

# WINS

---

*The purpose of this document is to provide information pertaining to NetBIOS operations in a Microsoft Windows networking environment, specifically, the operation and integration of the WINS services. This document assumes a working knowledge of the TCP/IP protocol stack and associated utilities.*

By [Rebecca Kastl](#)

*Last Updated: Jul. 5, 1999*

---

## Table of Contents

- [What is WINS?: A quick definition](#)
- [NetBIOS Naming](#)
- [Resolving NetBIOS Names and NBT](#)
- [Resolving NetBIOS Name in a Routed Network Using WINS](#)
  - [NetBIOS Name Registration](#)
  - [NetBIOS Name Renewal](#)
  - [NetBIOS Name Release](#)
  - [NetBIOS Name Query and Resolution](#)
    - [NetBIOS Name Cache](#)
    - [LMHOSTS Files](#)
    - [HOSTS Files](#)
    - [NBT Node Types](#)
    - [WINS Query Process](#)
- [WINS Push-Pull Relationships](#)
- [Helping the WINS-Challenged: WINS Proxy](#)
- [Integrating WINS and DNS](#)
- [Troubleshooting WINS](#)
  - [Static WINS Entries](#)
  - [WINS Registry Entries](#)
  - [Service Pack Updates](#)
- [WINS and the Network Browser Services](#)
- [Bibliography and Reference](#)

---

## What is WINS?

WINS stands for Windows Internet Name Service.

In much the same way that DNS provides a method to map IP addresses to human readable names for UNIX based systems and the Internet, WINS provides a way to map IP addresses to human readable names for Microsoft Windows based systems.

The primary difference between DNS and WINS is that DNS is hierarchical (meaning that levels of authority can be delegated at various points in the DNS "tree") and resolves the hostname (a separate

name space from NetBIOS) to an IP address. DNS can also be integrated with other DNS "trees" under a centralized "root" or point of reference. Such is the structure of the Internet's DNS '.com', '.net', etc. namespaces.

WINS is what's known as a "flat" name space, meaning that it does not support true integration with other networks outside of the enterprise the way DNS does. Additionally, WINS only resolves NetBIOS names to IP addresses (which has nothing to do with the Internet DNS hierarchy).

WINS differs from the standard DNS in three significant ways.

First, WINS is dynamic. When a Microsoft-based computer is brought up on the network, it sends a broadcast message to all systems connected to the physical segment. This is a dynamic registration process whereby computers come and leave the network with little or no intervention required (see also 'NetBIOS Name Registration' below). With DNS, name to IP address mappings are held in a database file (a 'flat' file usually called 'named.hosts') and require someone to administer the database. DNS queries by a requestor called BIND, whereas with NetBIOS, the query is handled by broadcasting or via directed datagrams to a WINS server.

Second, a NetBIOS name cannot be resolved via the Internet name space (DNS) without additional software and configuration. Microsoft's DNS server does integrate with its WINS service, essentially telling DNS to use WINS as its 'named.hosts' file.

Third, NetBIOS names must be unique throughout the entire network while with DNS, duplicate names can be used (even within the same domain).

DNS enabled networks can be viewed as cities within states within a single country (and referenced in the manner 'chicago.illinois.united-states'). Using the same analogy, WINS enabled networks can be viewed as the houses existing on an island which cannot be referenced beyond the context of the island, i.e., Joe's House.

## **NetBIOS Naming**

To understand WINS, one needs to understand NetBIOS, and NBT.

NetBIOS is an older name space supported by IBM products and has its roots in the older LAN Manager form of networking. NetBIOS utilizes a broadcast method to announce the network interface to the rest of the systems participating in the network.

