Quick Tips
for
NetWare Directory Services

Tip #1
In NetWare 4.02 there are some significant changes in how the NDS install code (DSI.NLM) will function.

Herewith are the enhancements that you will see upon the release of NetWare 4.02. These will provide more data integrity during install/uninstall of servers within a NetWare 4 tree.

INSTALL

1. Partitions will not be created during the installation of a new server. Any new container objects will be placed in the partition in which they were created.
*** This will help with avoiding the proliferation of partitions around the network. ***

2. Replicas will only be added to the server if:
a) bindery files exist on the server (i.e. the server is a 3.11/3.12 upgrade)
b) the number of replicas of a partition is less than 3
**** In the event of no bindery emulation need on that server - replication traffic will be lessened with no replica on that server. ***

UNINSTALL (Remove Directory Services)

1. There will be a check to determine if it is safe to remove NDS from the server - which is done by:
a) all replicas must have a state of "ON" - i.e. fully functioning - not in a split or join state, etc.
b) all servers in the replica list must have a status of "UP" - no down servers or links to servers in the replica list
*** This will insure that the removal of NDS from a server is reliable ***

2. The Uninstall process will save a mapping of names to object ids. This will be stored in a file that can later be used during a re-install and thus the trustee rights will not be lost upon re-installation of the server.
*** This will insure that the object deletes are completed and that a re-installation can be performed without losing the trustee's for all objects ***

3. The Uninstall process will automatically delete the server object and all volume objects associated with that server.
*** This insures the full cleanup of the server objects ***

Overall this will make installation of NDS on servers and the removal of NDS from servers more reliable, more robust, and able to maintain synchronization with each operation.

Tip #2
Have you ever wanted to grant administrative rights to only a portion of a tree... and yet not allow that administrator to perform partitioning operations? If so read on....

You can prevent an administrator from performing partitioning operations by following these steps:

1. Create the container
2. If you desire - make it a partition root
3. Create the administrator user object in that container

Then grant the following rights to that administrator:

a) All entry rights except for Supervisor
b) All attribute rights
c) Only grant read and compare rights on the ACL
d) Verify that the user has NO write rights on ANY container ACL in the tree - especially higher in the tree - ROOT

This will then restrict the administrator from performing partition operations in the tree...

Tip #3

Question: What are the Pros and Cons of larger versus smaller partitions in an NDS tree?

**** Please Note - this is only ONE answer to a very difficult question - more will follow later

What size is a larger or smaller partition? I will give some numbers here - but realize that all numbers for partitions are combinatorial... Larger partitions would probably have more than 3 - 5,000 objects. Again - do not read the numbers as "HARD" limits (i.e. the old syndrome of only placing 8 replicas of a partition in the tree)...

Perhaps a calculating example is the best way to describe this issue.

First example:

30,000 user objects, 5,000 misc objects (containers, printers, queues) in a partition
Replicate that partition on 20 servers.
Assume that each object changes 6 attributes daily - one login will change 3 attributes.
There will be 3,600,000 changes in the network to keep this replica in sync.

> (calculated by multiplying the number of objects changing, by the changes, by the servers needed to sync with - i.e. 30,000 x 20 x 6 = 3,600,000)

Realize that this is a LARGE replica on a Large number of servers on a Large network.

Second example:

Break the partition into thirds (totals equal 30,000 users, 5,000 misc)
5,000 user objects, 1,500 misc - partition 1
10,000 user objects, 2,500 misc - partition 2
15,000 user objects, 1,000 misc - partition 3

Place partition 1 on 5 servers, partition 2 on 7 servers, partition 3 on 8 servers (total of 20 servers)
Assume each user object changes 6 attributes daily (same as above)
That means that there will be:
150,000 changes for partition 1
420,000 changes for partition 2
720,000 changes for partition 3

Total of 1,290,000 changes in the network. The synchronization traffic is **only 1/3** of the original setup. However, there are still the SAME number of servers and SAME number of objects.

Third example:

Break the partition into sixths (totals equal 30,000 users, 5,000 misc)
4,000 user objects, 500 misc - in partitions 1-4
7,000 user objects, 1,500 misc - in partitions 5-6

Place partitions 1-4 on 3 servers and partitions 5-6 on 4 servers (total of 20 servers)
Assume each user object changes 6 attributes daily (same as above)
That means that there will be:
72,000 changes for partitions 1-4
168,000 changes for partitions 5-6

Total of 624,000 changes in the network. The synchronization traffic is **only 1/6** of the original setup. However, there are still the SAME number of servers and SAME number of objects. Placing data "CLOSE" to the user (i.e. keep the partitions relatively small and on smaller numbers of servers) is important to keeping synchronization traffic to a minimum in the network.

** NOTE ** where you place partitions and how large or small they are will depend on the cost of links, need for access to global corporate data, bindery emulation. This is just a small "tip of the iceberg" to help you understand a little more about the possible cons of very large partitions.

This is just a first answer for a VERY complex problem.

Tip #4
What is the limit on number of objects within a container?
NDS Design class #530 indicates 100, while the advanced 4.x administration states 500.
Why the discrepancy?

There is not a limitation within the base NDS - displaying the objects within a container has posed problems in the past. With NetWare 4.01 the utility limitation was about 1,200 objects that could be effectively processed. With the release of 4.02 and 4.10 (using the same utilities) that limit has risen to around 12,000-15,000 objects per container.

Tip #5
A) Will the NetWare 4.02 NWAdmin include:
1) RENAME CONTAINER -No
2) Will it put the little mask beside aliased objects? - Yes.

B) Why is there any limit on the number of objects in a container?
As noted in Tip #4 - I mentioned what the limits were with NDS containers, however, I did not specify why. The reason that there are any limitations is that NDS conforms to the x.500 standard specifying that data sent back to the client will not be in a specified order. This is due to the fact that if the data were sorted - would it be in a Spanish, French, English, or Japanese order - each would be a different sort order. The utilities then need to sort the data (i.e. contents of a container) before displaying that data on the screen, hence, the limitations are based on the workstation CPU speed, RAM and disk space as well as the length of time that the user is willing to wait for a refresh on the display.

