Windows 2000 Server

Chapter 7 - Windows Internet Name Service

While Windows 2000 uses Domain Name System (DNS) as its primary method for matching a host name to its IP address, Windows 2000 also supports Windows Internet Name Service (WINS) for the same purpose. WINS is the name resolution system used for Windows NT Server 4.0 and earlier operating systems.

Windows 2000 DNS uses hierarchical fully qualified domain names (FQDNs) rather than the flat NetBIOS naming conventions supported by WINS. However, WINS provides an important service for network administrators with heterogeneous systems supporting clients running older operating systems, such as Windows 95 and Windows NT 4.0. These older systems do support DNS name resolution but do not support dynamic updates to DNS records.

In This Chapter

Related Information in the Resource Kit

- For more information about DNS, see "Introduction to DNS" in this book.
- For information about the Windows 2000 implementation of DNS, see "Windows 2000 DNS" in this book.

WINS Overview

While WINS servers are not needed in a network consisting entirely of Windows 2000–based computers, they are crucial for any network containing computers based on the older architectures of Windows NT 4.0, Windows 98, or Windows 95. This section describes the high-level architecture as well as the new features offered in this latest version of WINS. It also briefly covers the basic background of how WINS developed from the NetBIOS naming conventions of the 1980s.

New for Windows 2000

The new implementation of WINS for Windows 2000 provides the following enhancements:

Persistent Connections Now you can configure each WINS server to maintain a persistent connection with one or more replication partners. This increases the speed of replication and eliminates the overhead of opening and terminating connections.

Manual Tombstoning You can manually mark a record for eventual deletion, called tombstoning. The tombstone state of the record then replicates across all WINS servers, preventing an active copy on a different server database from propagating the record.

Improved Management Utility The WINS management console is fully integrated with the Microsoft Management Console (MMC), a user-friendly and powerful environment you can customize for efficiency. Because all server administrative utilities included with Windows 2000 Server are part of MMC, new MMC-based utilities are easier to use and faster to learn. MMC-based utilities operate more predictably and follow a common design.

Enhanced Filtering and Record Searching Improved filtering and new search functions help you locate records by showing only those records that fit the criteria you specify. These functions are particularly useful for analyzing very large WINS databases.

Dynamic Record Deletion and Multi-Select Dynamic record deletion and multi-select help you manage the WINS database. With the WINS management console, you can point, click, and delete one or more WINS static or dynamic entries. This function was not available in earlier command-based utilities for WINS administration (such as Winscl). You can also now delete records that use names containing non-alphanumeric characters.

Record Verification and Version Number Validation Record verification compares the IP addresses returned by a NetBIOS name query of different WINS servers. Version number validation examines the owner address–to–version number mapping tables. These features quickly check the consistency of names stored and replicated on your WINS servers.

Export Function You can export WINS data to a comma-delimited text file, which you can import or process with Microsoft Excel, reporting tools, scripting programs, or other programs for analysis and reporting.

Increased Fault Tolerance for Clients Clients running Windows 2000 or Windows 98 can specify a maximum of 12 WINS servers per interface (up from the earlier limit of two). The extra WINS server addresses are used only if the primary and secondary WINS servers fail to respond.

Dynamic Renewal of Clients A WINS client does not need to restart after it renews its registration of local NetBIOS names. Nbtstat includes a new option, **–RR**, which provides the ability to release and then renew a NetBIOS name registration. This feature of Nbtstat can also be used on WINS client computers running under Windows NT 4.0 that have been updated to Service Pack 4 or later.

Read-Only Console Access to the WINS Management Console WINS Setup automatically adds a special-purpose local users group, the WINS Users group, when WINS is installed. By adding members to this group, you can provide read-only access via the WINS management console to WINS-related information on this server computer for non-administrators. Membership allows a user to view—but not to modify—information and properties stored at a specific WINS server.

All of these features make Windows 2000 WINS the ideal choice for NetBIOS name resolution. WINS makes life easier for managers of routed networks, and solves the problems of internetwork name resolution in complex wide area networks (WANs).

Origins of WINS

Whether your network uses DNS or WINS, name resolution is an essential part of network administration. Name resolution allows you to search your network and connect to resources using names such as "myprinter" or "ourfileserver" rather than memorizing a host's Internet Protocol (IP) address. Remembering IP addresses would be even more impractical when using Dynamic Host Configuration Protocol (DHCP) for address assignment because the assignments may change overtime.

WINS is supported by DHCP services. Whenever the computer you named "fileserver01" is dynamically assigned a new IP address, the change is transparent. When you connect to fileserver01 from another node, you can use the name fileserver01 rather than the new IP address because WINS keeps track of the changing IP addresses associated with that name.

WINS was created to solve the problems of broadcast-based name resolution and the burden of maintaining LMHOSTS files. With LMHOSTS files, name resolution information is stored in a static format, making it a management-intensive chore to maintain. With broadcast-based name resolution systems such as NetBIOS, larger networks became more congested as hosts come online and broadcast messages to all other nodes to resolve IP addresses. In addition to the congestion, these broadcasts cannot cross routers,

meaning that names can only be resolved locally.

WINS is built on a protocol, defined by an Internet Engineering Task Force (IETF) Request for Comments (RFC) that performs name registration, resolution, and deregistration using unicast datagrams to NetBIOS name servers. This allows the system to work across routers and eliminates the need for an LMHOSTS file, restoring the dynamic nature of NetBIOS name resolution and allowing the system to work seamlessly with DHCP. For example, when dynamic addressing through DHCP creates new IP addresses for computers that move between subnets, the WINS database tracks the changes automatically.

The complete Windows 2000 WINS system includes a WINS server, clients, proxy agents, a WINS database, and a WINS management console. Each of these is described in this chapter.

WINS is compatible with the protocols defined for NetBIOS name servers (NBNS) in RFCs 1001 and 1002, so it is interoperable with other implementations of these RFCs. Another RFC-compliant implementation of the client can talk to the WINS server and, similarly, a Microsoft TCP/IP client can talk to other implementations of the NBNS. However, because the WINS server-to-server replication protocol is not specified in the standard, the WINS server does not interoperate with other implementations of NBNS. Data cannot be replicated between the WINS server and the non-WINS NBNS. Without replication, name resolution cannot be guaranteed.

NetBIOS Legacy of WINS

To understand the need for WINS, you must understand the history of NetBIOS, which started more than 10 years ago as a high-level programming language interface to IBM PC-Network broadband LANs for PC-DOS applications. Microsoft used this NetBIOS interface for designing its networking components. NetBIOS is a session-level interface that applications use to communicate over NetBIOS-compatible transports. It establishes logical names on the network, establishes sessions between two logical names on the network, and supports reliable data transfer between computers that have established a session. Protocols implemented under Microsoft networking components, including TCP/IP, include a NetBIOS interface or a mapping layer. This layer allows nonnative NetBIOS components to fit into a NetBIOS environment. NetBIOS-based communications use NetBIOS names to uniquely identify resources and other nodes on the network.

NetBIOS names are 16 bytes in length. The NetBIOS name space is flat, meaning that names can be used only once within a network. (DNS, in contrast, uses a fully qualified domain name (FQDN), which combines the host name with the name of its domain. A NetBIOS name such as "WINserver01" might be "WINserver01.itreskit.com" as an FQDN.) For more information about NetBIOS names, see "NetBIOS Names" later in this chapter.

NetBIOS names are registered dynamically when computers and services start and when users log on. A NetBIOS name can be registered as a unique name, which maps to a single address, or as a group name, which maps to multiple addresses. Each of these name types is discussed in "Microsoft WINS Servers" later in this chapter.

NetBIOS Name Resolution

NetBIOS name resolution is the process of successfully converting a NetBIOS name to an IP address. A NetBIOS name is a 16-byte address used to identify a NetBIOS resource on the network. A NetBIOS name is either a unique name, exclusive to a single process on a single computer, or a group name, which might address multiple processes on multiple computers.

An example of a process that uses a NetBIOS name is the File and Printer Sharing for Microsoft Networks service on a computer running Windows 2000. When your computer starts, File and Printer Sharing for Microsoft Networks registers a unique NetBIOS name based on the name of your computer. The name registered by the service is the 15-character computer name plus a 16th character of 0x20. If the computer name is not 15 characters long, it is padded with spaces to make it 15 characters long.

When you initiate a file-sharing connection by name to a computer running Windows 2000, that connection uses File and Printer Sharing for Microsoft Networks on the file server you specify. File and printer sharing always corresponds to a specific NetBIOS name. For example, when you attempt to connect to a computer called CORPSERVER, the NetBIOS name corresponding to File and Printer Sharing for Microsoft Networks on that computer is:

```
CORPSERVER [20]
```

Note the use of spaces to pad the computer name. Before you can establish a file and print sharing connection, a TCP connection must be created. To establish that TCP connection, the NetBIOS name CORPSERVER [20] must be resolved to an IP address.

The exact mechanism by which NetBIOS names are resolved to IP addresses depends on which NetBIOS node type is configured for the computer seeking to resolve a name. RFC 1001 defines the NetBIOS node types; they are also listed in Table 7.1.

Table 7.1 NetBIOS Node Types

NetBIOS name resolution mode	Description
B-node	Uses IP broadcast messages to register and resolve NetBIOS names to IP addresses. Windows 2000–based computers can use modified B-node name resolution.
P-node	Uses point-to-point communication with a NetBIOS name server (in Windows 2000–based networks, this is the WINS server) to register and resolve computer names to IP addresses.
M-node	Uses a mix of B-node and P-node communication to register and resolve NetBIOS names. M-node first uses broadcast resolution; then, if necessary, it uses a server query.
H-node	Uses a hybrid of B-node and P-node. An H-node computer always tries a server query first and uses broadcasts only if direct queries fail. Windows 2000–based computers are configured to use H-node by default. To reduce IP broadcasts, these computers use an LMHOSTS file to search for name–to–IP address mappings before using B-node IP broadcasts.

Computers running Windows 2000 use B-node name resolution by default and use H-node when configured with a WINS server.

In order for remote NetBIOS names to be resolved, you must configure your computers running Windows 2000 with the IP address of a WINS server. You must configure Active Directory–enabled computers running Windows 2000 with the IP address of a WINS server if they are to communicate with computers running Windows NT, Windows 2000, Windows 95, or Windows 98 that are not Active Directory enabled.

Broadcasts in NetBIOS Name Resolution

Name resolution in a NetBIOS network in a small and self-contained network is broadcast-based. A name registration request can be broadcast and heard by all
B-, H-, and M-nodes on the local network. If no objections are received, the application broadcasting the request assumes that it has permission to use the name and issues a name overwrite demand. If the name is already in use, a negative name registration response is sent by the node using the name. In this case, the requesting application does not have permission to use the name.

In a larger, interconnected series of subnets, broadcast-based name resolution creates certain problems. First, nodes may interact with one another within a broadcast area, but they cannot interact across routers in a routed network. Second, broadcasts for name resolution generate significant network traffic. Third, every node within the broadcast area must examine each broadcast datagram,

consuming resources on every node. Broadcast-based name resolution works fine within a small LAN, but as the LAN grows and merges into a WAN, this method is not effective. Large LANs experience bandwidth problems, and once routers are introduced, the broadcast-based name resolution becomes inoperable.

WINS avoids the problems of NetBIOS name resolution by providing dynamic database maintenance for name registration and resolution. WINS reduces broadcast traffic while allowing the clients to locate remote systems easily across local or wide-area networks.

LMHOSTS Files

The LMHOSTS file was introduced to assist with remote NetBIOS name resolution. The LMHOSTS file is a static, local database file that maps NetBIOS names to IP addresses. This is similar in functionality to the Hosts file in DNS, but the Hosts file is used for mapping IP addresses for host names in the hierarchical DNS namespace, rather than NetBIOS names. Recording a NetBIOS name and its IP address in the LMHOSTS file enables a node that cannot respond to name query broadcasts to resolve an IP address for that NetBIOS name.

When Windows 2000 uses an LMHOSTS file to resolve remote NetBIOS names, it examines the LMHOSTS file that is stored in the directory %*SystemRoot%*\System32\Drivers\Etc.

As noted earlier, a computer in a Microsoft-based network can resolve NetBIOS names in several different ways. If one method of resolution fails, the computer tries the next method in a fixed order. In a broadcast-based network, the node first checks its remote name cache before broadcasting a name query (the name will be in the cache if it has been used recently or loaded from LMHOSTS). As a last resort, the computer uses the LMHOSTS file to obtain the IP address assigned to the NetBIOS name it is trying to resolve (for example, to obtain the IP address for the name of a computer across a router in a broadcast-based network).

For more information about LMHOSTS files, see "LMHOSTS File" in this book.