It should be noted that NetBIOS is *\*NOT\** NetBEUI. In much the same way that IP is the transport layer of TCP/IP, NetBEUI is (roughly) the transport layer in Microsoft/IBM networking products. NetBIOS is nothing more than a supported name space that utilizes the existing transport layer of the protocol stack in use. People who have worked extensively with IBM products have a tendency to lump NetBIOS and NetBEUI together and (mistakenly) refer to them interchangeably. It must be understood that these are two separate pieces of the MS network protocol model.

The definition of that name space is a 15 character human-readable name. Additionally, NetBIOS names contain a 16th digit/character stored in hex format. This 16th character is not readable; instead, it is used to identify the resource type.

A sample NetBIOS name of SERVER1 breaks down like so:

```
SERVER1      <0x1B>
^-----^
1-----15 16th char.
```

indicates a resource is named \\SERVER1 and that this resource is a Domain Master Browser (as indicated by the hex code '0x1B'). Different hex codes indicate different resource types. Below are listed the NetBIOS name registration codes in a WINS database:

<b>Registered NetBIOS Names for WINS</b>			
<b>Name</b>	<b>Number Hex (0x)</b>	<b>Type</b> U=Unique G=Group	<b>Usage</b>
<computer_name>	00	U	Workstation Service
<computer_name>	01	U	Messenger Service
<\\--__MSBROWSE__>	01	G	Master Browser
<computer_name>	03	U	Messenger Service
<computer_name>	06	U	RAS Server Service
<computer_name>	1F	U	NetDDE Service
<computer_name>	20	U	File Server Service
<computer_name>	21	U	RAS Client Service
<computer_name>	22	U	Microsoft Exchange Interchange (MSMail Connector)
<computer_name>	23	U	MS Exchange Store
<computer_name>	24	U	MS Exchange Directory
<computer_name>	30	U	Modem Sharing Server Service
<computer_name>	31	U	Modem Sharing Client Service
<computer_name>	43	U	SMS Clients Remote Control
<computer_name>	44	U	SMS Administrators Remote Control Tool
<computer_name>	45	U	SMS Clients Remote Chat
<computer_name>	46	U	SMS Clients Remote Transfer
<computer_name>	4C	U	DEC Pathworks TCPIP Service on Windows NT
<computer_name>	52	U	DEC Pathworks TCPIP Service on Windows NT
<computer_name>	87	U	Microsoft Exchange MTA
<computer_name>	6A	U	Microsoft Exchange IMC

<computer_name>	BE	U	Network Monitor Agent
<computer_name>	BF	U	Network Monitor Application
<username>	03	U	Messenger Service
<domain>	00	G	Domain Name
<domain>	1B	U	Domain Master Browser
<domain>	1C	G	Domain Controllers
<domain>	1D	U	Master Browser
<domain>	1E	G	Browser Service Elections
<INet~Services>	1C	G	IIS
<IS~computer_name>	00	U	IIS
<computer_name>	2B	U	Lotus Notes Server Service
<IRISMULTICAST>	2F	G	Lotus Notes
<IRISNAMESERVER>	33	G	Lotus Notes
<Forte_&#x24;ND800ZA>	20	U	DCA IrmaLan Server Service

It is important to note that NetBIOS names differ significantly from standard IP hostnames. A NetBIOS name may contain a space character (" ") as well as an underscore ("\_"). Neither of these characters are supported as an IP hostname (or DNS name). When naming Windows based systems, this should be taken into consideration.

When installing TCP/IP on a Windows NT machine, the installation process will check the NetBIOS name to see if it can be used for the IP hostname as well. If the name does not conform to the rules for IP hostnames, it will substitute an underscore "\_" with a hyphen "-", and will only use the NetBIOS name up to the first space " " and drop the rest.

## Resolving NetBIOS names and NBT

When a MS Network client wants to resolve a NetBIOS name, it sends a broadcast frame out to the entire segment asking "Who is \\SERVER1<0x1C>?".

\\SERVER1 recognizes that the client is asking for him and is looking for the Domain Controller service specifically (probably for user authentication). \\SERVER1 then replies to the requesting client "I'm \\SERVER1" and the connection between the two systems is setup.

