INDEX.HTML. Now if we go to <http://DomainName:8080/TestApp> we should see the following:



If you see this page then your ServletContext is working properly.

---

(6) Now we will create our first JSP (Java Server Page) to go into our Web Application. This will be saved to the SYS:/WEBAPPS/TESTAPP directory. We will name it DATEANDTIME.JSP. The contents are as follows:

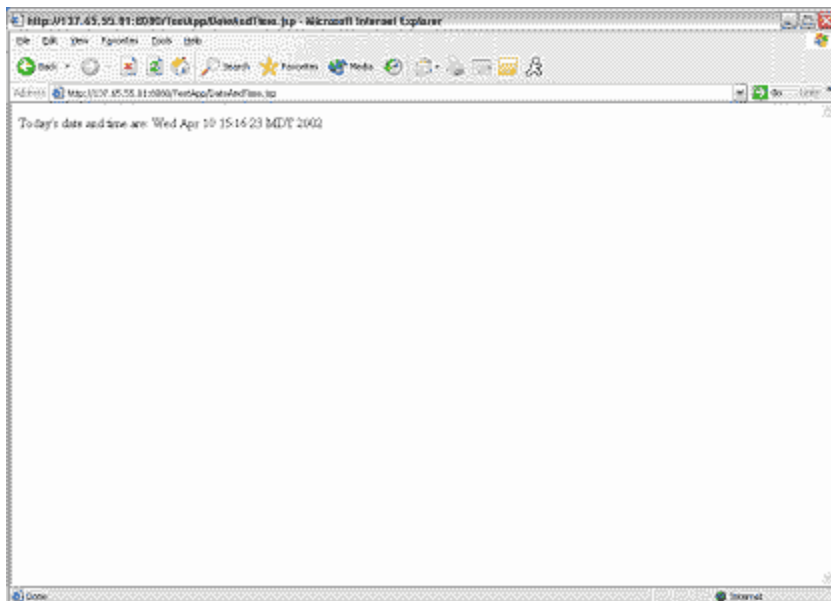
---

```
<HTML>
  <BODY>
    Today's date and time are: <%= new java.util.Date() %>
  </BODY>
</HTML>
```

---

Open a browser and go the following URL: <http://DomainName:8080/TestApp/DateAndTime.jsp>

If you see the following information then you know that your sample JSP is working.



---

(7) The next thing that we will do is to deploy a servlet to our Web Application. The first servlet that we will create will be called MYFIRSTSERVLET.JAVA and will need to be compiled into MYFIRSTSERVLET.CLASS

---

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

//
//
// MyFirstServlet
//
//
class MyFirstServlet extends HttpServlet
{
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException,
    IOException
    {
        // set content type
        res.setContentType("text/html");

        // Hello world
        PrintWriter out = res.getWriter();
        out.println("<H1>Hello World</H1> >");
        out.close();
    }

    public String getServletInfo()
    {
        return "My First Servlet";
    }
}
```

---

(8) After the servlet is compiled, place the MYFIRSTSERVLET.CLASS file in the SYS:/WEBAPPS/TESTAPP/WEB-INF/CLASSES directory. Next we will need to edit the WEB.XML file to add the information needed to run the servlet.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
    "http://java.sun.com/j2ee/dtds/web-app_2.2.dtd">
<web-app>
  <servlet>

    <servlet-name>
    MyFirstServlet
    </servlet-name>

    <servlet-class>
    MyFirstServlet
    </servlet-class>

  </servlet>

  <servlet-mapping>

    <servlet-name>
    MyFirstServlet
    </servlet-name>

    <url-pattern>
    /MyFirstServlet
```

```
</url-pattern>
</servlet-mapping>
</web-app>
```

---

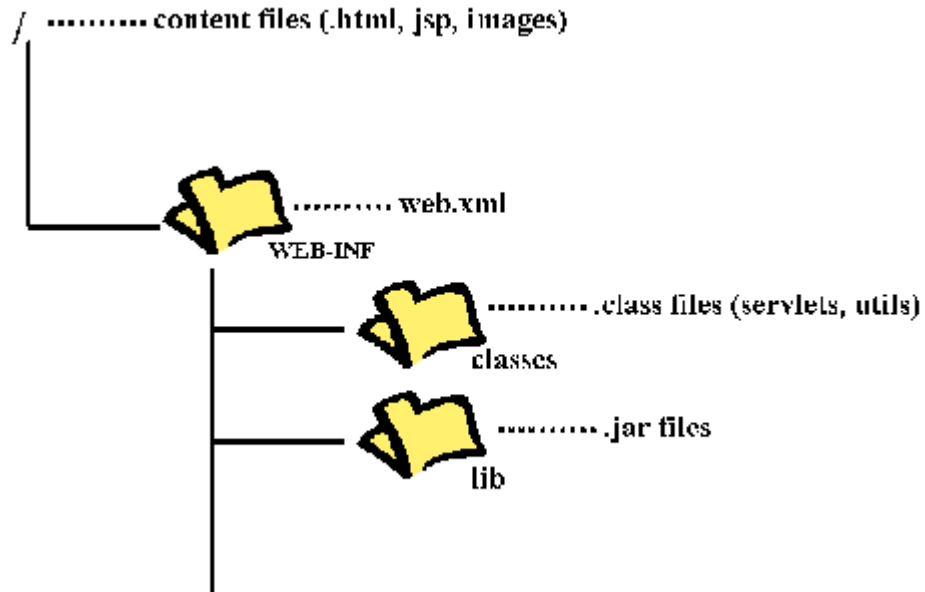
## WAR files

### Web Archives

- All components related to a web application are bundled together into a web archive file.
  - Web components include
    - HTML pages
    - JSP files
    - Servlets
    - JARs and supporting class files
    - Images, JavaScript files, stylesheet files, and more
    - Supporting classes