Despite the many uses of the LMHOSTS file, its design imposes some limitations. Its greatest limitation is that it is a static file, which means that entries must be updated by hand if the name or the IP address of the computer changes (such as when a computer is moved to a new subnet, or when a remote user dials in and connects via Routing and Remote Access). This limitation of the LMHOSTS file is exacerbated by the introduction of DHCP. A DHCP server assigns IP addresses to nodes dynamically, making it nearly impossible to keep the LMHOSTS file updated.

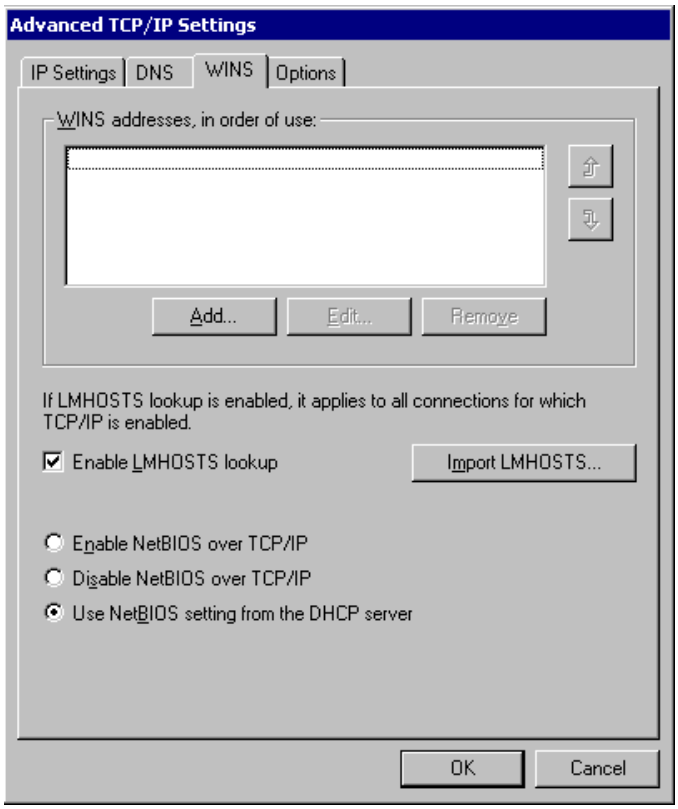
Continuing Need for WINS

If your network contains only computers running Windows 2000 or other TCP/IP-based systems that do not require the use of NetBIOS names (such as most versions of UNIX), your network no longer needs to use WINS. Instead, you should use Microsoft DNS service to resolve IP addresses.

However, many networks still include computers running Windows NT, Windows 98, and Windows 95. For these networks, WINS is needed to support earlier Windows and Microsoft TCP/IP clients. These networks will need WINS until all clients are migrated to Windows 2000.

Microsoft WINS Clients

To configure WINS clients with the IP address of one or more WINS servers, open **Network and Dial-up Connections** and click **Local Area Connections**. Click the **Properties** button, select the **Internet Protocol (TCP/IP) Properties** entry in the list, and click **Properties**, then click **Advanced** and select the **WINS Address** tab. Figure 7.1 illustrates this configuration page.



If your browser does not support inline frames, click here to view on a separate page.

Figure 7.1 WINS Service on a Windows 2000 Client

NetBIOS names are linked to the various network services that each client computer can use with other computers on the network.

Microsoft supports WINS clients on the following platforms:

- Windows 2000
- Windows NT Server

- Windows NT Workstation
- Windows 98
- Windows 95
- Windows for Workgroups
- LAN Manager 2.*x*

A WINS-enabled client communicates with the WINS server to:

- Register in the WINS database NetBIOS names of processes running on the client.
- Release from the WINS database the NetBIOS names of processes that are no longer running on the client.
- Renew client names in the WINS database.
- Resolve names by obtaining mappings for user names, NetBIOS names, DNS names, and IP addresses from the WINS database.

Clients that are not configured to use WINS can participate in these processes to a limited extent, but they must use WINS proxy agents to do so. For more information about proxy agents, see "Microsoft WINS Proxy" in this chapter. Each of the other tasks performed by a WINS client is described in this section.

How WINS Clients Register Their Names

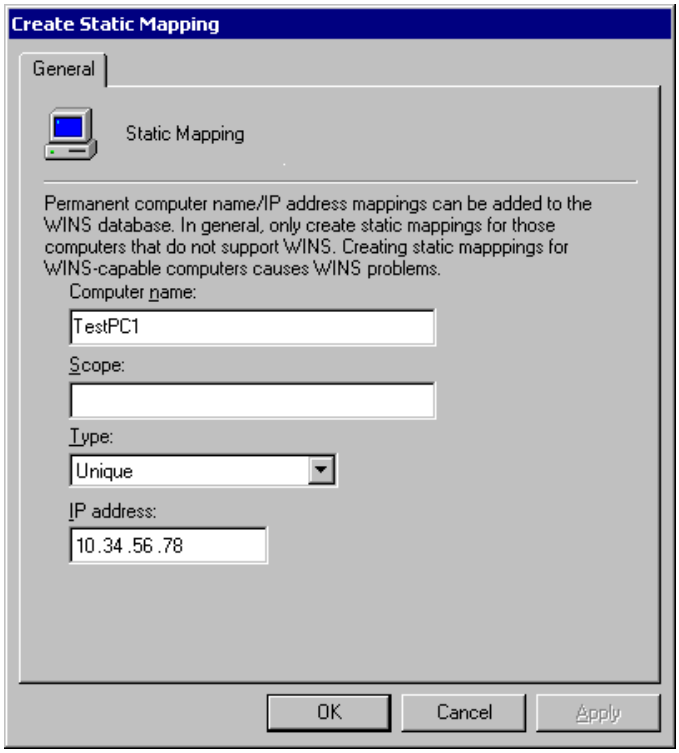
When a WINS-enabled computer starts, it attempts to register its NetBIOS names and corresponding IP address directly with the WINS server. If the registration fails, the WINS client tries again every 10 minutes until it is successful. The message the client sends is referred to as a name registration request. The WINS client sends one name registration request (which includes the computer IP address) for each NetBIOS-based networking service running on the computer.

Note that the IP address is dynamically assigned by a DHCP server if the client is DHCP-enabled. If DHCP is not used, the IP address is a statically assigned number which you must get from a network administrator and manually configure on the computer.

To create a static mapping with WINS

1. In the WINS management console, click **Active Registrations** in the console tree for the appropriate active WINS server.
2. On the **Action** menu, click **New Static**.
3. In the **Create Static Mapping** dialog box, type the static address in the **IP address** box.

Figure 7.2 shows the **Create Static Mapping** dialog box.



If your browser does not support inline frames, click here to view on a separate page.

Figure 7.2 Static Mapping in WINS

When the WINS server receives a name registration request for a unique NetBIOS name, it checks whether the name already exists in its WINS database. The WINS server responds with either a positive or negative name registration response. Table 7.2 describes each type of WINS server name registration response.

Table 7.2 WINS Server Responses

Server response	Explanation
No response	The WINS client sends another name registration request for the same name.
Positive	The WINS server does not find a duplicate name in the WINS database, and sends a positive response to the registering client. The response includes a Time-To-Live (TTL) value, which sets the time the server the name registration will be active in the database. The client must renew the registration before the TTL expires.
Negative	The WINS server finds an existing registration for the requested name in the database. The server sends a wait for acknowledgment (WACK) packet to the client and then sends a challenge, referred to the registered owner of the name. Having received a response from the registered owner, the server sends a

	negative name registration response to the WINS client attempting to register the name.

When a WINS server receives a name registration request for a name already in its database, the server sends a challenge, known as a name query request, to the owner of the registered name. The server waits 500 milliseconds between challenges, and if the client is multihomed, the WINS server tries each IP address it has for the computer until the WINS server receives a response or until it has tried all of the IP addresses.

Figure 7.3 illustrates the flow of messages between client, server, and challenged client. The first message is the name registration request; the last is the name response.

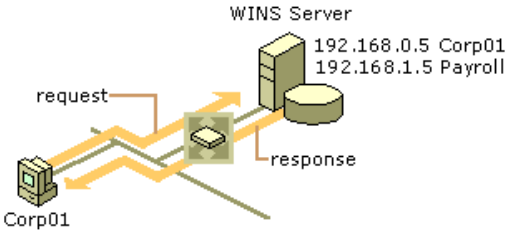


Figure 7.3 WINS Client Name Registration

In the first step in Figure 7.3, the Corp01 computer sends a message to its WINS server to register its address using a NetBIOS name registration message. The server replies with an acknowledgment of the address with a NetBIOS name registration response. Note that there is a second WINS server on the far side of the router; this server learns about the Corp01 address when the WINS server database replicates itself.

How WINS Clients Renew Their Names

WINS clients must renew their name registrations before the renewal interval expires. The renewal interval determines how long the server stores the name registration as an active record in the WINS database.

When a WINS client renews its name registration, it sends a name refresh request to the WINS server. The name refresh request includes the IP address and the NetBIOS name that the client seeks to refresh. The WINS server responds to the name refresh request with a name refresh response that includes a new renewal interval for the name.

When a WINS client refreshes its name, it performs the following steps:

1. When a client has consumed ½ of its renewal interval, it sends a name refresh request to the primary WINS server.
2. If its name is not refreshed by the primary WINS server, the WINS client tries to refresh again in 10 minutes and continues to try the primary WINS server repeatedly every 10 minutes for a total of 1 hour.
3. The WINS client, after trying to refresh its name registration with the primary WINS server for one hour, stops trying and attempts to refresh its name with the secondary WINS server.
4. If it is not refreshed by the secondary WINS server, the WINS client tries to refresh its name again using the secondary WINS server in 10 minutes and continues to try every 10 minutes for a total of 1 hour.
5. The WINS client after trying to refresh on the secondary WINS server for one hour, stops trying and tries to refresh using the primary WINS server.
6. This process of trying the primary WINS server and then the secondary WINS server continues until the renewal interval is consumed or the WINS client has its name refreshed.
7. If the WINS client succeeds in refreshing its name, the renewal interval is reset on the WINS server.
8. If the WINS client fails to register during the renewal interval on either the primary or secondary WINS server the name is released.

How WINS Clients Release Their Names

NetBIOS names can be released either explicitly or silently. They are explicitly released when a client shuts down gracefully. A silent release occurs when an client fails or is powered off. The silent release is noted at the WINS server when a name is not refreshed within the renewal interval.

When a name is released, the database entry is marked as released and given a time stamp with the current time plus the *extinction interval*. The extinction interval is the interval between when an entry is marked as released and when it is marked as extinct. The extinction timeout specifies the interval between when an entry is marked extinct and when the entry is finally scavenged from the database. This information is not propagated to partner WINS servers. If the release is explicit, the WINS server makes itself the owner of the record if it is not already.

In Windows 2000, the release of a WINS database entry is handled differently if the owner ID of the entry is different from the owner ID of the server that registered the name. If this is the case, the entry is marked as extinct and given a time stamp that is the current time plus the sum of the extinction interval and the extinction timeout. This is done to avoid windows of inconsistency between secondary and primary WINS servers. Because a released record is not replicated again, having already been replicated once, its name remains released on one WINS server and active on another for undesirably long periods.

Changing the released record to the extinct state results in its replication and enables rapid synchronization of WINS databases. Without extinction, inconsistencies might linger. For example, if the primary WINS server of a client is unavailable when the client shuts down, the name release would be directed to the secondary WINS server. If the primary WINS server is available again when the client restarts, the client would register and continue to refresh with the primary WINS server, which has not recorded any change in the status of the client, while the secondary WINS server would still reflect the released state of the client record.

How WINS Clients Resolve Names

WINS clients perform NetBIOS name–to–IP address mapping resolution by using the NetBIOS over TCP/IP (NetBT) component. A Windows NT–based computer is automatically configured to use one of four different NetBT name resolution modes (that is, methods for resolving names), based on how TCP/IP is configured on the computer. Table 7.1 describes the NetBIOS modes and how they resolve IP addresses from NetBIOS names.

To display a computer's TCP/IP configuration, including its node type, type **ipconfig /all** at the command prompt. For example, on a computer that is configured as a WINS client, the node type "Hybrid" appears when you type **ipconfig /all**.

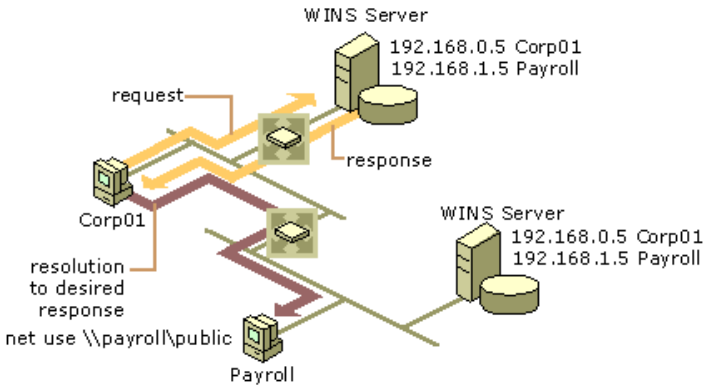
The name resolution process between an H-node WINS client and a WINS server follows this sequence:

1. When a user types a network-related command at the command prompt, such as **net use**, the client computer checks its NetBIOS name cache for the NetBIOS name and IP address of the destination host. If the client finds a mapping, the name is resolved without generating network activity.