(I actually glossed over a lot of the actual session setup. For more information on this, reference the SMB/CIFS resource page on this site.)

As you may have noticed, with all these systems registering and asking for each other, wouldn't things get a little "loud" on the network? Well, it gets even worse. Every new session begins with a broadcast frame (that must be processed by every computer on the segment). To make matters worse still, every system will announce itself every 15 minutes to let everyone know they are still there (this

is for browser purposes and will be addressed later). Add more than one protocol and systems announce themselves every 15 minutes for each protocol. Wait a minute, this is getting really loud. How does anyone get any work done?

\*\*\*\*\*

*\*NOTE: This includes switched environments as the broadcast frames are destined for all systems, effectively circumventing the purpose of a switched environment. Routers are the only devices that will not propagate such traffic to all segments.*

\*\*\*\*\*

\*\*\*\*\*

*\*NOTE: This is a legacy architecture from early IBM LAN Manager networks and should have been phased out years ago. Instead, for backwards compatibility reasons, the architecture has been left intact. With the release of NT 5.0, it is MS' intention to start to do away with this feature and move to the DNS model implemented by other NOS'. Although NT 5.0 will be the first MS network product that does not utilize NetBIOS inherently, the support will still be there. NetBIOS support will now be an option, and no longer a requirement.*

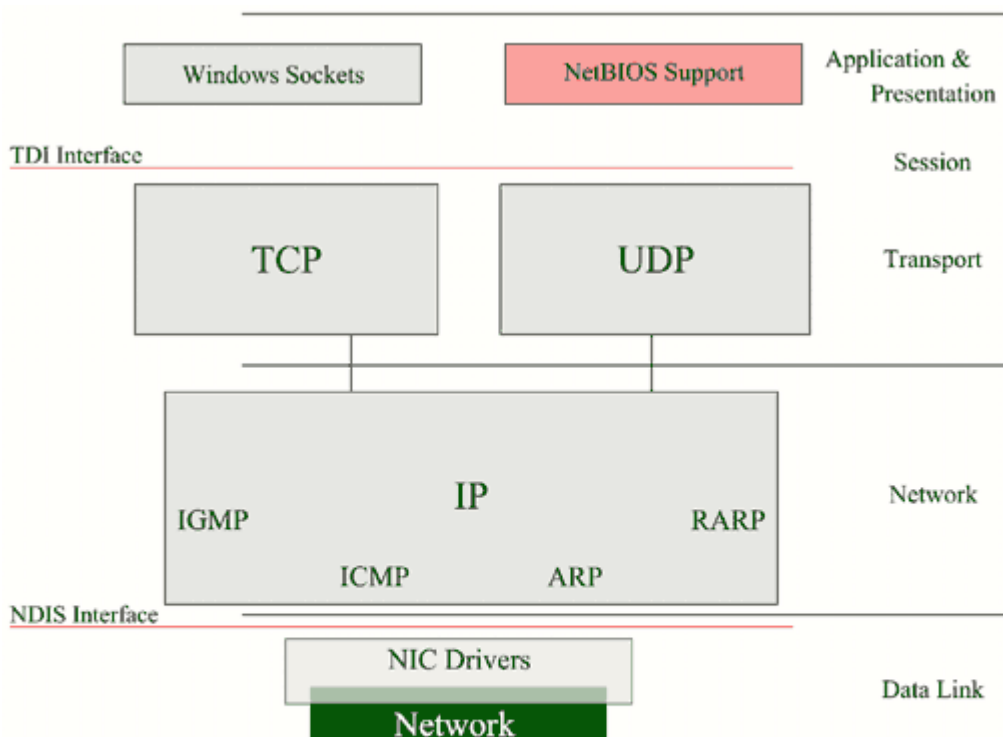
\*\*\*\*\*

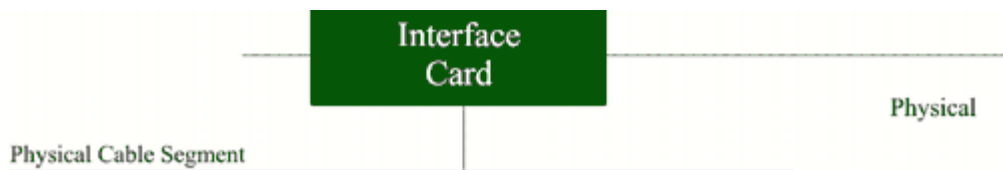
### NBT

For years, the underlying transport for NetBIOS has been NetBEUI (although it is not accurate to call NetBEUI a "transport" protocol in the sense of IP or IPX).

NBT stands for NetBIOS over TCP/IP (otherwise noted as NBF -- NetBIOS Frames). As part of the modular protocol stack implemented by MS called NDIS (Network Driver Interface Specification), the protocol stack becomes divided into different layers.

--MS TCP/IP and NBT Network Model:





Requests for NetBIOS-based resources are handed to the NBT "service", and then further down the protocol stack to be sent out on the wire. The NBT interface hides the specifics of the protocol stack from the requesting application.

The NetBIOS Interface service allows NetBIOS-based communications to occur in a MS network environment regardless of what the protocol stack is. Applications only need to know how to communicate with NetBIOS Support (at the {Application & Presentation} layer). They do not need to concern themselves with further details of the protocol stack. NBT encapsulates the NetBIOS data into a packet (or NBF) which is then handed to the connection method (TCP or UDP) and on down the stack from there. The 'Server' and 'Workstation' services are services that typically utilize NetBIOS.