Tip #6
Question: (in response to NDS Tip #1)
What if a server is dead or a link is gone and I have to remove directory services; will there be a way to do it?

Answer: This is in reference to NetWare 4.02

If the SYS: volume on a server fails so that it must be physically replaced the the following steps should be followed:

1. Use Partition manager to view the replicas on that server (write them down)
2. If any replicas are masters then perform the following two steps for each replica that is a master
    Load DSRepair on a server that holds a copy of that replica
    Use DSRepair to designate that replica as a master
3. Use NWADMIN to either delete the directory map objects associated with the volumes on the "dead" server
    or modify the volume attribute associated with each directory map object referencing the "dead" server
4. Use NWADMIN to delete all volume objects associated with the "dead" server
    (not just the volume that has crashed)
5. Use Partition manager to delete the "dead" server object
6. Use DSTrace to verify that there are no synchronization errors on the partitions on the "dead" server
7. If DSTrace displays errors on the replica list then DSRepair to repair the replica list
8. Remove the old and install the new hard drive
9. Restore the NetWare 4 system files to the DOS partition
10. Use INSTALL to place the server back into NDS tree
11. Place all previous replicas on that server (from written list in step 1)
12. Restore disk data from backup tape
13. Restore trustee rights from backup tape
14. If restoring the trustee rights fails then recreate trustee assignments manually
15. Check DSTrace to verify no synchronization errors are occurring with the server previously "dead"

Tip #7
Question:
If the customer has 4.01 installed, applied Update CD Vol 1 & updated with 310 DS. Do they have to upgrade to 4.02 or are thay already at "4.02 ". **** YES ****

Would they be missing anything if they do not "upgrade to 4.02". **** YES ****
Do we tell them to upgrade? **** YES ****

Answer:
Reason - they do not have the improved install (see NDS Tip #1) and the newest version of NWADMIN and the rest of the client utils.

Tip #8
Many have asked what are the plans for interoperability with x.500, DCE (IBM/OSF), XNS (Xerox), DNS (DEC), Banyan, etc.

Herewith are some thoughts that might help you address the issues that you are faced with daily.
The goal of NDS is to interoperate with other directories. There are 4 levels of interoperability that need to be considered - (if you look at NDS and any other directory here are the options):

1) Two separate databases, no interoperability **Benefits** - not much.

2) Two separate databases, data is synchronized between the databases using the external synchronization routines and registering for events on the NetWare server. **Benefits** - provides for administrative synchronization between databases. Could provide for a single point of adminstration between the two databases (i.e. create an account in one - will propagate to the other database).

3) Two separate databases sharing some information or using x.500 technology for exchange of information. **Benefits** - provides for single signon between directories using either shared information or

the exchange of x.509 certificates. This would allow a user to enter a single id and password at the workstation for authentication to the network, with background authentication facilitating the authentication to other directories or services.

4) One database that houses the information globally.

Most of the world is currently utilizing option number 1. There are many vendors working with us to facilitate the implementation of option 2. Most email, databases, PIM's, PBX's and many others make use of some sort of directory or address book. These vendors are working to integrate their storage of addresses or ids with NDS. There is also much work being done to solidify our interoperability with x.500. DCE and x.500 are very important and there are several vendors working on portions of this story. Most of this involves gateways and multiple types of authentication for the user. Option 4 - probably will never happen - just too many directories today.

Tip #9
Did the problem of changing an internal IPX address on a 4.x server go away with a certain level of NDS? Are there any problems with doing this.?

Answer (from page 7 of the NetWare 4.02 NDS Guidelines):

Renaming a Server or Changing the IPX Address
Before you rename a server or change the IPX address, make sure you check the servers' synchronization states. Also be sure that the server contains a replica of the partition in which the server object is contained.

Follow these steps to change the server name or IPX address:
Edit the Autoexec.ncf file.
Down the server.
Restart the server.
Type Set dstrace=on.
Type +limberSet dstrace=.
Type Set dstrace=*l.
Toggle to the Directory Services screen.
Check the Directory Services screen for the messages Limber: start connectivity check and Limber: end connectivity check. These messages mean that you have changed the server name or IPX address safely.

To change a server's context in the tree, follow these steps:
Use NWAdmin or Netadmin to move the server object in the tree.
Go to a server console.
Type Set dstrace=on.
Type Set dstrace=+limber.
Type Set dstrace=*l.
Toggle to the directory services screen.
Check the DIRECTORY SERVICES screen for the messages Limber: start connectivity check and Limber: end connectivity check. These messages mean that you have changed the server name or IPX address safely.

Tip #10
Herewith is a discussion of the 4 types of replicas used within NDS. (The first 3 will be just a review for you...)

MASTER - This is a writable replica that can also handle partition operations. There is only one master replica per partition. Only one partition operation is valid at any time for a partition and the master

enforces that requirement. For all non- partition operations this replica is equivalent to a read/write replica.

READ/WRITE - This is a writable replica that like the master can be updated from the client. Both read/write and masters are valid for login and authentication requests.

READ/ONLY - This is a replica that can not be changed from the client. It will be updated with the changed data in the replica from another read/write or master. This replica can not be used for bindery emulation due to the fact that there must be a writable replica on the server for bindery users.

SUBORDINATE REFERENCE - This is a replica of the partition root which includes the replica list (ring). As a child partition, it will reside on every server that holds a copy of the parent partition but not of itself. (i.e. this replica exists "Everywhere the parent is and the child is not". This replica is used to facilitate tree connectivity (walking the tree during resolve name requests).

Final note - there are 4 types of replicas - not just 3.

Tip #11
Something that I had thought everyone was aware of... but that is not the case... that DS.NLM will not be bound into SERVER.EXE in NetWare 4.10. It will be stored on SYS:System and this will provide the ability to reload NDS on the fly. A smaller NLM (Dsloader) will be bound into Server for loading DS.NLM.