2. If the client computer does not find the name in the NetBIOS name cache, the client makes three attempts to contact the first WINS server (if one is configured). If the first WINS server does not respond, the client attempts to contact the next WINS server until it has attempted to contact all configured WINS servers. If the name is resolved, the IP address is returned to the client.
3. If the name is not resolved by any WINS server, the client generates three B-node broadcasts on the local network. If the NetBIOS name is found on the local network, the name is resolved to an IP address.
4. If the NetBIOS name cannot be resolved using B-node broadcasts and LMHOSTS lookup is enabled, the client parses the local LMHOSTS file. If the NetBIOS name is in the LMHOSTS file, the name is resolved to an IP address.
5. If the NetBIOS name is not resolved from the LMHOSTS file, the client computer attempts to resolve the name through other host name resolution techniques. If the **Enable DNS for Windows Resolution** box is checked in the **WINS Address** property page of the **Internet Protocol (TCP/IP)** dialog box, it attempts to resolve the name using a local Hosts file or a DNS server.

   To resolve a host name, WINS checks the local Hosts file for a match against the local host name first. If the host name is found in the Hosts file, it is resolved to an IP address. The Hosts file must reside on the local computer.
6. If the name is not resolved from the Hosts file, the client sends a request to its configured DNS server. If the host name is found by a DNS server, it is resolved to an IP address.

   If none of these methods resolve the NetBIOS name, the **net use** command returns an error, indicating that the computer could not be found.



If your browser does not support inline frames, click here to view on a separate page.

Figure 7.4 Name Resolution via WINS

In Figure 7.4, the initial message is the name request from the client to its primary WINS server. This is followed by the response from the WINS server, returning the desired IP address. Once the response is received, the client uses this address to establish a connection to the desired resource.

Client Conflicts Detected During Registration

When a client node registers or refreshes a name, the name might already exist in the WINS database. The action taken by the WINS server depends on the state of the registered name. It might be active, released, or extinct. (An extinct name is referred to as a tombstone.) The name might be a unique name or a group name, owned by the server or a replica, a database entry copied from another WINS server, with a statically or dynamically assigned IP address. The IP address might be the same as or different from that specified in the registration request of the client.

Two cases are always handled the same way: Normal group entries and static entries are never overwritten. The WINS server always returns a negative name registration response to registration requests for a name that is already in the database as a group or static name. Internet groups get additional members through the use of Internet group registration. Internet groups, or "special groups" as they are sometimes called, are used for special, user-defined administrative groups. These internet groups are sometimes used to group resources such as, file servers and printers. In this case, if the record in conflict has been released or tombstoned, the name registration request is treated as the registration of a new name.

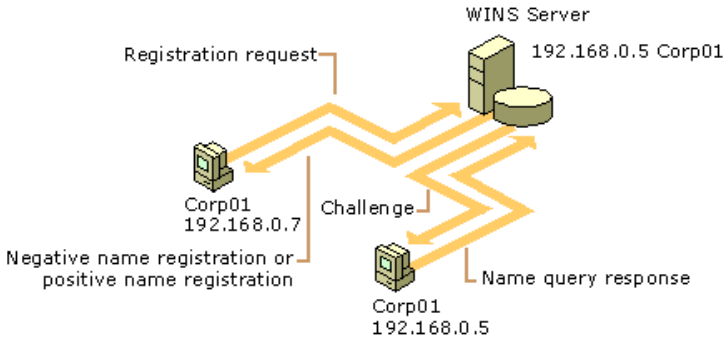
With unique, dynamic names, if the IP addresses are the same, the WINS server returns a positive name registration response and acts as described in Table 7.3.

Table 7.3 Name Registration Responses

State of Name	Server Action
Owned and active	Update time stamp
Replica and active	Update time stamp, take ownership, increment version ID
Released	Update time stamp, make active, increment version ID
Owned tombstone	Update time stamp, make active, increment version ID
Replica tombstone	Update time stamp, take ownership, make active, increment version ID

If the IP address of the registration request is different from the IP address of the database record, and the existing database record is already released or tombstoned, then the name registration is treated as new. The server sends a positive name registration response and updates the entry to reflect the new time, ownership, version ID, and active state.

If the existing database entry is active and has an IP address that is different from the IP address of the registration request, the WINS server must determine whether the name and IP address in the database entry are still in use. The WINS server does this by sending a name query request to the client computer with the IP address in question. Figure 7.5 shows the initial step of a registration request sent from the client to the server, followed by a challenge sent by the server to the old IP address.



If your browser does not support inline frames, click here to view on a separate page.

Figure 7.5 A WINS Server Challenges an Old Address

If the server receives a positive name query response from the old IP address, it rejects the new registration by sending a negative name registration response to the client that originally requested the name registration. If the old address does not respond to the name query request, the server assumes there is no computer with that name and IP address and accepts the new name registration. In this case, the last arrow indicates a positive name registration response.

WINS Client Behavior

This section looks at how WINS clients react to various basic scenarios, including:

- Daily startup of the WINS client.
- Plugging into a different subnet.
- Prolonged shutdowns.
- Joining two WINS systems.

Daily Startup

An active WINS client name registration in a WINS server database is replicated to all of the *push partners* of that server. A push partner is a WINS server that sends data to other servers to start replication. After some time, the active name registration is replicated to all WINS servers on the network.

When the WINS client is turned off at the end of the day, it releases the name. With the default extinction interval, it does not enter the extinct state during the night, and therefore it is not replicated again that night. When the computer is started the next morning, the WINS client registers the name again with the WINS server and receives a new version ID. This new, active name registration entry is replicated to the pull partners of the WINS server as on the previous day. A pull partner is a WINS server that requests data to be sent to it from other servers.

The number of name registration entries replicated each day is roughly equivalent to the number of computers started each day multiplied by the number of NetBIOS names registered by each computer.

On large networks (50,000 or more computers), the biggest traffic load may be the name registration requests generated when WINS clients start on the network. Fortunately, the difference in time zones in large enterprise networks provides some distribution of this WINS client startup load.

Plugging Into a Different Subnet

A roaming user who powers down a computer and then moves it to a different subnet with another primary WINS server generates name challenge traffic. Typically the name registration request is answered with the wait for acknowledgment message. Then, assuming the active entry was replicated, the new WINS server generates a name query packet to challenge the IP address currently in its database for the name. Because the computer that registered that address is no longer active on that subnet and no longer using that IP address, it makes no reply. Just to be sure that the lack of response is not a fluke, the WINS server repeats the query three times.

Usually the name challenge never travels over the subnet that the computer has left because the ARP request fails. However, the challenge message does travel on the subnet of the new WINS server and the links between the routers. The WINS server assigns a new version ID to the new entry so that it will replicate from its new owner to other WINS servers.

Prolonged Shutdowns

Some computers do not start up for a period longer than the *verification interval*. The verification interval is the interval after which the WINS server must verify that any old names that it does not own that are in its database are still active. The WINS server does not delete replica entries for these computers because the owned entries never become extinct, remaining active in the database. A computer that occasionally starts up refreshes all of its replica entries by giving each of them a new version ID. This new version ID prompts replication of the replicas to other WINS databases on the network.

Sometimes the computer stays shut down for an extended period, and therefore its replicas are not refreshed for a period longer than the verification interval. When a WINS server has such an old replica, it tries to verify this entry with the owning WINS server. If the owning server does not find this entry, it is removed from the database of the verifying server. If the entry is verified, its new state is recorded.

Joining Two WINS Systems

When two organizations merge, their computer systems must merge as well, including their WINS systems. When merging two WINS systems, the initial replication load and the potential for conflicting NetBIOS names might present problems. After a WINS server from one system is connected to a WINS server from the other, they eventually replicate their records with each other; because they have no shared records, the whole database from each system must be replicated. Then their replication partners copy the new entries, until all the databases have converged.

To avoid making this process any more difficult, merge the systems and force replication at a time when the connecting WAN links are more or less idle. When the databases contain conflicting names, the conflicts are resolved, which may result in other traffic. This process is described in "Client Conflicts Detected During Registration" earlier in this chapter.

Note The users of computers with conflicting names will probably call the help desk when they get the "duplicate name" messages and their computers refuse to open new sessions.

Best Practices for WINS Clients

To configure and manage clients properly requires some attention. The best options for WINS clients are outlined here.

Configure Clients with a Full List of WINS Servers

In previous versions of Windows NT, clients were only able to use a primary and secondary WINS server. For Windows 2000, WINS clients can be configured with up to 12 WINS servers. These servers can be configured either statically at the **Internet Protocol (TCP/IP)** properties dialog box or dynamically through DHCP (using option 44). By configuring additional WINS servers, clients gain additional fault tolerance.

Use Nbtstat –RR to Manage Client Connectivity

The Nbtstat command-line tool—new in Windows 2000—allows you to purge the local NetBIOS names cache of remote names and force immediate renewal and re-registration of the local names of the client. This is useful as a first recourse for troubleshooting WINS client connectivity problems; in particular, you can use this tool to repopulate the client entries and replicate them to the partner WINS server without rebooting the clients.

Client Configuration Practices

WINS is a client/server system requiring software on both the client and the server in order for the NetBIOS computer name–to–IP address resolution to occur. Getting the client configuration right can head off many problems.

Windows NT clients that participate in the WINS process register NetBIOS names. These names are configured through the **System** utility in Control Panel and can be altered at any time. Problems might arise if a user changes his or her computer's NetBIOS name to the same name as that of a Windows 2000 computer, or to the name of an existing Windows NT domain. This client impersonates the server and essentially is registered with the WINS service as a Windows 2000 computer. This problem only happens when the server or domain controller is not available to defend the name in a WINS challenge. To avoid this problem:

- The first, but least desirable, method of dealing with impersonation in Windows 2000 is to place static entries in the WINS database, ensuring that no user can configure his or her computer to dynamically impersonate a server. This method, as with any static process, is more administratively intensive than the use of dynamic registration of the computer NetBIOS name.
- The second method is to set the client computers' configurations so that users cannot alter the NetBIOS names of their computers. This method allows for minimal administrative overhead for WINS NetBIOS name registrations and provides a controlled client environment. You can control Windows 95 and Windows NT Workstation clients through system policies that determine what access a user can or cannot have to altering features on their own computers.

All clients should be upgraded to the newest client platform to control access to desktop configuration parameters. Use system policies in the Computer Management console to prevent users from changing their computers' NetBIOS names. To access the Computer Management console, right-click **My Computer** and choose **Manage** on the drop-down menu.

Microsoft WINS Servers

The WINS system for name resolution can be viewed as a set of tightly integrated components:

- **WINS server.** A computer that provides the WINS service and replicates the WINS database with other WINS servers so that complete name-to-address resolution information is available regardless of which WINS server a WINS client uses.
- **WINS client.** Any WINS-enabled computer that uses the WINS service to register or refresh its NetBIOS name and IP address.
- **WINS proxy.** A WINS-enabled computer that helps resolve name queries in routed TCP/IP intranets for computers that are not WINS-enabled.
- **WINS database.** Dynamically updated list of NetBIOS names and their associated IP addresses, including IP addresses assigned by DHCP. In networks with multiple WINS servers, the servers exchange database updates through replication.
- **WINS Management Console.** A plug-in to the Microsoft Management Console that provides a range of management tools.

The next sections take a closer look at each of these components, beginning with the WINS servers and proxies that form the backbone of the entire system.

Overview of WINS Servers

WINS servers prevent the administrative difficulties inherent in the use of both NetBIOS name query broadcasts and static mapping files such as LMHOSTS files. Microsoft WINS eliminates the need for NetBIOS name query broadcasts, saving valuable network bandwidth while maintaining a dynamic database of NetBIOS name–to–IP address mappings.

The databases replicated between WINS servers contain NetBIOS names and their associated IP addresses. When Windows-based computers log on to the network, their NetBIOS names and IP addresses are registered and added to the WINS server database, providing support for dynamic updates. The WINS server database is replicated among multiple WINS servers in a LAN or WAN. This database replication prevents users from registering duplicate NetBIOS names for different computers on the network.

A Microsoft WINS server solves the problems inherent in resolving names through IP broadcasts and frees network administrators from the demands of updating static mapping files. WINS automatically updates the WINS database when dynamic addressing through DHCP assigns new IP addresses—for instance, when computers move between subnets.