This allows for MS network oriented applications to function in a TCP/IP environment and still continue to use the NetBIOS facility.

So NBT provides us with a way to send and receive NetBIOS traffic across routers. But how does a machine resolve a remote system's name to an IP address? NBT you say? Remember that with NetBIOS, name resolution is performed via broadcasts (which will not cross the router). The answer to this dilemma is WINS. It is the goal of WINS to provide a central authority in a routed network to which all systems look for NetBIOS name to IP address resolution.

## Resolving NetBIOS Names in a Routed Network

WINS is a service that runs on a Windows NT Server. The WINS service collects information from other computers on the network (via the NetBIOS name registration process) and their associated IP addresses and stores them in a database. Later, the WINS service will return this information to interested computers looking for specific services/systems (via the NetBIOS name resolution process). For this to happen, we need to understand several processes that occur: NetBIOS name registration, NetBIOS name renewal, NetBIOS name release, and finally, NetBIOS name resolution.

### NetBIOS name Registration Process

Okay, now that we see how NetBIOS packets are handled in a TCP/IP environment, we can look at how a system (configured to use WINS) registers itself with the network.

For the purpose of this discussion, we will assume three systems configured as follows:

#1	Windows NT Server NetBIOS Name: SERVER2 IP Address: 10.1.1.2 Subnet Mask: 255.255.0.0 Primary WINS: 10.2.1.2 NBT Node Type: 0x8 (Hybrid) Default Route: 10.1.1.1 Primary DNS: <none>
#2	Windows NT Server NetBIOS Name: NT-WINS IP Address: 10.2.1.2 Subnet Mask: 255.255.0.0 Primary WINS: 10.2.1.2 NBT Node Type: 0x8 (Hybrid) Default Route: 10.2.1.1 Primary DNS: <none>
#3	Windows NT Server NetBIOS Name: SERVER1 IP Address: 10.3.1.2 Subnet Mask: 255.255.0.0 Primary WINS: 10.2.1.2 NBT Node Type: 0x8 (Hybrid) Default Route: 10.3.1.1 Primary DNS: <none>

When SERVER2 boots up, it will initialize the protocol stack. Part of the initialization process includes registering with the defined WINS service (at 10.2.1.2). To register itself, SERVER2 will send a directed datagram to 10.2.1.2, using UDP, port 137 with a **name\_registration\_query** for the WINS service at 10.2.1.2 to register 'SERVER2<0x20>' in its database.