### Web archives are JAR files with the ".war" extension

- Definition
- How to Create
- How to deploy



## Upgrading from Tomcat 3.3.1 to Tomcat 4.0

**NOTE** - For the following examples it is assumed that your Tomcat was installed in the <C:\Program Files\Apache Group\Jakarta-Tomcat-3.3a> If this is not the case, you will need to modify the instructions accordingly.

- Move your <NPS> directory from the <Tomcat-3.3a\webapps\> directory to the <Tomcat-4.0.3\webapps\> directory
- Copy the mod\_jk.conf file from the <Tomcat-3.3a\conf\jk\> directory <Tomcat-3.3a\conf\jk\> directory to the <Tomcat-4.0.3\conf\> directory
- Copy the workers.properties file from the <Tomcat-3.3a\conf\jk\> directory <Tomcat-3.3a\conf\jk\> directory to the <Tomcat-4.0.3\conf\> directory

Modify the mod\_jk.conf on the line that starts out with <JkWorkersFile> to point the workers.properties that you just moved.

For example before the change it might read JkWorkersFile "C:\Program Files\Apache Group\Jakarta-Tomcat-3.3a\conf\jk\workers.properties.

And after the change it might read JkWorkersFile "C:\Program Files\Apache Group\Jakarta-Tomcat-4.0.3\conf\workers.properties.

- Note that Tomcat 4.0.3 does NOT have the <JK> directory under the <conf> directory anymore.
- Then do a search and replace for all instances of "C:\Program Files\Apache Group\Jakarta-Tomcat-3.3a\"
- Replace it with "C:\Program Files\Apache Group\Jakarta-Tomcat-4.0.3\"
- This is done in mod\_jk.conf. This will make sure that it is looking for the webapps in the correct directory structure.

Then in the Server.xml file in the < Tomcat-4.0.3\conf> directory find the section titled

<! - - Tomcat Examples Context - - >

In this section copy the lines that read

```
<Context Path = "/examples" docbase="examples" debug=0
  reloadable="true" crosscontext="true">
```

**below the following closing tags (About 56 lines down)**

\*\*\*\*\*

So

```
</ResourceParams>
```

```
</Context>
```

\*\*\*\*\*

Becomes

```
</ResourceParams>
```

```
</Context>
```

```
<Context Path = "/examples" docbase="examples" debug=0
  reloadable="true" crosscontext="true"> </Context>
```

Now edit these lines to read

```
<Context Path = "/nps " docbase= "C:\Program Files\Apache Group\Jakarta-Tomcat-4.0.3\webapps\nps" debug=0
reloadable="false" crosscontext="true">
```

Note - The docbase Path should reflect the change to  
"C:\Program Files\Apache Group\Jakarta-Tomcat-4.0.3\webapps\nps"

Finally, go into your Apache <conf> directory.

Example:

```
C:\Program Files\Apache Group\Apache\conf
```

And edit the HTTPD.conf file.

At the very end of the file there should be a line like

```
Include "C:\Program Files\Apache Group\Jakarta-Tomcat-3.3a\conf\jk\mod_jk.conf"
```

This should be changed to

```
Include "C:\Program Files\Apache Group\Jakarta-Tomcat-4.0.3\conf\mod_jk.conf"
```

Make sure that you SAVED the changes to each file.

And that they are not just sitting opened with the changes present but unsaved.

At this point restart Tomcat and Apache and everything should be working.

- Please verify that the Portal is still functioning After these steps and Before you install SP1.

## Increasing the amount of RAM that Tomcat Can Use

(1) Open your TOMCAT33.NCF file found in the SYS:/TOMCAT/33/BIN directory. Here we will need to add two parameters to the file: one for the minimum heap size and one for the maximum heap size. The parameters are as follows:

```
-Xms128m -Xmx256m
```

(2) Now we will insert these parameters into the TOMCAT33.NCF file. At the bottom of the file you will find the following line:

```
java -envCWD=$TOMCAT_HOME -classpath $TOMCAT_CLASSPATH -Dtomcat.home=SYS:\tomcat\33
org.apache.tomcat.startup.Main -f sys:/tomcat/33/conf/nwserver.xml %1
```

(3) We will add the heap size parameters to this line, right after the "java" directive. The new line will look as follows.

```
java -Xms128m -Xmx256m -envCWD=$TOMCAT_HOME -classpath
$TOMCAT_CLASSPATH -Dtomcat.home=SYS:\tomcat\33
org.apache.tomcat.startup.Main -f sys:/tomcat/33/conf/nwserver.xml %1
```