WINS servers also provide the following benefits:

- Dynamic database that supports NetBIOS name registration and resolution in an environment where DHCP-enabled clients are configured for dynamic TCP/IP address allocation.
- Centralized management of the NetBIOS name database and replication to other WINS servers.
- Reduction of NetBIOS name query broadcast traffic.
- Support for client computers running Windows NT Server, Windows NT Workstation, Windows 95, Windows 98, Windows for Workgroups, and LAN Manager 2.*x*.
- Support for transparent browsing across routers for client computers running Windows NT Server, Windows NT Workstation, Windows 95, Windows 98, and Windows for Workgroups.

Microsoft WINS servers communicate with other Microsoft WINS servers to fully replicate their databases with each other. This ensures that a name registered with one WINS server is replicated to all other Microsoft WINS servers within the intranet, providing a consistent enterprise-wide database. When a network uses multiple WINS servers, every WINS server is configured as a pull partner or a push partner of at least one other WINS server.

A pull partner is a WINS server that requests new WINS database entries, called replicas, from its partner. The pull occurs at set intervals, defined by the replication interval, or in response to an update notification from a push partner.

A push partner is a WINS server that sends update notification messages after the database receives the number of updates that exceed the update count threshold or sends them immediately if the server is configured to send updates when an address changes (by selecting the **On Address Change** check box in the **Replication Partners** window of the WINS console). If you have configured the WINS server this way, it propagates the triggers received from a partner to all other partners when its WINS database changes. The

partners of the WINS server then pull these changed entries from the WINS server with the updated database.

Registration of Group Names

In addition to registration of unique names, mentioned in the preceding description of NetBIOS, WINS allows registration of group names. WINS recognizes two types of groups: normal groups and special groups.

Normal Group Names

A normal group name has several key differences compared to a normal unique name. Most important, it does not actually have an address associated with it. It is assumed to be valid on any subnet. The same group can be registered at more than one WINS server. The whole group has a single time associated with it that indicates the last time a node on any subnet registered or refreshed the name. When it receives a name query for the group, WINS returns the limited broadcast address (255.255.255.255). The WINS client then issues a broadcast to its subnet to resolve the name.

As the group names replicate from one WINS server to another, the name is added to the database of servers that do not already have it. However, since there is no address to propagate with the name, the entry for the group is just a name, without an associated address. When a group name is not refreshed, it is released and eventually becomes a tombstone.

The released and tombstoned states, however, have a slightly different meaning for group names than for unique entries. The WINS server answers name queries for released and tombstone groups. Unique name registrations that clash generate a negative response. For group entries, you can think of released and tombstoned states as pseudo states (pseudo-released and pseudo-tombstoned). These two states change at the end of the extinction interval, a configured value that establishes how long entries linger in the released and tombstoned states. After that interval, the version ID is incremented; this change means that the state information is then replicated to other WINS servers.

Special Group Names

When a name registration is received for a special group, WINS stores the actual address, rather than the limited broadcast address. A time stamp, reflecting the last registration or refresh received for that entry, and an owner ID are stored with each address entry in the group. When the WINS server receives a name query for such a group, it returns the IP addresses that have not expired. These groups, like normal groups, are replicated from the WINS server where they first registered to replication partners of that server.

Static NetBIOS name mappings can be any of the types listed in Table 7.4.

Table 7.4 Static NetBIOS Name Mappings

Type Option	Description
Unique	A unique name that maps to a single IP address.
Group	Also referred to as a "Normal" Group. When adding an entry to Group by using the WINS snap-in, you must enter the computer name and IP address. The IP addresses of individual members of Group are not stored in the WINS database. Because the member addresses are not stored, there is no limit to the number of members that can be added to a Group. Broadcast name packets are used to communicate with Group members.
Domain	A NetBIOS name–to–IP address mapping that has 0x1C as the 16th byte. A domain group stores up to 25 addresses for members. For registrations after the 25th address, WINS overwrites a replica address, or if none is present, it overwrites the oldest registration. Domain names are used to add a static entry for the computer, specified by name in a static mapping to a list of domain controllers used on the network.
Internet group	Internet groups are user-defined groups that allow you to access group resources, such as printers, for easy reference and browsing. The default 16th byte of an Internet group name is set to 0x20. An Internet group can store a maximum of 25 addresses for members. When you add an Internet group three unique records are added: InternetGroupName<0x20> is used for file registration. InternetGroupName<0x0> is used for workgroup registration, and InternetGroupName<0x3> is used by messenger service. (The messenger service is used for pop-up messages on the screen. For example the printer messages that tell you that printing is complete.) This is similar to the domain group. Internet group members can be added via dynamic group registrations. A dynamic member, however, does not replace a static member added by using the WINS management console or importing the LMHOSTS file.
Multihomed	A unique name that can have more than one address, used for multihomed computers. No more than 25 addresses can be registered as multihomed. For registrations after the 25th address, WINS overwrites a replica address, or if none is present, it overwrites the oldest registration.

Secondary WINS Servers

Client computers should be configured with both a primary and secondary WINS server. If the primary WINS server cannot be reached for a WINS function (such as registration, refresh, release, query), the client requests that function from its secondary WINS server. The client periodically retries its primary WINS server.

Note While Windows 2000 Advanced Server supports the use of clustering for WINS servers, in almost all cases this service is unnecessary. Configuring secondary WINS servers provides the same function. In addition, maintaining secondary WINS servers is easier, and secondary WINS servers can be at a different location For more information about clustering WINS servers, see "Burst Handling" later in this chapter and "Windows Clustering" in the Microsoft® *Windows® 2000 Server Resource Kit Distributed Systems Guide*.

In networks with both a primary and secondary WINS server, it is best to configure half the clients with one server as the primary and the other server as secondary, and configure the other half of the clients with the opposite selections for primary and secondary servers. This cuts the burden on each server in half, while ensuring that a secondary server does not sit idle until the primary server fails.

Microsoft WINS Proxy

RFC 1001 recommends against using the B-node name resolution in a routed network—that is, relying on broadcasts for name queries. However, in practice, B-nodes are sometimes useful in routed networks, and sometimes B-nodes cannot be removed or updated. For this reason, Microsoft introduced WINS proxies. A WINS proxy is a WINS-enabled computer that helps resolve name queries for computers that are not WINS-enabled in routed TCP/IP networks.

By default, computers that are not WINS-enabled use B-node name resolution. The WINS proxy listens on the local subnet for B-node name service broadcasts (such as registration, refresh, release, query) and responds for those names that are not on the local network. A WINS proxy communicates with the WINS server with directed datagrams to retrieve the information necessary to respond to these broadcasts.

The WINS proxy resolves names for non-WINS clients in this way:

1. When a non-WINS client sends a name query broadcast, the WINS proxy accepts the broadcast and checks its cache for an IP address associated with the NetBIOS name.
2. If the WINS proxy has the IP address in its cache, the WINS proxy sends this information to the non-WINS computer as a NetBIOS name response.
3. If the IP address is not in cache, the WINS proxy queries a WINS server for the IP address associated with the requested name.
4. If a WINS server is not available on the local subnet, the WINS proxy can query a WINS server across a router, caching the NetBIOS names and IP addresses for subsequent queries.

The role of the WINS proxy is similar to that of the DHCP and BOOTP relay agents, which forward DHCP client requests across routers. Because the WINS server does not respond to broadcasts, a computer configured as a WINS proxy should always be installed on subnets containing computers that are not WINS-enabled.

The WINS proxy checks broadcast name registrations against the WINS database by sending name query requests to ensure that the names do not conflict with other names in the database. If a name exists in the WINS database, by default the WINS proxy might send a negative name registration response to the computer trying to register the name. In response to a name release request, the WINS proxy simply deletes the name from its cache of remote names.

When the WINS proxy server receives a name query, it checks its remote name table. The WINS proxy always differentiates name queries for names on the local subnet from remote names elsewhere in the network. It compares the address of names it resolves to its own address using the subnet mask, and if the two match, the WINS proxy does not respond to the name query.

If the WINS proxy does not find the name in the remote name table, it queries the WINS server, and then enters the name into the remote name table in a "resolving" state. If the WINS proxy receives a query for the same name before the WINS server has responded, the WINS proxy does not query the WINS server again. When the WINS proxy receives the response from the WINS server, the WINS proxy updates the remote table entry with the correct address and changes the state to "resolved." The WINS proxy only sends a reply message to the client if the WINS proxy has the response already in its cache.

The behavior of a B-node client does not change when a WINS proxy is added to the local subnet. If the first name resolution query times out, the client tries again. If the WINS proxy has the answer cached by the time it intercepts the new query, the WINS proxy answers the client.

Note Only one computer should be configured as a WINS proxy on each subnet. Because each WINS proxy on a network relays every broadcast it hears, configuring more than one WINS proxy per subnet can overload the WINS servers.

When the WINS proxy receives the next name query for that name, it again sends a response to the client. NetBIOS contains no provision for a name server to "deliver" a name resolution to a client; a name is always resolved in response to a query. Therefore, computers using the WINS proxy for B-node name resolution must be configured to retry the name query. To reduce duplicate traffic, only one WINS proxy should be active on any given subnet.

The name–to–IP address mappings that the WINS proxy receives from the WINS proxy are stored in the WINS proxy server cache for a limited time. By default, this value is 10 minutes; the minimum value is 1 minute.

To configure a computer as a WINS proxy server, you must edit the registry of that computer. The value of the **EnableProxy** registry entry must be set to 1 (REG_DWORD). This entry is located in the following registry subkey:

HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \Netbt \Parameters

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Microsoft Management Console (MMC) or Control Panel whenever possible.

Querying with a WINS Proxy

In Figure 7.6, a small broadcast-based LAN consisting of two clients (A and B) is connected to a larger network through a router. A NetBIOS application on client B wants to communicate with client C. Normally, this would not be possible because client C is on the other side of the router from client B. However, by configuring a computer running Windows 2000 Professional to act as a WINS proxy on the LAN, clients B and C can communicate.

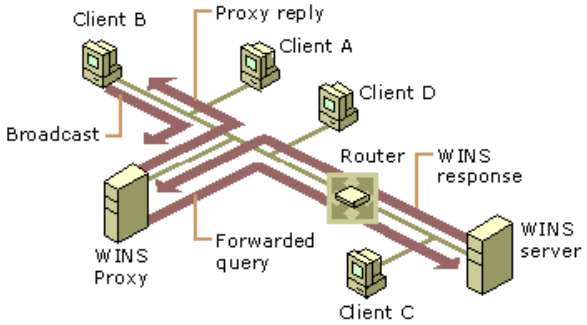


Figure 7.6 Operation of a WINS Proxy Server

Client B broadcasts a name query request to obtain the IP address of client C. Client C does not receive the request because the router does not pass along the broadcast. The WINS proxy sees a name query request broadcast for a node on a different subnet and sends a name query request, a directed datagram, to the WINS server. The WINS server returns a positive name query response containing the IP address for client C to the WINS proxy, where it is cached for future queries. The WINS proxy also passes this information to client B.

Burst Handling

With the addition of burst handling features in Windows 2000, WINS servers can now support high-volume—or "burst"—server loads. In these situations, many WINS clients actively seek to register their names with their local WINS server at the same time. In burst mode, the WINS server responds positively to clients that submit registration requests before the WINS server has processed and physically entered these updates in the WINS server database.

Burst mode uses a burst queue size as a threshold value to determine how many requests to process normally before enabling burst mode handling. By default, the burst queue allows 500 requests before a WINS server engages burst handling. For more information about changing the burst queue size, see "Configuring Burst Mode Support" in this chapter.

The burst queue allows WINS to handle intermittent periods of heavy registration and refresh traffic, such as when the WINS server is either started with a clean database or when many WINS clients come online for the first time. Either situation creates a large number of requests for registration and refreshment of names.

The function of burst handling is to answer requests superficially (with a positive response), therefore decreasing the load on the network. Burst handling also extends and varies the delay interval to distribute the load over time.

Burst handling is enabled for any WINS server running Windows NT Server 4.0 with the current service pack, as well as with Windows 2000 Server. A WINS server that supports burst handling initiates burst handling once the number of WINS client registration requests exceeds the burst queue size.

How Burst Handling Works

With burst handling, additional client requests beyond the burst queue size are immediately answered with a positive response from the WINS server. The response also includes a varying delay interval or a Time to Live (TTL) interval to help regulate the client registration load and handling of requests received in a single burst period.

Including a delay interval in the success responses lowers the rate at which new WINS clients attempt to refresh and retry name registration, and it regulates the burst of WINS client traffic.

For each additional round of 100 client requests, the delay interval is incremented by the WINS server by an additional 5 minutes until the delay interval reaches a maximum of 50 total minutes. If WINS client traffic is still arriving at burst levels once the delay interval reaches its maximum, the WINS server answers the next round of 100 client requests with the initial delay interval value of 5 minutes, and the incrementing process begins again.

For example, if the default burst queue size of 500 entries is used, the WINS server replies normally to the first 500 requests. It replies immediately to the next 100 WINS client registration requests by sending early success responses. Those early success responses use a starting delay interval value of 5 minutes.