The advantages of this architecture will be that it will allow a new DS.NLM to be copied to multiple servers and then reloaded while the servers are still up. By using DSLoader to maintain the library connections, then DS can be unloaded and reloaded without taking the server down or losing the client connections. In fact the users currently connected to the server as DS is reloaded will not be affected. This will allow an administrator to keep all servers in the network running the same version of NDS easily.

Note - this reload will only be effective for DS.NLM in 4.10. Perhaps in later versions of NetWare there will be more NLM's that will operate in this manner to efficiently give the network a 7x24 uptime.

Tip #12
In a tree there may be many partitions, but this example will mention only two partitions A and B. Partition B is subordinate to A - that is A is the parent partition and B is the child partition. There will be 7 servers for this example numbered Srv1 - Srv7. Servers 1, 2, 3, and 5 hold an instance (replica) of partition A. Servers 3, 4, 5, 6, and 7 hold an instance (replica) of partition B.

Operations:
Split (create a new partition) If there is a container named C in partition B and a partition is created from that container, then there will an instance of partitions B and C on servers 3,4,5,6, and 7. This operation only requires the operation to traverse the network - all data for both partitions is located on all servers.

Join (merge with parent) If partition B is merged with A for a result of just one partition then there will be some traffic on the wire.
Servers 1 and 2 will receive an instance of the data in partition B
Servers 4, 6 and 7 will receive an instance of the data in partition A

Once all servers that will be participating in the join operation have a replica of both partitions then the join will procede. For this reason, the join operation may place data on the wire.

Hope that this provides some help on wire traffic differences between these two operations.

Tip #13

It has come to our attention that many companies are placing instances (replicas) of the root partition on all or most of their servers to faciltate faster tree walking/name resolution. We have seen several tree examples recently where there were N servers and a replica of the root partition was on all N servers. This is not a good design for several reasons. Reason 6 is probably the strongest reason for more efficient placement of the root partition.

Reason 1 - Partition operations (involving that partition) would require that every server be up and reachable. If one of those many servers were down or unreachable then the administrator could not "split or join" i.e. - create a new partition or merge with a partition with it's parent.

Reason 2 - Synchronization traffic will increase as more servers hold a copy of the root. This is an exponential problem. As objects or properties are changed or ACL's on the root, then all of that information will need to sync to each of the servers holding the replicas.

Reason 3 - Each of these servers holds the same replica list and thus must be in communication every other server. Time to synchronize will increase linearly as the number of servers increases.
Reason 4 - NDS cleanup process occur when all synchronization has been successful. If a server is down or unreachable then cleanup processes will not run until all servers have been reached and synchronized to that time in the database.

Reason 5 - Increased rights for the administrators to place a replica on each server will result in a decrease in security as more persons will need to have rights to place instances of the root replica on servers.

Reason 6 -"Every time a child partition of the root is updated (object and property changes, partitioning etc.) every server that holds an instance of the child's replica list (i.e. every server that holds a copy of root will either have a Master, R/W, R/O, or Subordinate Reference of the child partition). Since the SyncUpTo time vector exists on the replica list then ALL servers holding the child replica's (i.e. subordinate references also) will be contacted for every change in that replica to update the time vector value in the replica list. Thus if all servers hold the parent (i.e. root) then they all will be holding a replica for the child - thus they will all be contacted for changes to objects in the child partition. Remember that the subordinate replica contains a complete copy of the root-most container object for that partition. It holds all ACL's, attributes, partitioning information, timestamp values for the partition, pointers to all other instances of that partition, etc. It is much more than just a pointer.

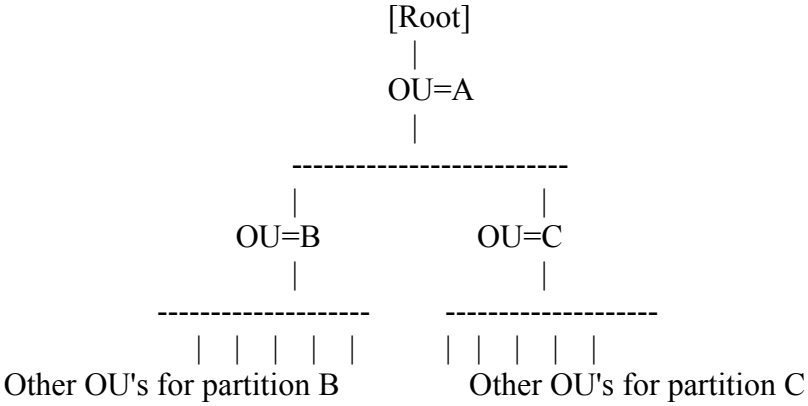
Solution - It is important to remember that there are tradeoffs in placement of the root partition. Many instances may bring faster name resolution, but may sacrifice performance on the WAN due to synchronization traffic. This could also cause administrative problems by requiring that all servers be in communication and less efficient security. Though it is important to place instances of the root partition close to the users to facilitate efficient tree walking - name resolution, it is not prudent to place an instance of the root on all servers in the tree. Instances of the root partition should be placed in each campus environment, hubs of the company, key locations - but not on every server in those locations.

Tip #14

In the past recommendations for large comanies deploying NW4 is that they create a "sub-admin" user that has all rights to a branch of the tree. The logic was that beyond managing users, etc. that they could add servers to their containers without using the Admin users login.

One difference with NW 4.02 is that if a partition (i.e. the root partition) does not have at least 3 replicas that when a new server is added to the tree a replica of that partition MUST be put on the new server. This requires the sub-admin to connect as the Admin user, since the sub-admin does not have rights to do this.

A solution to this difficulty would be to partition the tree for the sub-admins. When you create the sub-admin accounts it is a good policy to partition the tree so that the sub-admins own a partition root and the subtree beneath that partition root. If you have a tree with the following layout:



The administrator with rights to the root and containers A, B and C needs to perform the following steps:
1. Creates containers B and C
2. Create the B and C partitions using partition manager - they are now partition roots
3. Creates an administrator in each of the B and C containers.