The WINS server can't just register any and all requests that come to it (otherwise we'd have complete chaos), so the WINS server needs to do some checking before registering the NetBIOS name in its database.

First, the WINS server will look in its database to see if the NetBIOS name is already registered. If the name is not already registered, the WINS server will register the name in its database and the name registration process is complete.

If the name is already registered, the WINS server will send a challenge to the currently registered owner of that name. If the WINS server does not receive a response to its challenge, it will retry three times, once every 500 milliseconds. If the currently registered system responds, the WINS server will send a **negative\_name\_query** response to the requesting client. If the WINS server does not receive a response from the current registered owner, the WINS server will then register the name in the WINS database by sending the requesting client a **positive\_name\_registration\_response**.

This covers two common scenarios with WINS:

1. You have re-addressed SERVER2 (for instance, moved it to a new subnet, thereby requiring a new IP address) and are re-registering the name with a new IP address, or
2. There is another machine on the network with the same NetBIOS name (which is not allowed).

\*\*\*\*\*

*\*NOTE: LAN Manager 2.2c for DOS and MS Network Client 3.0 WINS clients do not register NetBIOS names with a WINS server. WINS is only used as a resolver.*

\*\*\*\*\*

\*\*\*\*\*

*\*NOTE: If you examine the WINS database, you will see multiple entries for the same NetBIOS name. If you look closer, you will notice that they are not really the same because each one will have a different hex code suffix. It is common to see at least three entries for each machine on the network; one each for the 'Server', 'Workstation' and 'Computer Browser' services.*

\*\*\*\*\*

## NetBIOS Name Renewal

I'm going to skip this. This is a fairly esoteric and irrelevant issue of WINS. If you'd like to read more on this subject, you can find fairly detailed information in:

**Networking with MS TCP/IP: 2nd Edition** by Drew Haywood; New Riders Publishing, Copyright© 1997.

## NetBIOS Name Release

I'm going to skip this. This is also a fairly esoteric and irrelevant issue of WINS. If you'd like to read more on this subject, you can find fairly detailed information in:

**Networking with MS TCP/IP: 2nd Edition** by Drew Haywood; New Riders Publishing, Copyright© 1997.

## NetBIOS Name Query and Resolution

In a standard MS Network environment, when a system needs to locate a resource on the network (a domain controller, for instance), the client will send out a broadcast frame on the wire looking for the resource. Essentially, it sends a packet to every system connected to the same physical segment, asking "Where is SERVER1?". The problem is that each time a broadcast packet is sent out, it creates a software interrupt at every system listening on the wire, and creates unnecessary network traffic. How would you feel if every time one of your co-workers wanted to find Joe, he came and asked you "Are you Joe? I'm looking for Joe." After awhile, this could get very annoying, and you'd probably like to set up a map and show that idiot where in the hell Joe lives and tell him to stop bothering you.

Well, that is the function of WINS.

In a WINS environment (using the configuration above), when a system wants to locate another system on the network, it will query a WINS server by sending a UDP packet to port 137 containing a **name\_query\_request**. The WINS server will receive that packet and look in its database of systems for the associated NetBIOS name to IP address mapping.

So how does this happen? There is more going on here than just a system querying a WINS server



for a NetBIOS to IP address mapping. NetBIOS name resolution is a series of processes rather than just a single one. Let's look at our server above (SERVER2 @ 10.1.1.2) querying our WINS server (NT-WINS @ 10.2.1.2) for the IP address for SERVER1 @ 10.3.1.2.

Before we begin, you will need to understand a few terms that will be used here.