The WINS server continues to handle burst-level request traffic in this manner until the server reaches its maximum intake level of 25,000 name registration and refresh queries. At this point, the WINS server begins dropping queries.

Configuring Burst Mode Support

You can change the level of burst mode support or disable it by using the **WINS Services Properties** dialog box from the WINS server. To reach this interface, open Control Panel, double-click **Administrative Tools**, then choose **Computer Management** and open the **Services and Applications** section. In this section, click **WINS**, and then, on the **Action** menu, click **Properties**. To further configure or disable burst mode support where desired, click **Advanced**.

Four buttons are available for configuring burst mode: **Low**, **Medium**, **High**, and **Custom**. **Custom** allows you to enter a number of queries from 50 to 5,000. **Low**, **Medium**, and **High** configure burst queue sizes of 300, 500, and 1,000, respectively.

By default, burst handling is enabled, and the burst queue is sized to Medium.

To modify burst handling

1. In the console tree in MMC, click the name of the WINS server for which you want to modify burst handling properties.
2. On the **Action** menu, click **Properties**.
3. Click the **Advanced** tab.
4. In **Enable burst handling**, modify default settings as needed.
5. To view a description of a dialog box item, right-click the item, and then click **What's This?**

Clustering

Windows 2000 supports clustering of WINS servers. However, before simply adding WINS service to a set of clustered servers, be sure to consider both the advantages and disadvantages of doing so. In many cases, where the overall number of WINS servers is small, clustering WINS is simply not necessary—replication makes WINS fault tolerant. Instead, configure your WINS clients with the address of a secondary WINS server to ensure uninterrupted service.

To add WINS to a cluster

1. Be sure the WINS service is installed and started on both servers.
2. Right-click in the resources dialog box and, on the menu that appears, click **New Resource**.
3. Click **Next**, and choose the group you want to add.
4. Choose the possible owners—that is, the other members of the cluster.
5. Set the dependencies for the resource: the disk, the IP address, and the network name.
6. Type the path to the backup database, and click **Finish**.

Be sure that the owner to which you add WINS service has a disk, an IP address, and a name resource. Also, the database path must end with a backslash (****)and specify a location on the dependent disk that you select. For example, if the dependent disk is drive G, you must choose a database path on drive G.

To test that the clustering is working correctly, bring the dependent disk online (it begins offline), then right-click on the window and move the group to the other node. Groups will show the drive as a resource that can be moved; the drive is moved from one node to the next with the group to which it belongs. You should see the entry in the Owner category change as the resource moves.

For more information about clustering WINS servers, see "Windows Clustering" in the *Windows 2000 Resource Kit Distributed Systems Guide*.

Note If you choose to cluster your WINS servers, be sure to equip those servers with a hard disk with high-speed I/O that is dedicated to WINS service. This helps speed up the database response, and ensures that clustering efficiency is high.

Best Practices for WINS Servers

Keeping WINS servers up and functioning prevents WINS clients from reverting to B-node name resolution and flooding a network with broadcast requests. Here are a few suggestions for keeping servers operating efficiently.

Use the Default Configuration

The default settings of WINS, set when the service is first installed, provide the optimal configuration for most conditions and should be used in most WINS network installations. If you modify the default settings, be sure that the need to do so is clear and necessary, and that you understand all the implications.

Minimize the Number of WINS Servers

Using too many WINS servers can complicate network problems, so be conservative when adding WINS servers to your network. Use the minimum number of WINS servers to support all your clients while maintaining acceptable performance.

When planning your servers, remember that each WINS server can simultaneously handle hundreds of registrations and queries per second. In part, this is because the data exchanged between WINS servers and WINS clients is typically small. The average WINS record is about 40 bytes.

WINS network traffic during client registration can be much less than that of DHCP, which uses client broadcasts to discover servers. By default, most WINS clients first send directed point-to-point datagrams to the primary WINS servers.

In general, avoid deploying large numbers of WINS servers unless they are strictly necessary. Limiting the number of WINS servers minimizes WAN traffic related to WINS replication, provides good NetBIOS name resolution, and reduces administrative problems without sacrificing functionality. To design a WINS installation that includes more than 20 WINS servers, seek assistance from Microsoft Product Support Services.

Requests to WINS servers are directed datagrams, meaning that WINS requests are routed. Therefore, one WINS server is adequate for a network of 10,000 nodes, although to provide fault tolerance, at least two WINS servers are recommended. Because the data exchange between WINS servers and clients is typically about 40 bytes in size, and WINS communicates using directed datagrams, a single WINS server may be enough for very small networks.

Based on the number of CPUs in the computer, WINS determines how many threads to create to handle client queries; it creates one thread per CPU. Each name registration takes about 40 milliseconds with logging enabled. If logging is disabled, registrations are much faster, but this configuration introduces the risk of losing the last few updates to the WINS database when a failure occurs.

Use High-Performance Disk Hardware

WINS causes frequent and intense activity on server hard disks. To provide the best performance, consider RAID-based solutions that improve disk access time when you purchase hardware for a WINS server. You should include WINS when evaluating the performance of a server. By monitoring system hardware performance in the most demanding areas of utilization (CPU, memory, and disk I/O), you make the best assessments of when a WINS server is overloaded and should be upgraded.

Add Network Interface Hardware Carefully

Be careful when adding more network adapters to a computer running Windows 2000. You can increase the reliability of mission-critical systems while adding the hardware simply by reducing the number of services running on the computer. Many of the services running on a mission-critical computer (such as a primary domain controller) can be offloaded to other computers and then returned to the upgraded original computer once the change is complete.

Configure Each Server to Point to Itself

Each WINS server you install on your network must register in WINS its own set of unique and group NetBIOS names. WINS service problems can occur when registration and ownership of a WINS record become split—that is, when names registered for a particular WINS server are owned by different WINS servers. To prevent these problems, configure each WINS server as its own primary and secondary WINS servers.

WINS Server Fault Tolerance

To prevent a WINS failure from affecting server communications, you may want to consider using an LMHOSTS file to provide secondary name resolution in the event of a WINS failure. While these files are not a recommended solution, in rare circumstances they may provide an effective stopgap measure. LMHOSTS files must be tightly managed because changes in the NetBIOS environment will not automatically update static name files.

To use LMHOSTS for name resolution, you must make a correctly configured LMHOSTS file available for locating Windows 2000 computers when WINS servers fail. A master LMHOSTS file should contain static IP address mappings for Windows 2000 computers. This file should be distributed to each Windows domain using one of the following three options:

- The typical Windows 2000 LMHOSTS file contains a universal naming convention (UNC) path to a central file. By pointing to a single file, you need only maintain one copy of the LMHOSTS file.
- For computers without Windows 2000, you can schedule a job using the scheduler service to distribute the master LMHOSTS file to the required servers automatically. The **winat** command from the *Windows 2000 Resource Kit* may make this task easier. Send the file to the primary domain controller (PDC) and one backup domain controller (BDC) on each domain.
- The least efficient option is to manually copy the file to each server and client that needs it, or to update each LMHOSTS file locally. This may be still be worthwhile for a network with a single WINS server.

Once the LMHOSTS file is prepared and distributed, if a server fails, the local LMHOSTS file of each server references the master LMHOSTS file on the PDC sharepoint. If you use an #INCLUDE statement, the central LMHOSTS file is also available from alternate servers.

Do Not Use Extended Characters

Do not use extended characters in NetBIOS names, especially the underscore (_) and the period (**.**). The underscore character is converted to a dash in DNS host names. For example, NTServer_1 becomes NTServer-1, leading to failure of name resolution of a name that may, in fact, be recorded in the DNS files.

Align the Lease and Refresh Periods for DHCP and WINS

When you configure a network to use both DHCP and WINS, set the DHCP lease period to be roughly equal to or greater than the WINS renewal period. This prevents a situation in which the WINS server fails to notice that a DHCP client releases a DHCP-assigned IP address; the client cannot send a WINS renewal request if the client fails to renew its IP address. If another computer is assigned that IP address before the WINS server notes the change, the WINS server mistakenly directs requests for the address to the new client.

WINS Database

When a client needs to contact another host on the network, it first contacts the WINS server to resolve the query using mapping information from the database of the server. The relational database engine of the WINS server accesses an indexed sequential access method (ISAM) database. The ISAM database is a replicated database that contains NetBIOS computer names and IP address mappings.

For a WINS client to log on to the network, it must register its computer name and IP address with the WINS server. This creates a single mapped entry in the WINS database for the client. Because these entries are updated each time a WINS-enabled client logs on to the network, information stored in the WINS server database remains accurate.

Managing the WINS Server Database

The Windows 2000 WINS database uses the performance-enhanced Extensible Storage Engine, an updated version of the generic storage engine that serves both Microsoft Exchange 5.5 servers and Windows 2000 servers. This database imposes no limit to the

number of records that a WINS server can replicate or store. The size of the database depends on the number of WINS clients on the network, but it is not directly proportional to the number of active client entries. As inactive entries proliferate, the WINS database grows, and many WINS client entries become obsolete. Eventually, these entries clutter the database.

To recover the unused space, the WINS database is compacted. In Windows 2000, WINS server database compaction occurs as an automatic background process during idle time after a database update. Because the database compaction is also dynamic, you do not need to stop the WINS server to compact the database; this is also known as online compaction. However, while WINS performs regular online compaction, this reduces but does not eliminate the need for offline compaction. The WINS service will still need to be stopped periodically for offline compaction. For more information, see "Managing the WINS Server Database" later in this chapter.

The database files, stored in the directory %*SystemRoot%*\System32\Wins, are described in Table 7.5.

Table 7.5 WINS Server Database Files

File	Description
J50.log and J50*xxxxx*.log	A log of all transactions done with the database. This file is used by WINS to recover data if necessary.
J50.chk	A checkpoint file, used when the WINS database starts up to determine whether the last shutdown was clean and all databases are consistent. If not, the checkpoint file helps determine from what log file to begin recovery.
Wins.mdb	The WINS server database file, which contains two tables, an IP address–to–owner ID mapping table, and a name–to–IP address mapping table.
Winstmp.mdb	A temporary file created by the WINS service. The database uses it as a swap file during index maintenance operations. It might remain in the directory %S*ystemRoot%*\System32\Wins after a crash.

Caution The files J50.log, J50*xxxxx*.log, Wins.mdb, and Winstmp.mdb should not be removed or tampered with in any manner.

The WINS management console provides the tools you need to maintain, view, back up, and restore the WINS server database. For example, you use the WINS management console to back up the WINS server database files.

Backing Up the WINS Database

The WINS management console provides backup tools so that you can back up the WINS database. After you specify a backup directory for the database, WINS performs complete database backups every three hours, by installation default. For specific instructions on how to back up and restore the WINS database, see the Windows 2000 Server Help. You should also periodically back up the registry entries for the WINS server.

Repairing a WINS Database

If your WINS database becomes corrupted, you can use various options to renew its integrity. In cases in which the corruption is limited to a specific set of records, you can repair them by selectively increasing or decreasing the starting version number used by the WINS server that owns the affected records. If you choose this method, you can adjust the starting version used by the server to force replication of uncorrupted WINS records, which removes the affected records from other WINS servers.

If the corruption can't be repaired, you can delete the WINS database and entirely restore it from a backup (assuming that one exists). You can use the WINS backup feature in the WINS management console to make backup copies of the WINS database.

Sometimes you can repair WINS database corruption by increasing the highest version number of the local WINS server database; to do this, you must increase the value specified in the **Starting Version Count** box in the WINS server preferences. Then, the next time WINS restarts, the specified WINS server updates its local version number for any records it owns.

Increasing the value of the starting version number on the owning WINS server forces replication for all records owned by the specified WINS server to other remote partner WINS servers during the next replication cycle.

Note that you can set the value of the starting version number only to a value higher than the existing highest version number used by any locally owned records on the selected server. If there are no locally owned WINS records for the server, you can only set the starting version number to a higher number than the current starting version number count. Once a higher value is set, you cannot lower the value without first deleting the local WINS database and reinstalling WINS on the server computer.

Also, values entered and used for **Starting Version Count** are interpreted as hexadecimal numbers. WINS might adjust the value you specify to a higher value to ensure that database records are quickly replicated to other WINS servers. The maximum value that the WINS management console accepts as a valid starting number is a hexadecimal value of **FFFFFFFF**.

Using Replication to Restore Data

If the time to WINS convergence is low (that is, changes are replicated among the WINS servers quickly), the preferred method of restoring a local WINS server database is to use a replication partner to restore data after corruption. This method is most effective if the WINS data is mostly up to date on the replication partner.

The easiest way to restore a local server database is to replicate data back from a replication partner. Two registry entries control this feature: **InitTimeReplication** and **InitTimePause**. **InitTimeReplication** is an entry in the following subkey:

HKEY_LOCAL_MACHINE \System \CurrentControlSet \Services \Wins \Partners \Pull and Push

The value of this entry is 1 by default and causes WINS to replicate with the partner at initialization time. **InitTimePause** is an entry in the WINS Parameters subkey that tells WINS to pause while the replication takes place. These entries are discussed in this section.