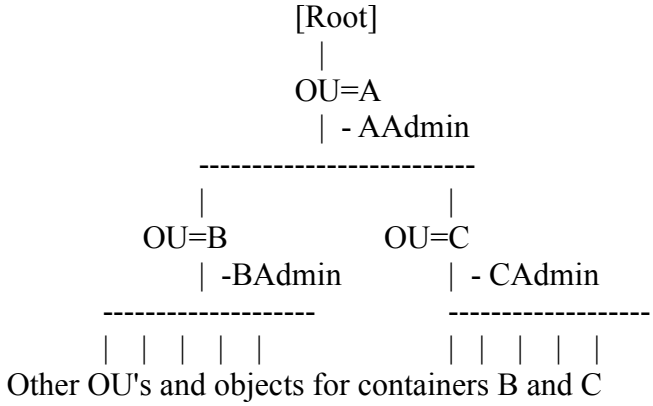
Then the sub-admins in B and C will have the authority to install servers into their respective partitions, place replicas upon those servers and thus manage that segment of the tree. Installing a server into partition B or C may or may not place a replica upon the server (based upon the install rules - i.e. bindery upgrade or less than 3 replicas). If the replica is needed then the administrator will have all rights to perform that operation. This management of the tree will not require the assistance of the administrator in container A. This allows the sub-admins in containers B and C to be independant of each other and also of the administrator at the root of the tree.

If container B or C is not a partition root then until there are 3 instances of the root partition administrative rights in the root partition will be required to install a server as install will place an instance (replica) of the partition on the server being installed.

Tip #15

I have had several questions come in concerning Administrative rights within NDS and the ability to mask those rights or to prohibit the masking of those rights.

In the following example of a tree:



There are three admin objects AAdmin, BAdmin, and CAdmin.

Question 1 - is there anyway for the AAdmin (administrator in container A) to prohibit admins BAdmin or CAdmin from restricting the rights for AAdmin to either containers B or C?

Answers

If the lower level administrative user has full rights to the container (i.e. BAdmin has write rights to container B) then prohibiting the BAdmin user from placing an IRF at container B and restricting the rights to that container and below for the AAdmin user is not possible. This is due to the architectural design of NDS to be able to place "firewalls" between administrators, business units, etc.

If the lower level administrative user only has partial rights to the container then the upper level administrator (AAdmin) could insure that he would never be blocked out of the lower container, however that would not give the BAdmin or CAdmin administrators full rights.

Question 2 - is there anyway for an upper level admin (i.e. AAdmin) to prohibit admins BAdmin or CAdmin from being able to create, modify or delete data in container A?

Answer

This can be done by giving the lower administrators (BAdmin or CAdmin) all rights to their respective containers and explicitly only giving them NO write rights at the root or A container for any object within that container. They could thereby browse, read, and compare in the upper container but would not be able to make any changes in that container.

Tip #16

Question on Bindery Emulation:

A number of 3.x programs want a SUPERVISOR account in Bindery mode. Is this the way to handle this?

i) Create a new container on the server
ii) Create supervisor account in this container
iii) Create a replica of a partition (containing this container) on the server

Can another server use the SUPERVISOR account created above?

Answer: **NO**

The supervisor is an object that only exists on a server when a bindery context is set for the server and the replica holding that context(s) is placed on the server.

Creating a supervisor account will create an object named supervisor in the directory but this account will not be available as the server supervisor in the bindery services (emulation) mode on the server. Thus there will be two accounts - separate and distinct called supervisor. If the bindery context is removed then access to all objects in bindery services on that server will disappear.

Tip #17

Question: What happens if a VLM client logs into a NW4.1 server using the /b (Bindery option)?

Answer: Login with the /b option will place the client in a bindery services mode of data retrieval from NDS. The effect will be the same as if the client had used NETX to login to the server. The user will only have access to the containers in the bindery context on that server. The user will not be able to run the NetWare 4 utilities (they require an NDS connection). This is indeed a bindery services connection.

Tip #18

Question:

If I install a server (ABC) into the tree and I do not place any replicas on ABC, how does NDS provide ABC

with access to the root of the tree. Is this accomplished by a subordinate reference to the root partition ?

Answer: If server (ABC) is installed into the tree with no replicas then there are also no subordinate references (no parent partitions, no sub refs - remember sub refs only exist where the parent is and the child is not). Finding of the tree from server ABC is done by using the NDS server client - which operates like the user client. It will use sap to find the nearest server in the tree and connect to that server. Using that server then the server client is able to walk the tree.
Even though server (ABC) has no replicas - it does have NDS data. Every server in the network has 4 NDS partitions (different from the logical partitions created within NDS). They are:
System - holds system and server specific data - no synchronization
Dynamic Bindery - holds SAP's - server centric, no synchronization
Schema - holds the schema for the tree - synchronizes to other schema partitions
External Reference - holds the external referenced objects on that server - no synchronization

Tip #19
As many, many of you have asked for assistance on NW 4 NDS error messages. Attached are the error codes for NDS for NetWare 4. There will be a later revision with more detail on solutions for each error. (see error code listing)

Tip #20
Question:
If server X does not have a replica of the root, and no servers are available with a replica of the root then can server X continue to be used for logon using NDS?

Answer:
If server X does not have a replica (see Tip #18) then there is no user data (i.e. public/private keys stored on that server). As such this server can not be used for initial login. In order to login a user/service must utilize a writable instance (replica - either master or r/w) of the user/service data. A read/only replica or a subordinate reference can not be used for login. A server with no data also can not be used for login. This is due to the fact that upon login there are 3 properties that are updated for the user/service.

Once a user/service is authenticated to the network, then a server with either a read/only, subordinate reference, or no replicas can also be used during the background authentication for additional services for the client.

One other thought... a server with no replicas does not sap the tree name - removing replicas can reduce network saps.

Tip #21
Question: Has anyone out there seen a NW4 installation where the user has chosen to have a dedicated server for NDS replica management? If so, what are the goods and bads or gotchas of having a single server be responsible for all replicas.....? Performance issues? Management issues? Configuration issues?
Fault tolerance issues?