### NetBIOS Name Cache

The NetBIOS Name Cache is a memory area where MS networking products store NetBIOS name to IP address mappings that have been recently used. By default, such mappings are held for a period of about 600 seconds (10 minutes). After this time expires, if they haven't been used, they are expunged from the NetBIOS name cache. You can view the contents of the NetBIOS name cache by typing:

```
C:\> nbtstat -c
```

from the command prompt. We will discuss NBTSTAT further when we discuss troubleshooting tools later in this paper. You will probably see a message that says 'No names in cache.' This is okay (as long as we have a WINS server to query) for now.

### LMHOSTS file

A LMHOSTS file is a file that MS networking products use in much the same way as UNIX systems use a /etc/hosts file. This is an ASCII text file that maps NetBIOS names to IP addresses. While this solves the problem of resolving NetBIOS names in a routed environment, it can become an unwieldy administrative burden very quickly in an environment with a lot of computers coming off and on the network.

The format of an LMHOSTS file looks like this:

```
10.1.1.2      SERVER2
10.2.1.2      NT-WINS
10.3.1.2      SERVER1
```

with an IP address in the left hand column and the corresponding NetBIOS name in the right hand column. Should you desire to use LMHOSTS files, it is very important that this convention be adhered to (duh!).

*(Further discussion on [LMHOSTS files](#))*

### HOSTS file

As referenced above, a HOSTS file is equivalent to a /etc/hosts file you will see on UNIX based systems. This file has basically the same format as the LMHOSTS file, but there are some differences.

*(Further discussion on [HOSTS files](#))*

### NBT Node Type

All WINS clients are defined as a specific "node type". On an NT system, the node type can be found in the registry at:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBT\Parameters
Key: NodeType
```

Value Type: REG\_DWORD  
 Valid Range: 1, 2, 4, 8  
 Default: 8

On a Windows 95 machine, the node type can be found in the registry at:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\VXD\MSTCP
Key: NodeType
Value Type: REG_DWORD
Valid Range: 1, 2, 4, 8
Default: 8
```

The values of this key define different behavior for NetBIOS name resolution. The definitions of the various node types are as follows:

<b>b-node</b> <0x1>	b-node (broadcast-node) systems resolve NetBIOS names via subnet broadcasts only. (See <a href="#">RFC 1001</a> : Appendix A for information on extending a broadcast area beyond a single B-LAN through the use of IGMP. Note: This RFC is fairly dated (March, '87) and as soon as I locate an updated RFC, I will amend this page.)
<b>p-node</b> <0x2>	p-node (point-to-point-node) systems resolve NetBIOS names via point-to-point communications with a WINS server only and do not submit subnet broadcasts, even in the event of a name query failure.
<b>m-node</b> <0x4>	m-node (mixed-node) systems primarily employ a multicast (broadcast) method for locating systems. This is designed with the assumption that most systems will reside on the same subnet. In the event that a multicast request fails to resolve a NetBIOS name, an m-node system will query a WINS server.
<b>h-node</b> <0x8>	h-node (hybrid-node) systems are the default node-type in a WINS enabled environment. This node type will utilize point-to-point communications with its configured WINS server as its primary means of resolving NetBIOS names. In the event that the WINS server is unable to resolve a name (or is unavailable for some other reason) an h-node system will resort to subnet broadcasts for name resolution. This is the default node type for MS Windows systems.
<p>Miscellaneous notations regarding NetBIOS node types:</p> <ul style="list-style-type: none"> <li>• p-node and b-node systems do not inter-operate for NetBIOS name resolution, even on the same subnet.</li> <li>• End-node communications are not altered in any way with any of the node types.</li> <li>• All systems, regardless of node type, consult their NetBIOS name cache first before proceeding with further resolution methods.</li> </ul>	

First regardless of its configuration, SERVER2 will look in its NetBIOS name cache for the IP address of the system it wishes to communicate with.