Name: InitTimeReplication

Data Type: REG_DWORD

Description: If the value of **InitTimeReplication** is set to 1, the default value, the WINS server pulls replicas of new database entries from its partners when the system is initialized or when a replication-related parameter changes; if the value is 0, replication occurs only as often as specified by the value set for **Replication interval** in the Replication Partner Properties dialog box (shown in Figure 7.10).

Name: InitTimePause

Data Type: REG_DWORD

Description: The value set here determines whether WINS starts in a paused state and remains in that state until its first replication is complete. If the value of **InitTimePause** is 1, WINS starts in a paused state; if the value is 0, the default value, WINS does not start in a paused state. In the paused state, WINS does not accept any name registrations, releases, or queries. WINS remains in the paused state until it has replicated with its partners or until its first replication attempt has failed. Note that if the value of **InitTimePause** is

set to **1**, then **InitTimeReplication** (in the Pull partners subkey) should be set to **1** or be deleted from the registry.

Compacting the WINS Database

In Windows 2000 Server, the WINS service performs dynamic Jet compaction of the WINS database while the server is online. This reduces the need to use Jetpack.exe for offline compaction. Therefore, this procedure might not be as critical now as it was in the past for WINS and DHCP servers running earlier versions of Windows NT Server.

Windows 2000 Server includes the Jetpack.exe utility so that it can be used to compact the WINS and other Jet databases (such as DHCP) when those databases are offline. Microsoft recommends you use Jetpack.exe to compact a Jet database periodically whenever the database grows beyond 30 megabytes or more in size.

Use the Jetpack.exe command-line tool to perform offline compaction. The correct syntax for Jetpack.exe is:

jetpack <*database name*> <*name of the temporary database*>

Suppose that you have a temporary database with the file name Tmp.mdb, and the WINS database has the file name Wins.mdb. To compact the database in this example, you enter the following commands:

cd *%SystemRoot%***System32\Wins**

net stop wins

jetpack wins.mdb tmp.mdb

net start wins

Jetpack compacts the WINS database; first it copies database information to the temporary database file, Tmp.mdb, then it deletes the original WINS database file, Wins.mdb. Finally, it renames the temporary database file to the original file name, Wins.mdb.

Scavenging the Database

Like any database, the WINS server database becomes littered with junk entries over time and must periodically be cleaned and backed up. Scavenging the WINS server database takes care of this. It is usually performed at the same time as regular backups.

Scavenging updates the name state of WINS database entries, clearing the local WINS server database of released entries. It also clears away entries replicated from a remote WINS server that were not removed from the local WINS database when they were removed from the remote database. This scavenging process occurs automatically over intervals defined by the relationship between the renewal and extinction intervals defined in the **Configuration** dialog box. You can also find this dialog box on the **Name Record** tab of the **Server Properties** page, and the configuration page is shown in Figure 7.8.

Table 7.6 describes the effects of scavenging on WINS database entries.

Table 7.6 State of WINS Database Entries Before and After Scavenging

State before scavenging	State after scavenging
Owned active name for which the renewal interval has not expired	Unchanged
Owned active name for which the renewal interval has expired	Marked released
Owned release name for which the extinction interval has not expired	Unchanged
Owned released name for which the extinction interval has expired	Marked extinct
Owned extinct name for which the extinction timeout has not expired	Unchanged
Owned extinct name for which the extinction timeout has expired	Deleted
Replica of extinct name for which the extinction timeout has expired	Deleted
Replica of active name for which the verification interval has not expired	Unchanged
Replica of active name for which the verification interval has expired	Revalidated
Replica of extinct or deleted name	Deleted

Scavenging maintains the correct state information in the database by examining each record the WINS server owns, comparing its time stamp to the current time, then changing the state of those records whose state has expired (changing a record's state from active to released, for example).

Scavenging occurs on a preset schedule. The scavenging timer starts when the server starts up and is equal to half the renewal interval. Because of this, the WINS service should not be stopped or restarted before half the renewal interval has passed, or scavenging will not occur. Scavenging first occurs after half of the renewal interval has elapsed. During the first scavenging, all scavenging actions are performed except one: the deletion of the tombstones. Tombstones are not deleted until at least three days have elapsed since startup of the server, to allow sufficient time for their replication. Scavenging recurs at one-half the renewal interval (or can be initiated manually).

Scavenging follows the algorithm shown in Figure 7.7.

```
Get records owned by self
If Current Time > Time Stamp
Change State
Active -> Released
Released -> Tombstone
Tombstone -> Delete from database
Get replica Tombstones
If Current Time > Time Stamp
Delete record from database
Get Active replicas
If Current Time > Time Stamp
Verify with owner that record still exists
If exists
Time Stamp = Current Time + Verification Interval
```

```
Else
Delete record from database
```

Figure 7.7 WINS Scavenging Algorithm

The results of this scavenging algorithm are also detailed in Table 7.6.

Consistency Checking

Consistency checking helps maintain database integrity among WINS servers in a large network. When consistency checking is initiated using the WINS management console, WINS pulls all of the records directly from each owning server in its database, including any servers for which it has stored local records that are not among its replication partners.

All records pulled from remote databases are compared to records in the local database using the following checks for consistency:

- If the record in the local database is identical to the record pulled from the owner database, its time stamp is updated.
- If the record in the local database has a lower version ID than the record pulled from the owner database, the pulled record is added to the local database and the original local record is marked for deletion.
- If the records have the same version ID but a different name, the local record will be marked deleted and the pulled record will replace it.

Note that if a WINS database is extremely large, the consistency checking process might be network-intensive. In Windows 2000, consistency checking can be performed using the WINS management console, by checking the **Enable Periodic Database Consistency Checking** box on the **Name Record** tab of the server properties page.

WINS Database Files

The format of the WINS database increases the speed and efficiency of data storage by writing current transactions to log files rather than to the database directly. Therefore, the most current view of the state of the WINS database requires examination of the database plus any transactions in the log files. These files are also used for recovery; if the service fails (for example, due to a power failure), the log files can recreate the correct state of the WINS database.

Log files are always about 1 megabyte in size; however, they can grow quickly on a very busy WINS server. When a WINS server reaches the maximum size of its current log file, it creates another log file.

The effective size of the database of each connected WINS server is roughly identical. Each database has the same number of entries, neglecting latencies, and the size of the database is proportional to the number of entries. Unique entries typically occupy 42 bytes (they require no scope ID). Internet Group entries might occupy as many as 25 addresses and, therefore, more bytes. The real size might be much larger because unused space is only reclaimed efficiently by compacting.

The name–to–IP address mapping table stores NetBIOS names and the IP addresses currently assigned to them. The entries in this table are created from NetBIOS name registration requests received over TCP/IP nodes and from replicas received from other WINS servers. A *clustered index* on the name field enables quick retrieval of records required for name queries. A clustered index is an index in which the logical or indexed order of the key values is the same as the physical stored order of the corresponding rows that exist in a table. A primary index is built from the concatenation of the owner ID and version ID fields, and stored in ascending order. This allows quick access to records falling within ranges of version IDs for a particular owner.

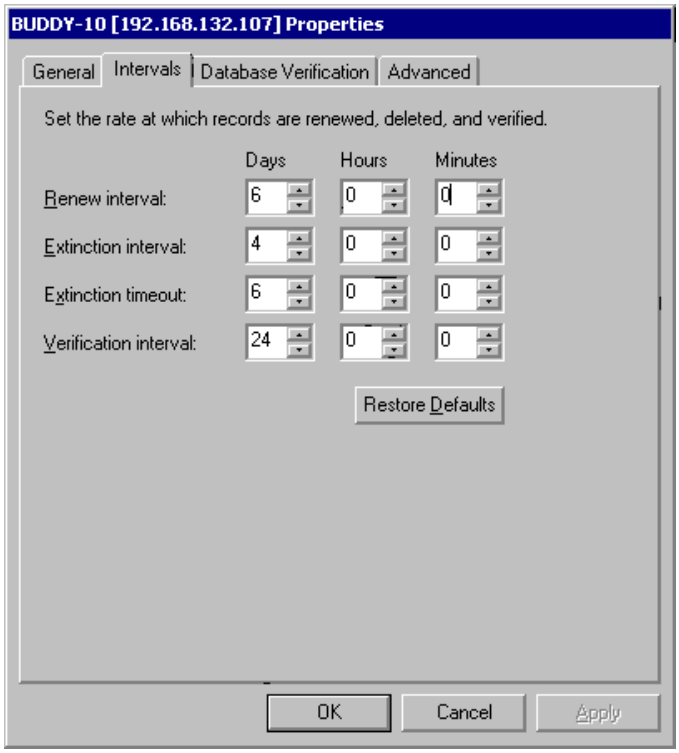
The IP address–to–owner ID mapping table contains a row for each WINS server that has entries in the name–to–IP address mapping table. Rows contain the IP address of WINS servers and their identifier as stored in the owner ID field of the entries that this server owns.

Timers

The WINS database records are governed by four configurable timer values:

- Renewal interval
- Extinction interval
- Extinction timeout
- Verification interval

Microsoft has chosen the defaults for these four values with care and, in general, they should not be modified. They keep the level of network traffic and the load on WINS servers at a minimum. They represent the best tradeoff, considering the various configurations in which WINS servers might be deployed, between these goals and minimizing the window during which the databases remain out of sync. Considerations such as long weekends, avoidance of unnecessary replication traffic, ability to handle a large number of clients quickly under worst-case scenarios—as well as tradeoffs between removing clutter quickly and retaining entries to ensure replication—have been factored into the determination of these intervals.



If your browser does not support inline frames, click here to view on a separate page.

Figure 7.8 Name Record Configuration Dialog Box

These four values are listed in Table 7.7, and are described in more detail in this section. They are configured in the dialog box shown in Figure 7.8.

Table 7.7 WINS Server Timers

Configuration option	Description
Renewal Interval	Specifies how often a client reregisters its name. The default is six days.
Extinction Interval	Specifies the interval between when an entry is marked as released and when it is marked as extinct. The default depends on the renewal interval and, if the WINS server has replication partners, on the maximum replication time interval. Maximum allowable value is six days.
Extinction Timeout	Specifies the interval between when an entry is marked extinct and when the entry is finally scavenged from the database. The default depends on the renewal interval and, if the WINS server has replication partners, on the maximum replication time interval. The default is six days.
Verification Interval	Specifies the interval after which the WINS server must verify that old names it does not own are still active. The default depends on the extinction interval. The maximum allowable value is 24 days.

Renewal Interval

The renewal interval is also known as the name refresh timeout, or the Time To Live (TTL). When a name is registered with the WINS server, the database entry is time stamped with the sum of the current time and the renewal interval. A client must refresh its name with the WINS server within this interval or the name is released. When a name is released, either by timing out or by the explicit release of the name by the client, no action is taken other than changing the entry to the released state and time stamping the entry with the sum of the current time and the extinction interval. This change is not replicated to other WINS servers. When a name is in the released state and a new registration comes in with a different address, the name can be immediately given to the new client without challenge because it is known that the old client is no longer using the name. The default renewal interval is six days.

Extinction Interval

The extinction interval is also known as the name age timeout and the tombstone interval. This is the interval at which released names enter the tombstone state. At this time, the entry is time stamped with the sum of the current time and the extinction timeout; and the entry's version ID is updated, ensuring that this information is propagated to all WINS servers at the next replication.

When tombstone entries are created, they are time stamped with the sum of the current time and the extinction timeout at the pulling WINS server. The default extinction interval is based on renewal and replication times. This is typically six days in Windows 2000.

Extinction Timeout

The extinction timeout is also known as the tombstone timeout. Tombstone records older than the extinction timeout are removed from the database. As noted earlier, manual extinction is available in Windows 2000 from the WINS console. The default extinction timeout is six days.

Verification Interval

Replication should ensure that the databases stay synchronized. However, under certain abnormal conditions, names no longer in use could remain in the database, creating database clutter. For example, if a tombstone record is removed before being replicated, the active state of the record in the replica databases never changes. This could happen if the replication partner was not reachable during the extinction timeout period.

When an active entry is replicated, it is time stamped with the sum of the current time and the verification interval on the pulling WINS server. If, at scavenge time, WINS finds records older than the verification interval, WINS sends a query to the WINS server that owns

that name, asking if the version ID is still valid. If the owning WINS server responds negatively (invalid record), the record is removed. If the owning WINS server sends a positive response (valid record), the time stamp is updated. If the WINS server cannot be contacted, the entries are left until the next verification interval or until the administrator triggers scavenging. No records are removed if the owning WINS server cannot be contacted. The default verification interval is 24 days.