Answer: Interesting question - we have had 5 requests in the last week for the scenario of having a single server or set of server be responsible for all replicas even though each replica might also exist on other servers in the tree. The largest of which had 3 central servers with 500 replicas on each of them (the same 500). Done correctly - this could be a decent design. We then took this to the superlab this last week for testing to prove or disprove our gut feelings. Herewith are our findings (advantages, disadvantages, actual results, and recommendations):

Advantages:
Having a central server makes backing up the NDS tree easier - the entire tree can be backed up from that one

server - no WAN traffic to locate objects in other replicas.
Tree walking on the central server is faster - on external or subordinate references.

Disadvantages:
That single server will have to communicate to **ALL** servers in the tree that hold any type of replica. It will be updating replicas, sub ref time stamps, external reference backlinks, etc on every server.
This server will have to receive **ALL** updates made in the tree (i.e. object creations, deletions, modifications, login timestamps, etc). This means that every change in the network will be replicated to this single server.
Execution of DSRepair on the server holding all replicas has the potential of consuming many hours.

Actual results:
Having a network of servers, each with it's own partition and replicated on a single server or set of servers caused difficulties when the number of servers and partitions reached certain thresholds. In striving for 500 replicas on one server - with one instance of each on a separate server meant that the outlying servers had 2.88 minutes each 24 hour period to communicate with the central server (i.e. 2.88 minutes x 500 servers = 86,400 seconds - one 24 hour period). The central servers were pentiums and each replica had 100 objects. During the testing, objects were added at the rate of 1 per second per replica. This means that with 300 replicas, every 300 seconds replica xxx would get an object. This is much more NDS activity than would happen on a normal business network - but was done to stress the environment. This was to simulate 500 small branch offices with a central site. As the number of servers installed reached 200, responsiveness from the central servers to the outlying servers became strained but still functioned efficiently. As the number of servers reached 300, servers experienced some communication problems due to timeouts. As server n was waiting to synchronize, there might be 10, 20 or 30 other servers ahead of it trying to talk to the central server. This caused timeouts on about 2% of the servers. As the number of servers reached 350 servers, LAN traffic was at 400 packets per second, with the wire utilization at about 12%. Very few errors, but thru put was hampered due to the number of packets on the wire. Installation of servers when under 250 performed as expected. Installation of the servers above 300 consumed much more time due to the fact that the central servers - with masters - was very busy synchronizing to other servers. A test of running DSRepair on the central server took over 1 hour for just the first phase of execution due to the number of objects (50,000+ objects, plus backlinks for servers, etc.).

Recommendations:
More detail later in the NDS Implementation Guide...
Having a server that houses the replicas of the tree does indeed have some benefits. Using moderation in the placement of replicas on that server is strongly encouraged! Placing replicas on a single server should follow these thoughts. Placement of 50-120 replicas upon a server with 100 to 3000 objects per replica with only 2 other instances of that replica upon other servers, will work very well. If the number of servers in the replica rings increases - then decrease the total number of replicas on the site server. If the number of objects increases then decrease the number of replicas... etc. If there are 500 replicas in the tree then place 1/5 of them on each of 5 servers at the central site, not 500 on one server... etc.

Tip #22
Questions: Can I run NetWare 4.01 and 4.02 in the same tree? Can I run 4.02 and 4.10 in the same tree?

Answers: Synchronization between 4.01, 4.02 and 4.10 will function correctly. I would heartily recommend that if anyone is on 4.01 (any version 291-310) that they upgrade to 4.02 as soon as possible.

The new functions of 4.10 will require the servers to be at 4.10 (obviously) to get the new functions (i.e. move subtree). Other than the new functions, 4.02 and 4.10 will interoperate and perform synchronization without missing a beat....

Tip #23
Question: In NetWare 3 there was a server centric "Guest" account - How do I accomplish that same function in

a NetWare 4 tree.

Answer:
1) There could be a guest account(s) set up in the NDS tree. Then rights could be granted to these users for access to resources (i.e. servers, volumes, services, etc) in the tree. This is the easy way out... and requires a separate account for guest priviledges. You can not have guest rights when authenticated as yourself.

2) A better solution with one caveat is to grant the rights that would have been granted to the guest users to public. This will give those "guest" rights to ALL connected users (including those that are NOT authenticated and NOT logged in). Thus a user access the resources (servers, volumes, services, etc) that are for guest use. The caveat is that NO authentication is required to access these resources and this could be a weak security policy for the NDS tree.

3) The best solution is to grant the rights to [root] rather than public. This will give those "guest" rights to ALL authenticated users in the tree. Thus a user can authenticate as themselves and access the resources (servers, volumes, services, etc) that are for guest use. This allows for flexibility for granting rights to public services without compromising any security.

Tip #24
Question: If I give an object supervisor rights to another object can I mask their rights to a specific property?

Answer: **NO**
If a object (i.e. user) is given supervisor rights to another object then that will give them supervisor property rights to all properties. This will supersede any specific property rights that may or might be granted. In other words be careful in granting the supervisor object rights as this grants supervisor rights to the properties of the object that can not be masked.

Tip #25
Question: Do we want to encourage companies developing NDS NetWare 4 trees to use the country object? Why or Why not? and if Why - then Why the change????...

Answer: If companies have a desire to interoperate with ANCS, the Internet, or any other x.500 name service that will use the NADF (North American Directory Forum), then we recommend that they build their directory tree with a country object. If they do not build with a country object today and then desire to associate themselves with any x.500 name service for data exchange or retrieval, their tree will need to be modified in order to incorporate the country object. This would mean a change to the tree - fairly minor to accomplish this, and a change to the clients net.cfg file - could be significant.

In order to provide a unique name in the NADF and x.500 environment it is imperative that names follow these standards. In order to interoperate with other directories a country object will be necessary. NDS (alone) will function just fine without a country object.