**\*\* IMPORTANT \*\*** Each java process is allocated 512 meg of user space within the RAM that it can use. Some of that is used for java threads, socket communication, etc. So what is really available is not much more than 400 meg which can be allocated to a java process. This information can be seen at the server by typing in the command JAVA -SHOW and then finding out what the ID is for the Tomcat java process. Then at the server console you can type JAVA -SHOWMEMORY# with the # representing the Tomcat process ID. Then switch over to the logger screen and you should see how much memory is being allocated within the Tomcat process. An example is shown below:

```
JAVA -SHOW
```

```
Classname                               ID
=====
org.apache.tomcat.startup.main..... 390
```

```
JAVA -SHOWMEMORY390
```

```
Memory Statistics For Class: org.apache.tomcat.startup.Main
```

```
-----
Reserved Heap: 68161536
Committed Heap: 11124732
Reserved Virtual Memory Pool: 67108864
Committed Virtual Memory Pool: 11272192
NLM Data Memory: 225280
Per Thread Data And OS Stacks: 3403776
Virtual Memory Pool Overflow: 0
JVM Tracking Memory: 23381
Socket Communication Memory: 101616
-----
```

```
Total Committed Virtual Memory: 22622204
Total Physical Memory: 3528773
Total Committed JVM Memory: 26150977
```

**\*\* NOTE \*\*** The main statistics to look at here are the reserved heap and the committed heap. The reserved heap represents our -Xmx switch. This is the maximum memory that is allocated to the individual java process. What we show here by default is 68161536 which equates to 65 meg. The committed heap represents the -Xms switch. This is the minimum amount of memory that will be committed to the individual java process. In this instance we have 11124732 which equates to 8 meg of committed memory.

(4) To verify that our switches have changed our committed and reserved heap sizes we will need to take down java and then restart tomcat. See example below:

```
JAVA -EXIT
```

**\*\* IMPORTANT \*\*** If the console comes back with MODULE JAVA.NLM UNLOADED, then you can startup tomcat by typing in TOMCAT33 at the console prompts. If it comes back that it is still cleaning up resources in the background, and you have a console prompt, then you can type in JAVA -EXIT again to force java down.

(5) Once tomcat is started again then you can check the memory again and it should now show up.

```
Memory Statistics For Class: org.apache.tomcat.startup.Main
```

```
-----  
Reserved Heap: 272633856  
Committed Heap: 136314876
```

**\*\* TROUBLESHOOTING \*\*** If you do not show the new memory size then one of the following probably happened.

- (A) You reloaded Tomcat before JAVA was completely unloaded.
- (B) You specified a maximum heap (-Xmx) that was too large. The most that this can be is around -Xmx386m without additional configuration.
- (C) Your -Xms parameter is larger than your -Xmx parameter.

On NetWare 6 we have the option of loading the AUTOEXEC.BAT file with a -u switch to increase the amount of memory that can be used by the JVM. This switch can be used in conjunction with java heap parameters to dedicate more memory to memory-intensive Web Applications. Examples of memory-intensive web application on NetWare 6 are Novell Portal Services 1.5 and eGuide 2.0.

There are a few things that should be noted before using the -u parameter. The -u parameter allows you to specify a higher amount of memory than exists on your server. This could create problems if you are trying to specify memory that you don't have. So what should be done to avoid this? Well, don't exceed the amount of RAM that you have on your server with the -u parameter. In fact, you would be well advised not to go above three-quarters of the amount of RAM that you have. Remember that the JVM is not the only application on the server that will be using the RAM. Take care to leave some RAM available for other processes.

(6) Edit the AUTOEXEC.BAT file found at the root of the C:\ drive. It should look similar to the example below:

```
C:  
CD \NWSERVER  
SERVER
```

(7) After editing the file to add the -u switch, it should look similar to the example below, with the exception that the number value may be different depending on the desired amount of RAM needed for the JAVA process.

```
C:  
CD \NWSERVER  
SERVER -u1000000000
```

**\*\* NOTE \*\*** In this example we are specifying 1 GIG of RAM to be used. The parameter is measured in bytes. This will get past the limit specified in step (4). You should now be able to increase your heap size above 400 MEG.

## Debugging and Troubleshooting

Breakdown and separation of components

`http://137.65.215.65/nps/servlet/portal?render=on`

- `http`
- `137.65.215.65`
- `/nps`
- `/servlet/portal`
- `?`
- `render=on`

`https://137.65.215.72:2200/eMFrame/webacc?taskId=fw.AuthenticateForm&merge=fw.AuthForm`

- `https` - protocol secure site
- `137.65.215.72` - document root
- `:` - indicates that a port will follow
- `2200` - different port
- `/eMFrame` - could be a directory or an alias
- `/webacc` - could be a directory or an alias
- `?` - represents that parameters will be passed
- `taskId=fw.AuthenticateForm` - parameter
- `&` - joins more than one parameter
- `merge=fw.AuthForm` - parameter