Server Clocks

The replication and scavenging algorithms rely on a reasonably consistent system clock. Of course, setting the system clock forward or backward affects these algorithms. However, because the time stamps are always entered locally, the WINS servers do not need to be time-synchronized. As long as the time is consistent on each server, the intervals and timers function correctly.

Deletion of WINS Database Records

The WINS management console provides improved database management by supporting the following deletion operations:

- Simple deletion of WINS database records stored on a single-server database.
- Tombstoned deletion of WINS database records replicated to databases on other WINS servers.
- The ability to select multiple groups of displayed database records when performing either simple or tombstoned deletion.

In addition, the WINS management console allows you a simpler and more convenient tool for administratively removing dynamically registered records. In previous releases of WINS, the WINS management console utilities only removed static mappings.

WINS records can be removed in one of two ways: either through simple deletion or by using tombstoned deletion. The rest of this section discusses how to use both to manage your WINS database.

When simple deletion is used, records selected using the WINS console are removed from the current local WINS server that you are managing.

If WINS records deleted this way have been replicated to other WINS servers, these additional records will not be removed fully. The records on other WINS servers remain in those databases unless you specifically use the WINS console to remove them from each server, one at a time. In addition, records deleted on just one server might reappear when replication next occurs between WINS servers configured as replication partners.

When you use tombstoned deletion to remove a record owned by your selected server, the selected records are removed from all WINS servers that replicate the records as described in this section.

The owning WINS server changes the status of selected WINS records from active to tombstoned in its database. WINS then treats the records as inactive and released from use. Once these records are tombstoned locally, the owning WINS server neither responds to nor resolves NetBIOS name queries for these names from other WINS clients and WINS servers unless the records are registered again by the WINS client. The owning WINS server replicates the selected records as "tombstoned" to other WINS servers during subsequent replication cycles.

The records are not forcibly and immediately removed from WINS; instead, they are flagged for eventual deletion. The exact replication cycle interval is configured based on the Name Record properties of the server set in the WINS console. Records are not removed from WINS databases until their extinction interval has actually expired. This allows other WINS servers to learn that these records are no longer in use, update their replicated record mappings, and further replicate this updated WINS data to other servers. The records are marked extinct on all replicated WINS servers.

Once all WINS servers have completed a full replication cycle, the tombstoned records expire and are removed from the database on each WINS server during the next database scavenging operation. Once scavenging occurs on all servers, the records no longer appear in the WINS management console and are no longer physically stored in the WINS database.

Note that even if records are manually tombstoned (or otherwise marked as released by WINS), released records remain in the WINS database briefly before being removed during subsequent scavenging operations. Exactly how long they remain depends on the length of time required by the WINS server to determine extinction. Typically, the time to extinction for records is equal to the sum of the extinction interval, the extinction timeout, and the verification interval.

Example of Record Registration and Extinction

As an example, a WINS client registers the name TESTPC1 with the WINS server WINS1, and the server provides a refresh interval of three days. Once the name is registered, WINS1 replicates this record to its replication partners, such as WINS server WINS2. When the verification interval has expired, WINS2 verifies the record with WINS1. WINS1 takes no further action with this record; it simply waits until the client refreshes its name or reregisters it.

If the client does not refresh its name within the refresh interval, WINS1 sets the state for the name TESTPC1 to "released." If the client does not refresh the name within the extinction interval, the name is tombstoned; at that point, it is again replicated to WINS2 (because the version ID of the record for TESTPC1 is increased).

When the record is replicated, WINS2 copies the tombstoned entry from WINS1 and stamps it with the current time plus the verification interval. WINS2 does not query WINS1 until after the verification interval elapses. On the other hand, WINS1 waits for the duration of the extinction interval for the client to refresh or reregister its name. If the client does not do so, WINS1 removes the name from the database. WINS2 then queries WINS1 when the verification interval expires; if the record is not present at WINS1, then WINS2 removes the record from its database.

If WINS2 queries WINS1, and WINS1 does not respond (due to failure, maintenance, or simply a slow link), then the record is not removed. In this case, the entry's verification interval is reset, and WINS2 queries WINS1 after the verification interval has again expired.

Manual Tombstoning

With earlier versions of WINS, records were not deleted on multiple servers simultaneously. A window existed during which replication could occur between servers whose records were inconsistent. In other words, deleted records could return to a server from which they were just deleted.

The manual tombstoning option of Windows 2000 WINS prevents this problem. The length of the tombstoned state is greater than the propagation delay incurred with replication across the network. When the time limit is reached, tombstoned records are deleted by normal scavenging.

Manual tombstoning provides an excellent way of dealing with static records, too.

When the tombstoned records are replicated, the tombstone status is updated and applied by other WINS servers that store replicated copies of these records. Each replicating WINS server updates and individually tombstones these records. Once all WINS servers have replicated these records, the records are automatically removed from WINS after the period set by the verification interval of each server.

Manual tombstoning is available from both the WINS graphical user interface and the WINS command-line interface. To access this feature, open the **WINS** dialog box, select the owning server, then view all the records of that server. Highlight the record you want to delete, and delete it from the **Action** menu. At this point, you can either delete or tombstone the record. While the ability to manually tombstone records requires Windows 2000 WINS servers, tombstoned records replicate normally to Windows NT 3.51 and Windows NT

4.0 servers.

Best Practices for WINS Databases

With dynamic compaction and the WINS management console, WINS databases are much easier to maintain in Windows 2000, but they still require certain administrative practices and regular upkeep.

Perform Periodic Consistency Checking For Windows 2000, WINS consistency checking is available from the WINS management console. Use this feature periodically to check the WINS database for consistency.

Consistency checking consumes a great deal of network and computer system resources because the WINS server must replicate itself for each owner whose records are being checked for consistency. For this reason, check the consistency of WINS database records during times of low network traffic, such as at night or on weekends.

Perform Regular Offline Compaction Dynamic database compaction occurs on WINS servers as an automatic background process during idle time after a database update. This dynamic database compaction occurs while the database is in use; you do not need to stop the WINS server for dynamic compacting.

Although dynamic compacting greatly reduces the need for offline compaction, it does not fully eliminate the need for it. Offline compaction using the JETPACK utility reclaims more space than dynamic compaction and should be performed once a month for networks with 1,000 or more WINS clients. For smaller networks, manual compaction may be useful if only performed every few months.

Although manual compaction of the WINS server database is not as important for Windows 2000 Server as it was for earlier versions, it is still useful. You should perform monthly or weekly offline compaction for disk defragmentation and improved disk performance. Monitor any changes to the size of the server database file, Wins.mdb, which is located in the directory *%SystemRoot%*\System32 \Wins.

Checking the file size of Wins.mdb both before and after compaction allows you to measure growth and reduction. This information helps you determine the actual benefits to using offline compaction. Based on this information, you can gauge how often to repeat offline compaction for measurable gains.

Perform Regular Backups to Ease Restoration In addition to tape backups of the WINS server computer, the WINS management console offers a backup option that allows you to restore a WINS database after the database file has been corrupted. For more information on restoring data after corruption or loss of the WINS server database, see "Restoring Data" in the "Troubleshooting" section of this chapter.

You can also restore the database through a replication partner. If the WINS data is current on the replication partner, you can use this data to update the failed server. Two registry entries control this feature, **InitTimeReplication** and **InitTimePause**. **InitTimeReplication** is in the following subkey:

HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \Wins \Partners \Pull and Push

The value of this entry is 1 by default, which causes WINS to replicate with the partner at the time specified in the key. The **InitTimePause** entry is stored in the following subkey:

HKEY_LOCAL_MACHINE \SYSTEM \CurrentControlSet \Services \Wins \Parameters

It tells WINS to pause while the replication takes place.

Use the Scavenge Function The **scavenge** function is an automatic function of Windows 2000 WINS. It releases records old enough to be released, removes extinct records, and verifies records after the verification interval passes.

Using the default WINS configuration in Windows 2000, **scavenge** runs after WINS has been running for 72 hours, or half the renewal interval. If WINS is stopped and restarted before 72 hours have elapsed, the 72-hour window to the next scheduled scavenge is reset. If the WINS service is stopped on a daily basis, scavenging cannot take place.

To verify that scavenging is occurring, on the WINS server properties page, on the **Advanced** tab, select the **Log Detailed Events to Event Log** check box. This feature adds overhead processing, and should be used only when verifying the scavenging process. If scavenging is not taking place, establish a scavenge policy as part of your WINS database maintenance.

Avoid Using Static WINS Entries Static WINS entries require administrative action to assure their successful and intended use. However, static entries can be useful for specific purposes, such as protecting registration of names used by critically important servers.

For example, you can add a static entry to the WINS database to prevent other computers from registering the name of a critical server while that server is down. Reserving names in this manner prevents anyone from hijacking the server name (via DHCP) by registering another computer with the same name on the network. If the server is not responding at the time, a WINS server issuing a name challenge does not receive a response indicating that the name is in use, and the address is taken over by the new computer.

The biggest disadvantage of using static WINS entries is that it complicates administration of name and address changes in your network. For example, if either the IP address or the computer name of a static WINS entry changes, you might need to update other configurations, such as DHCP servers, DNS servers, end systems, LMHOSTS files, and so forth.

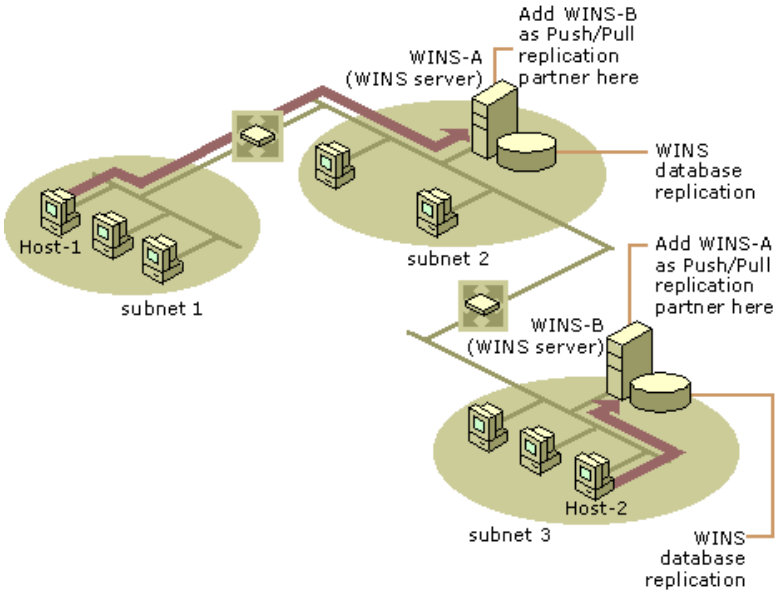
If you do use static WINS entries, use reservations to minimize the impact on DHCP. For each IP address used in static WINS mapping, use a corresponding client address reservation to reserve the IP address at the DHCP server. Also, if you do use static entries, carefully monitor and track the servers where these entries are added (the owning servers). Ideally, all static entries should only be entered on a single server. This makes later removal of these entries easier. For more information about address reservations, see "Dynamic Host Configuration Protocol" in this book.

WINS Replication

You can configure all WINS servers on a network to fully replicate their database entries to other WINS servers. This replication ensures that a name registered with one WINS server is eventually registered to all other WINS servers. This section examines the replication process in detail.

Overview of the Replication Process

Replicating databases between WINS servers maintains a consistent set of WINS information throughout a network. An example of WINS database replication is shown in Figure 7.9. Two WINS servers, WINS-A and WINS-B, are both configured to fully replicate their records with each other.



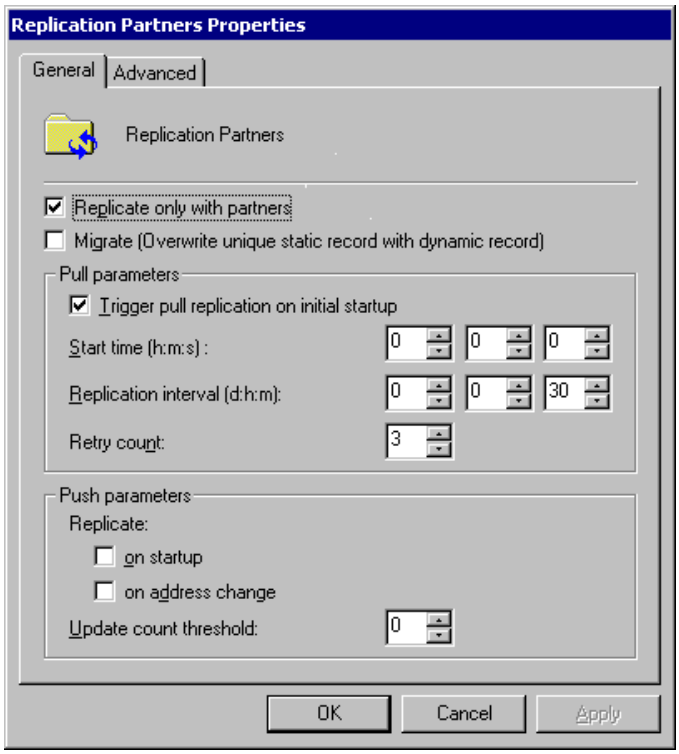
If your browser does not support inline frames, click here to view on a separate page.