Summary: If a company plans on interoperating with other companies, name spaces, or the internet then building the tree with a country object at the top is advisable. I realize that this is a change from the past, but it is in our customers best interest to help them understand why they might want to design their tree in this manner and avoid the changes to the tree at a later date.
Why the change????
The reason that this has not been the case in the past is that the libraries with the utilities used a simplistic approach to resolving typeless names by assuming the top level was an Organization. With the release of the libraries and utilities for 4.10 this has been resolved. Now NDS is resolving the name rather than the client and can now handle the following structure (container) objects:
C - country, S - state, L - locality, O - organization, OU - organizational unit

The current libraries and utilities will efficiently support country, organization, and organizational unit. If state or locality are created by other utilities then the 4.10 utils will display the information. Future work provides utilities and other applications that support all of the x.500 naming structures that are in the 4.10 NDS.

Tip #26
Question: Can a user login through an alias in the bindery? in the Directory? If so, where will the user get container login scripts from? Can rights be given to an alias for the user?

Answer: An alias are not valid objects in the bindery and as such are not valid in bindery services. In the Directory aliases are in great use. An alias is merely a pointer to a real object (server, user, queue, etc) in the Directory. The alias can be used for login/authentication. When this happens the user will use the alias as a pointer to get to the real object. Container login scripts will be executed from the containers of the distinguished name for the real object, not for the alias. The effective rights for the user will be derived from the rights for the real object and the containers in the fully distinguished name of the real object. Rights associated with the alias are not part of the effective rights for the user.

Note -For those that might be familiar with this functionality in 4.01. This was different due to the way that Login.exe functioned. That has been corrected for the utilities shipped with 4.02 and 4.10.

Tip #27
Question: What are the benefits of using an alias?

Answer: Aliases are very beneficial for making the tree structure more usable for clients. An alias contains very little data - hence replication traffic is only the name and some data for a pointer to the real object. Aliases change infrequently. They only change when a) they are created b) they are deleted or c) the real object is renamed, deleted, or renamed. Here are some ideas of where aliases could be used.

Alias of a user where the alias is placed near the top of the tree. This helps the mobile computer user during authentication. Provides less that the user has to remember when away from his desk.
Alias of a printer, print queue, server, server volume in a container. Placing aliases of real objects that exist elsewhere in the tree in the user's container enables the user to locate those services without tree walking - without leaving that container.
Alias of a container provides the capability for ease of transitions. During the rename container or move subtree there is an option to leave an alias for the container. This allows for administrators to perform these operations within the tree and not affect the users. Realize that if this alias is opened in the utilities, you will see all of the objects beneath that container (i.e. leaf objects and other containers). These other objects are the real objects. If you delete a user (thinking that it is just an alias) the real object is deleted.

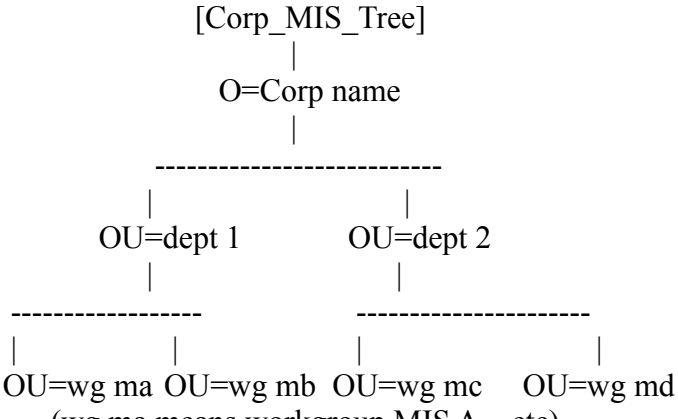
Tip #28
One difficulty with using an alias. If you backup an alias and then restore the alias, it will be restored as a real object and not as an alias. The problem is that an alias is backed up as a real object. This has being opened as an SPD and is planned to be fixed in 4.1.

Solution: They are resolving this even as I write! The problem was that the backup code was de-referencing the alias and thus it is a real object upon restore.

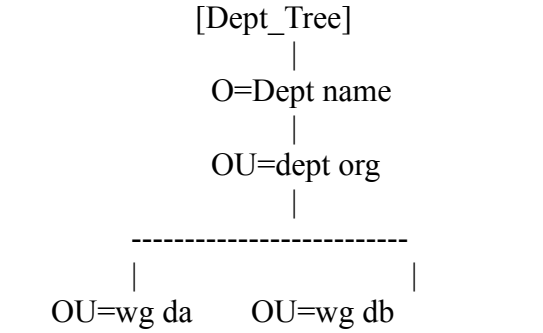
Tip #29
What is the recommended procedure for building directory trees that will be merged (when possible?) Do the highest levels need to be named distinct and different?. Is this really an issue with merge and prune/graft? If a customer has setup a network with multiple same-name trees, will the merge tree command allows the trees be consolidated? Must trees have unique name before merge? How to rename tree?

Answer: A few thoughts about NDS in general. Trees operating on the same wire network should have different tree names. NDS will not allow you to create a tree if one already exists on the network. However, it is possible to create two trees on two physically separate networks and then connect the networks. NDS does not have a problem with this, but confusion for the clients might be inevitable. This is due to the fact that clients will connect to the named tree, however, which tree they actually connect to is determined by the server that responds first for the named tree. If that is not "their" tree, then there will not be a user object for them in that tree. Realize that the Merge utility allows you to rename a tree.

Later work when merging trees can be lessened by following these steps.



(wg ma means workgroup MIS A... etc)
This is the destination tree.



This is the source tree that will be merged

Note the addition of the OU=dept org in the tree to be merged. The O=Dept name container should have very few objects. The following steps explain why:
1) Merge dept tree with MIS tree
2) This creates one tree named Corp_MIS_Tree
3) There are 2 O's directly under the root of the tree namely O=Corp name and O=Dept name
More than likely the corporation will not want 2 O's.
4) Use move subtree to move the OU=dept org under O=Corp name (becomes a sibling to OU=dept 1 and OU=dept 2).
5) This places the departmental tree at the correct level in the corp tree.
If the merge tree does not have the O=Dept name then all dept ou's (OU=wg da and OU=wg db) will need to be moved individually. This causes more work for the administrator.