Since the default node type of a WINS enabled client is h-node (and I fail to see any logic to configure your systems otherwise) we'll examine the details of h-node NetBIOS name

query/resolution and assume that all systems above are configured as h-node NBT clients (keep in mind that the name query/response algorithm will be different for other node types).

\*\*\*\*\*

*\*NOTE: There is also a complete [NetBIOS name resolution flow chart](#) that details how different node types resolve names using the various components discussed here.*

\*\*\*\*\*

Let's look at how SERVER2 queries NT-WINS for the IP address of SERVER1:

First SERVER2 will consult its NetBIOS name cache and see if it has the IP address information cached. As indicated above, this can be determined by typing:

**C:\> NBTSTAT -c**

at the command prompt. If an entry for SERVER1 appears in the returned name table, then the information is cached. If it does not appear, then SERVER2 must resort to other means of name resolution.

Since we are assuming that SERVER2 does not have the IP address information for SERVER1 cached, the next step is to query NT-WINS and see if it knows the IP address for SERVER1. SERVER2 will send a UDP packet to NT-WINS at port 137 (name services). At NT-WINS, the WINS server is listening on port 137. When NT-WINS receives the UDP packet, it will process it to the UDP level where it will see that the packet data contains a *get\_name\_query* request from SERVER2 for SERVER1.

On NT-WINS, the WINS service will look into its database of IP address-to-NetBIOS name mappings and discover an entry for SERVER1 at 10.3.1.2. The WINS service at NT-WINS will return a packet to SERVER2 with a *name\_query\_response* and the IP address for SERVER1.

Let's assume for a moment that SERVER1 did not register with the WINS service at NT-WINS. How would SERVER2 find the IP address for SERVER1 now? Well, if the WINS service at NT-WINS could not find an IP address-to-NetBIOS name mapping in its database, it would return a *requested\_name\_does\_not\_exist* response to SERVER2. At this point, SERVER2 would send out 3 b-node broadcasts to the local subnet.

Since SERVER1 is not on the same subnet as SERVER2, the broadcast attempts will fail. Now what? We're not done yet. SERVER2 will check to see if it is configured to use a [LMHOSTS](#) file. If SERVER2 is configured to use a LMHOSTS file, it will check that file for an IP address-to-NetBIOS name mapping.

What if there is no entry for SERVER1 in the LMHOSTS file, or there is no LMHOSTS file altogether? At this point, SERVER2 would again check its TCP/IP configuration to see if it is configured to use a DNS server. In the UNIX world, systems use a local file called `/etc/hosts` for "static" name-to-IP address mappings. NT was designed for this capability as well. Should all of the above methods of name resolution fail, NT will resort to a standard *gethostbyname()* call and query the [HOSTS](#) file. If this file does not contain an IP address mapping for SERVER1, then SERVER2 will query its designated DNS server using the same *gethostbyname()* call.

Here, things get a little tricky because NetBIOS names allow characters such as underscores ("\_") and spaces (" "), while DNS does not. Thankfully, that is not an issue here, but I will try to address it later.

If SERVER2 has not been able to resolve the IP address for SERVER1 by this point, then the name resolution will fail and an error will be returned to the calling process at SERVER2.

You may have noticed that I have gone to great lengths to state "WINS service at NT-WINS" instead of just saying "NT-WINS". There is a reason for this. Even a system that is running the WINS service **must** have its own IP address entered into the WINS configuration of its TCP/IP protocol information. Without this entry, the system will not be able to query its own WINS service to resolve NetBIOS names.

## WINS Push-Pull Relationships

You may or may not have noticed that the TCP/IP configuration in Microsoft networking products only allows for 2 WINS servers to be specified. Those familiar with the UNIX world will know that multiple DNS server entries can be placed in `/etc/resolv.conf`. It would seem that it is a shortcoming of WINS that only two WINS servers can be specified, but it isn't. This is merely a design feature.