Summary:
Trees on the same wire should not be named the same.
Highest level container in the trees (either C or O) must be different before renaming a tree.
Merge tree ONLY merges at the root of the tree
Move subtree can then be used to manipulate the tree

Helpful hint:
If two different named trees are created with the same name highest container. Net.cfg files will have the name of the tree and the default name context. As the merge is initiated, rename the highest level container to a new name, merge the tree, move the subtree to just below the highest level container of the destination tree, delete the highest level container of the source tree, and then the default name context for the users will not need to be changed in Net.cfg. Tree name will need to be changed for those users in the source tree in Net.cfg.

Tip #30
Question: We need more details on NetSync.

Answer: It is important to realize that NetSync is an administrative tool. It is used to administer a NetWare 3 and NetWare 4 mixed environment from the NetWare 4 utilities. After the NetWare 3 server bindery data is "uploaded" to the NetWare 4 server, all future additions, deletions, modifications to the Directory data should be performed from the NetWare 4 utilities. If a user is added using syscon on a NetWare 3 server then the data will need to be "re-uploaded" from the NetWare 3 server again. The ONLY data that is bi-directionally synchronized is the password data. All changes made to users, groups, print servers and queues, from the NetWare 4 Directory will be synchronized to other NetWare 4 servers holding that partition and to all NetWare 3 servers that are "NetSync'ed" to any of those NW 4 servers. A single NetWare 4 server can synchronize up to 12 NetWare 3 servers.

Tip #31
Question: Why can't I use a password from an NDS object to unlock the server console within Monitor.NLM when using RCONSOLE. It currently uses the password from supervisor which requires bindery services on that server.

Answer: Monitor is server centric for NetWare 4. As such it uses the Supervisor object (and password) for any authentication. Allowing Monitor to use the password from an NDS object that has Supervisor rights on the file server(which could be many NDS user objects) is something that is under advisement at the moment.

Tip #32
Question: What SAP codes will be new to our WAN when we roll out NW4x and NDS?

Answer: New SAPs for NetWare 4 are:
Type 278h NDS Server - saps the tree name. Servers with no replicas will not issue this SAP.
Type 26Bh Time Server - Single Reference, Reference, and Primary servers SAP their time service. Secondaries do not SAP. By default TimeSync will SAP, however, if configured then all TimeSync SAPs can be eliminated.

Tip #33
Question: It has been mentioned that there were ways that you could login to the Directory without taking up any connections on any servers.... HOW? But to browse the tree (e.g. access and open the Directory I'd think you'd have to be logged in...)

Answer: The key here is the definition of the types of connections. Here are the definitions of the three types of connections.

Not_Logged_In - this is a connection from a client to a server. There has been no authentication data (username/private-public key) exchanged between the client and server. It is merely a physical connection and is usually the result of a server responding to a Get Nearest Server from Netx or the VLM's. From this connection the client may exercise any rights granted to Public in the Directory. This is usually just access to the Login directory on the SYS volume.

Authenticated - this is a connection where the client has retrieved the private key (through the exchange of username/password data). This connection is not licensed. There are no physical server resources (print queues, printers, drives mapped, etc) associated with this type of connection.

Licensed - this connection is an authenticated connection that is using a physical resource on the server and as such will consume 1 license from the pool of licenses for that server.

NDS connections are "free" - that is authenticated but not licensed. The way that you could browse the Directory without taking up a licensed connection would be to authenticate to the network. You must be logged in and authenticated to run the utility. This gives the client an authenticated but not licensed connection. Then

by having a copy of NWADMIN or some other browser (such as an appware application) on the client local drive (so that a MAP is not needed to a NetWare server), you could run the browser and walk the NDS tree. Because these connections are "free" the client will be able to walk the entire tree - across servers without being licensed.

Tip #34
Question: With the new functions in 4.10 can I move any subtree in an NDS tree?

Answer: The function of move subtree (or move partition as it is called within the utils) allows you to move a complete subtree within an NDS tree. It will allow you to move any container and all subordinate objects to a new location in the tree. This upper container needs to be the root of a partition. If that uppermost container is not the root of a partition, then the operation will create one for you. There is no limit to the number of containers and objects that may exist beneath the container that is the root of the subtree. However, there can not be any subordinate partitions to that container/partition root. If the partition to be moved has a subordinate partition, then a "join" operation (called merge with parent in the utils) could be performed before and then the move subtree is allowed.
Summary: You can move any subtree that is fully contained by a partition to anywhere else in the NDS tree so long as it does not violate the schema (i.e. an organization container can not be placed under an organizational unit container, etc).

Tip #35
Would it be advisable to create an organizational structure containing the country object but also create an alias of the Organization at the root to allow easier access and provide better handling of the NET.CFG.

Answer: This could be a good solution for a tree that is migrating from having the upper level of an Organization (O=Novell) to having a Country (C=US) with the O=Novell directly under the country object. Realize that it could be good for migrating but if the tree is being designed and you want a country object, implement the tree with the country object. The solution (in the above example) would be:

Original tree has O=Novell at the top, thus all Net.cfg files have the context set with O=Novell as the highest (rightmost) level in the tree.
1) Create a C=US under root (at the same level as O=Novell)
2) Move the O=Novell subtree (the entire original tree) to below the C=US object.
This would require one of two scenarios: 1) make the entire tree one partition for the move 2) move the lower partitions twice (like the chinese game children play - pyramids) with an intermediate move by following these steps a) create a temporary Org container b) move all subordinate partitions to that level c) move the upper partition (containing O=Novell) under C=US d) rebuild the tree by moving all of the subordinate partitions...
This second solution may be the only solution when the tree is too large to contain in a single partition.
3) When moving the O=Novell partition leave an alias there that is O=Novell and points to the new O=Novell.C=US object in the tree.
When users authenticate with their original context that had O=Novell as the highest level, they will find their real object by using the alias. Then they will authenticate to their object that ends with .O=Novell.C=US A container script could be written that would automatically update the users Net.cfg file by using the Ncupdate.exe. In this way, an administrator could implement a country object and not affect the usability of the tree for the users.
Due to the work of moving the subtree and affecting a change in all users Net.cfg files, this should only be used for a migration from a tree without a country. For designing a tree - implement this object up front if desired.