In a large TCP/IP environment, it is desirable (if not sometimes necessary) to have more than one WINS server. In having multiple WINS servers, your servers can share information across domains, subnets, WANs, etc. This is accomplished by setting up your WINS servers as Push-Pull partners.

Using the WINS Manager utility, and selecting the '**Replication Partners**' from the '**Server**' menu, you can select other WINS servers to participate in your replication configuration.

I am not going to go into the details of Replication configuration. As far as what can be discussed, the topics range from the simple to the extremely complex. If you need to cover the complex issues of planning WINS replication, I suggest you look for an engineer with experience in this area.

## Helping the WINS-Challenged: WINS Proxy Clients

One of the primary limitations of WINS is that it requires a client be capable of querying the WINS server. In a non-routed environment this really isn't an issue (since NetBIOS name requests are broadcast to the local segment), but in a routed environment (where broadcast frames don't leave the local segment) this can create a problem.

Microsoft has made Windows systems (Win 3.x, Win 95 and WinNT) capable of becoming what is known as WINS Proxy clients. What this basically does is allow [RFC1001](#) compliant systems that cannot query a WINS server directly, to have their NetBIOS name requests forwarded by the WINS Proxy system to the WINS server and the response returned. Unfortunately, this doesn't help UNIX systems out at all (unless they are using SAMBA which introduces a whole different set of issues that I will not go into here).

## Integrating WINS and DNS

Microsoft has also enabled it's own DNS service to query WINS for hostnames. I'm going to forgo a detailed explanation of this feature for the time being. In the future I intend to add a very detailed explanation of what's going on here (maybe I'll get lucky and Windows 5.0/2000 will be released and I can forget about all of this WINS crap).

The short of it is that there is an option in Microsoft's DNS service administration GUI that allows you to point it at a WINS server where the name resolution is actually performed. The default behavior of this configuration is to convert the standard, all upper-case NetBIOS name into an all lower-case IP hostname and append the DNS domain name to it for resolution. This allows UNIX (and other non-MS systems) to query via standard DNS `gethostbyname()` functions for a name to IP address mapping without requiring additional software to be installed on the clients.

## **Troubleshooting WINS: Don't shoot the messenger**

OK, everything was running fine when you left work yesterday, but you come into work this morning and nothing's going right. Your systems can't seem to find the database server and London is reporting severe lag time trying to connect to your Chicago offices. Unfortunately, fixing WINS problems is not an easy task (even for the most experienced engineers).

### Static WINS Entries

One of the first steps to take in attempting to resolve any WINS problem is to remove any static entries. Period. Microsoft engineers have been repeatedly stumped when attempting to troubleshoot WINS problems, only to find out that the culprit is a static entry. Removal of static entries is usually the most efficient way to resolve most WINS problems.

One of the primary reasons that static entries cause so many problems is that they will prevent dynamically created WINS entries from being replicated for multi-homed hosts (since static entries take precedence over dynamic entries). In effect, a machine that is connected to two different subnets will not appear to certain domains or subnets because a static entry will point to the machine's IP address on a network that is unreachable (because of routing or security policies).

### WINS Registry Entries

There are a number of registry entries that govern the way that WINS manages its data. Most of these entries can be modified using the WINS Administration utility. This is the preferred method for modifying WINS parameters. For those who are interested, MS has published a [KB article](#) that details the registry entries that affect WINS.

### Service Pack Updates for WINS

With the release of Service Pack 4, Microsoft has provided an additional feature to the WINS Administration utility that has been long desired by administrators everywhere: the ability to delete single WINS entries and circumventing the tombstoning and aging processes. This allows for incorrect or stale WINS entries to be immediately removed from the WINS database.

## **WINS and the Network Browser Services**

Coming soon...

---

## **[Bibliography & Reference](#)**

---

Copyright© 1998-2001, Neohapsis, Inc. *Questions, comments or feedback, send E-mail to*  
[webmaster@neohapsis.com](mailto:webmaster@neohapsis.com)