Tip #36
A good recommendation is to have admins enter a password when they load REMOTE.NLM. If there are

problems with NDS or if you loose your RCONSOLE session while running DSREPAIR.NLM, you can still access the server using RCONSOLE.

Tip #37
Great question on Netsync: We have determined that the snyc process doesn't make use of the bindery context path introduced with NW 4.1. All objects synced to NW 3.x servers have to be in the container specified first in the context path. Is this true?

Answer: Yes and No... When uploading from the NetWare 3 server (i.e. installing NetSync on the NW 3 server) it will sync all of the bindery objects to the first container specified in the multiple bindery path (context) on the NW 4 server. When syncing object changes from the NW 4 server to the NW 3 servers it will sync all objects in ALL of the containers in the multiple bindery path.

If there are 8 containers in the bindery path and this NW 4 server is sync'ing to 5 NW 3 servers, then the contents of all 8 containers would be in the binderies of each of the 5 servers. Any changes made to the objects in any of the 8 containers would sync to the 5 NW 3 servers.

Solution: To upload a NW 3 bindery to a different container than the first container, you will need to reset the multiple bindery path specifying the container you want to upload to as the first container. Then install NetSync on the NW 3 server which uploads the bindery. Then reset the multiple bindery path on the NW 4 server and now changes made to that container (even though it is not the first container) will be sync'ed to the NW 3 servers.

Tip #38
Question: How can I delete a subordinate reference replica? (a common question...)

Answer: Since a subref replica is placed "everywhere the parent IS and the child IS NOT" by the system, there is not a way to delete a subref replica directly. However, a subref can be deleted indirectly by removing the parent replica on that server. This will then delete the subref on that server. Just remember that administrators can neither directly create or delete subordinate reference replicas.

Tip #39
All of my users have Supervisor rights at root of the SYS volume... WHY?

This occurs when you have granted Supervisor rights to a container object, which would give all users in that container those rights (Supervisor) to all objects in that container. This would include all server objects. This right is the only right that transfers to the file system directly. Hence, your users have supervisor file rights to the root of every volume on each server in that container.

Tip #40
When you run dsrepair, does it go out to other servers to query/verify/validate NDS information (I would expect it does)? If so, what are the implications of running dsrepair on multiple servers? Should the user wait a while between dsrepair runs on different servers?

It doesn't actually go into the databases on other servers, but it will check the sync state of the other servers that hold replicas of partitions on that server. The only implication is if you are doing an operation locally that has the database locked and another server tries to check the sync state, you will get a 659 error on the server running dsrepair. The users should allow the changes to synchronize across the network before running dsrepair on additional servers.

Tip #41
What is the current opinion on the use of country code? Some people have stated that we should NOT use the

country code. One of the DS TIPS stated that we should use country code for future compatability, I would like the current stance.

The relies on re-reading Tip #25. Basically, If companies have a desire to interoperate with ANCS, the Internet, or any other x.500 name service that will use the NADF (North American Directory Forum), then we recommend that they build their directory tree with a country object. If they do not build with a country object today and then desire to associate themselves with any x.500 name service for data exchange or retrieval, their tree will need to be modified in order to incorporate the country object. This would mean a change to the tree - fairly minor to accomplish this, and a change to the clients net.cfg file - could be significant.

What this really means is that the customer needs to determine where they want their tree interoperate... with just the company, a few select companies, or the internet. Each company decision will be unique and some may want to hook into the internet in the future, and will opt for no country code at this time, but rather will convert their tree later. Others will build with a country code now and not have to update the Net.cfg files etc. in the future...

Will I still have to use Typefull naming if I use the country code ?

No - All name resolution is now done by NDS. NDS understands the full hierarchy of the naming structure and will understand all levels. In the past name resolution was performed on several levels (API, etc). With NDS performing this you can type in:

.CN=Joe.OU=Core.OU=Eng.O=Novell.L=Utah.C=US
or
Joe.Core.Eng.Novell.Utah.US

and NDS will resolve both the same! It can utilize both the locality and country containers.

Tip #42
Are there any other implications of having multiple trees on a company's internet other than having the clients attach to the wrong tree? I have a large account that is attempting to roll out 4.1 on their infrastructure. Guess what, they have three trees all with the same name. Are there any problems with Timesync in this environment?

Having multiple trees on the same (company) internet is not a problem IF all trees have unique names. There are companies with 100+ different named trees on the same wire - no problem! If the trees have the same name as in the question, then clients will attach to a wrong tree by getting attached to the first to respond server. BTW - the only way to create this situation is to create the trees on isolated networks and then connect them - the install utility will not allow two different trees to be named the same on the same wire. This could create alot of frustration as users attempt to authenticate, and are refused, they then re-boot their workstation and try again and behold, they get authenticated. The dilemma is that they won't know why it worked one time and didn't work the other three times. If a user sits in close proximity to a server in another tree, that user may never be able to get attached to the correct tree. I see this as a major headache for the corporate IS or Help desk... This will also create frustration when installing servers into the tree - you will install into the tree - but do you have the right tree, with the correct containers??

TimeSync is independant of tree names and in configuring or sapping timesync, then only place one reference and several primaries in the entire network (for all trees). Do not allow each tree to have it's own reference and/or primaries. The biggest problem that could occur will be if every tree has a reference server - then you have some real gotchas as different trees may have different times, and a server from tree XXX picks up time from a server in tree YYY.

Tip #43

What are 601 errors, What are the circumstances that 601 errors appear.

A very common error. It simply means that the object (entry) does not exist. It can occur for a variety of reasons - with the most common being during authentication or any other request from the client for an object name that is invalid.

Tip #44
The questions from the college bowl (approx 50 questions) are included. This was for the SE training college bowl and are just fundamental questions about NDS. Attached are the questions.

Tip #45
Could you provide me with a complete list of DSTrace options. Rather than simply enable DEBUG I would like to be able to provide filtering and specific options like *H, *U, etc.

Do you have such a thing?

I have received MANY requests for such a document. Here is the document that was handed out at Brainshare last year... Attached is the trace doc.