Figure 7.9 WINS Replication Overview

In Figure 7.9, a WINS client, HOST-1 on subnet 1, registers its name with its primary WINS server, WINS-A. Another WINS client, HOST-2 on Subnet 3, registers its name with its primary WINS server, WINS-B. If either of these hosts later attempts to locate the other host using WINS—for example, HOST-1 queries to find an IP address for HOST-2—replication of WINS registration information between the WINS servers makes it possible to resolve this query.

Note WINS replication is always incremental, meaning that only changes in the database are replicated each time replication occurs, not the entire database.

For replication to work, each WINS server must be configured with at least one other WINS server as its replication partner. This ensures that a name registered with one WINS server is eventually replicated to all other WINS servers in the network. A replication partner can be added and configured as either a push partner, a pull partner, or a push/pull partner, which uses both methods of replication. The push/pull partner is the default configuration and is the type recommended for use in most cases.



If your browser does not support inline frames, click here to view on a separate page.

Figure 7.10 Replication Partners Properties Dialog Box

When WINS servers replicate, a latency period exists before the name-to-address mapping of a client from any given server is propagated to all other WINS servers in the network. This latency is known as the convergence time for the entire WINS system. For example, a name release request by a client does not propagate as quickly as a name registration request. This is because names are commonly released and then reused with the same mapping, such as when computers are restarted or when they are turned off for the evening and restarted in the morning. Replicating each of these name releases would unnecessarily increase the network load of replication.

Also, when a WINS client computer is shut off improperly, such as during an unexpected power outage, the computer's registered names are not released normally with a request to the server. Therefore, the presence of a record in the WINS database does not necessarily mean that a client computer is still using the name or its associated IP address. It only means that a computer recently

registered that name and its associated IP address.

Note The primary and secondary WINS servers assigned to any client must have push and pull relationships with each other. You might want to keep a list of pairs of push/pull WINS servers for use when assigning servers to clients.

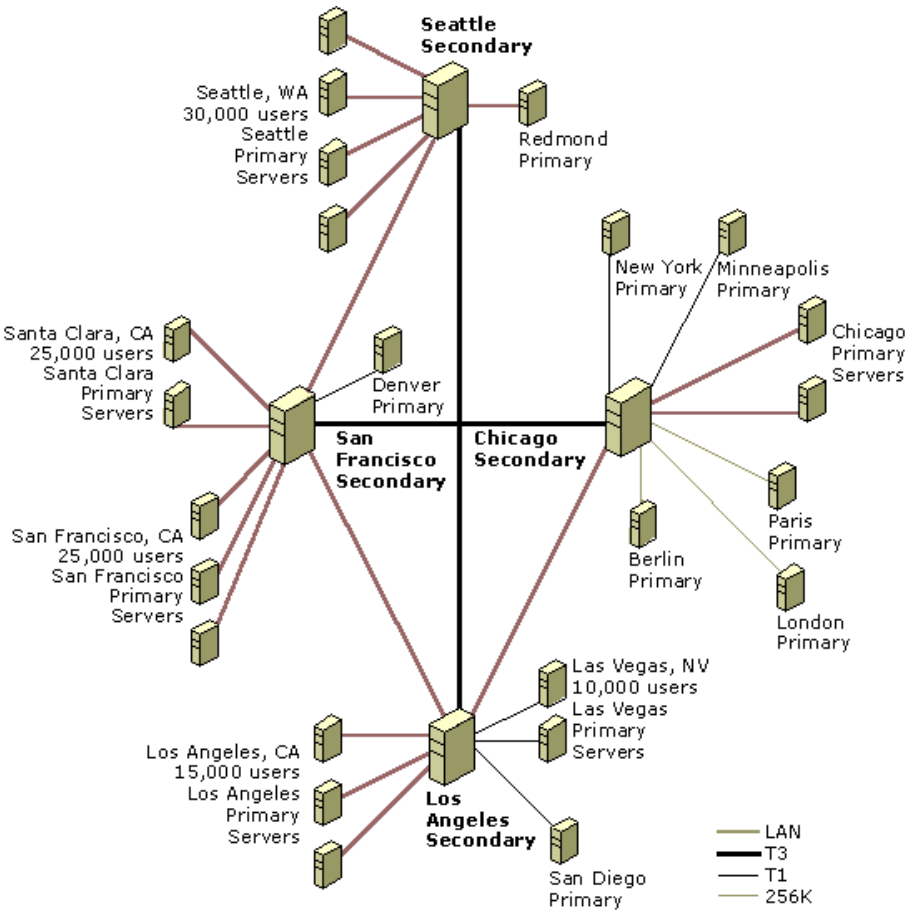
To replicate database entries, each WINS server in a network must be configured as either a pull partner or a push partner with at least one other WINS server.

WINS Server Push and Pull Partners

The WINS database is collectively managed by the WINS servers, each of which has a copy of the WINS database. To keep these copies consistent, servers replicate their records among themselves. Each WINS server is configured with a set of one or more replication partners. When new computers are added or substituted on the network, they register their name and IP address with another server, which in turn propagates the new record to all other WINS servers in the enterprise. The result is that every server has the record pertaining to that new computer.

Detailed Replication Example

The figure below shows an extremely large WINS implementation, serving more than 100,000 nodes. In a configuration with so many WINS servers, it is tempting to create many push/pull relationships for redundancy. This can lead to a system that, while functional, is overly complex and difficult to understand and troubleshoot.



If your browser does not support inline frames, click here to view on a separate page.

Figure 7.11 Large-Scale WINS Deployment Using Hub Topology

The hub structure imposes order on the sample configuration shown in Figure 7.11. Four major hubs are located in Seattle, San Francisco, Chicago, and Los Angeles. These hubs serve as secondary WINS servers for their regions while connecting the four geographic locations. All primary WINS servers are configured as push/pull partners with the hubs, and the hubs are configured as push/pull partners with other hubs.

For example, assume the primary WINS servers in Figure 7.11 replicate with the hubs every 15 minutes, and the hub-to-hub replication interval is 30 minutes. The convergence time of the WINS system is the time it takes for a node registration to be replicated to all WINS servers. In this case the longest time would be from a Seattle primary server to a Chicago primary server. The convergence time can be calculated by adding up the maximum time between replication from the Seattle primary to Seattle secondary, Seattle secondary to San Francisco secondary, San Francisco secondary to Chicago secondary, and finally Chicago secondary to Chicago primary. This yields a total convergence time of 15 + 30 + 30 + 15 minutes, or 1.5 hours.

However, the convergence could be longer if some of these WINS servers are connected across slow links. It is probably not necessary for the servers in Paris or Berlin to replicate every 15 minutes. You might configure them to replicate every two hours or even every 24 hours, depending on the volatility of names in the WINS system.

This example network contains some redundancy, but not much. If the link between Seattle and Los Angeles is down, replication still occurs through San Francisco, but what happens if the Seattle hub itself goes down? In this case, the Seattle area can no longer replicate with the rest of the WINS system. Network connectivity, however, is still functional—all WINS servers contain the entire WINS database, and name resolution functions normally. All that is lost are changes to the WINS system that occurred since the Seattle hub went down. A Seattle user cannot resolve the name of a file server in Chicago that comes online after the Seattle hub does down. Once the hub returns to service, all changes to the WINS database are replicated normally.

Small-Scale Replication Example

While the large-scale deployment shown in the four-hub diagram of Figure 7.11 is possible, it is also valuable to examine a much smaller example of replication. The simplest case involves just two servers, as shown in Figure 7.12.

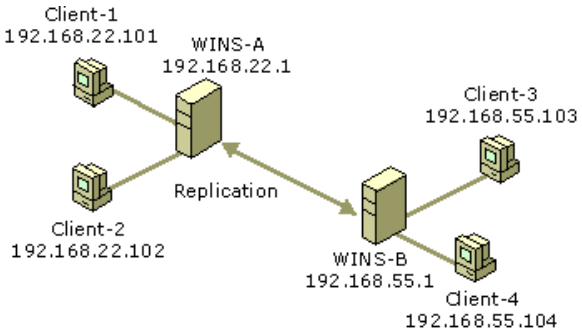


Figure 7.12 Database Replication Between Two WINS Servers

Tables 7.8 and 7.9 are the database tables for WINS-A and WINS-B on January 1, 2000. All four clients are powered on in the morning between 8:00 A.M. and 8:15 A.M. Client2 has just been shut down. WINS-A and WINS-B have the following parameters:

- WINS-A and WINS-B are push/pull partners to each other.
- The replication interval is 30 minutes.
- The renewal interval is 4 days.
- The extinction interval is 4 days.
- The extinction timeout is 1 day.
- The verification interval is 24 days.

Before replication, WINS-A has two entries in its database. These entries are for Client1 and Client2, as shown in Table 7.8.

Table 7.8 WINS-A Database Before Replication

Name	Address	Flags	Owner	Version ID	Time stamp
Client1	192.168. 22.101	Unique, active, H-node, dynamic	WINS-A	4B3	1/5/00 8:05:32 AM
Client2	192.168. 22.102	Unique, released, H-node, dynamic	WINS-A	4C2	1/5/00 8:23:43 AM

Before replications, WINSB has the two entries shown in Table 7.9, one each for Client3 and Client4.

Table 7.9 WINS-B Database Before Replication

Name	Address	Flags	Owner	Version ID	Time stamp
Client3	192.168. 55.103	Unique, active, H-node, dynamic	WINS-B	78F	1/5/00 8:11:12 AM
Client4	192.168. 55.104	Unique, active, H-node, dynamic	WINS-B	79C	1/5/00 8:12:21 AM

Client1, Client3, and Client4 were time stamped with the sum of the current time and the renewal interval at the time they booted, and Client2 was time stamped with the sum of the current time and the extinction interval when it was released. The version IDs indicate the value of the registration counter at the time of registration. The registration counter is incremented by 1 (hexadecimal) each time it generates a new version ID in the database. Each WINS server has its own registration counter. The version ID jumps from 4B3 for Client1 to 4C2 for Client2. This indicates that 14 registrations (or extinctions or releases to active transition) took place between the registration of Client1 and Client2.

Replication takes place at 8:30:45 by WINS-A's clock. WINS-B's clock is 8:31:15 at this time. Of course replications will not all take place in the same second, but the servers use these times to generate the time stamps. Note that replication does not mean both pull at the same time—each pulls according to its own schedule. After replication, WINS-A's database contains the entries shown in Table 7.10.

Table 7.10 WINS-A Database After Replication

Name	Address	Flags	Owner	Version ID	Time stamp
Client1	192.168. 22.101	Unique, active, H-node, dynamic	WINS-A	4B3	1/5/00 8:05:32 AM
Client2	192.168. 22.102	Unique, released, H-node, dynamic	WINS-A	4C2	1/5/00 8:23:43 AM
Client3	192.168. 55.103	Unique, active, H-node, dynamic	WINS-B	78F	1/25/00 8:30:45 AM
Client4	192.168. 55.104	Unique, active, H-node, dynamic	WINS-B	79C	1/25/00 8:30:45 AM

After replication, WINS-B's database contains the entries shown in Table 7.11.

Table 7.11 WINS-B Database Before Replication

Name	Address	Flags	Owner	Version ID	Time stamp
Client1	192.168. 22.101	Unique, active, H-node, dynamic	WINS-A	4B3	1/25/00 8:31:15 AM
Client3	192.168. 55.103	Unique, active, H-node, dynamic	WINS-B	78F	1/5/00 8:11:12 AM
Client4	192.168. 55.104	Unique, active, H-node, dynamic	WINS-B	79C	1/5/00 8:12:21 AM

Client1 has been replicated to WINS-B, and Client3 and Client4 have been replicated to WINS-A. The replicas have all kept their original owner and version ID and have been time stamped with the sum of the current time and the verification interval. Client2 has not been replicated, because it is in the released state. This is a little unusual (but possible) because Client2 shut down before its first replication. If Client2 had not been shut down until after the replication, WINS-B would have a replica of Client2 in the active state. This replica would remain in the active state even after Client2 released, because the change in state would not be replicated.

Assuming Client2 remains shut down for the duration of the extinction interval, it is placed in the tombstone state. At the first scavenging after 8:23:43 AM on January 5, 2000 (assuming an extinction interval of four days), the database on WINS-A contains the entries shown in Table 7.12.

Table 7.12 WINS-A Database After Scavenging

Name	Address	Flags	Owner	Version ID	Time stamp
Client1	192.168. 22.101	Unique, active, H-node, dynamic	WINS-A	4B3	1/9/00 6:35:26 AM
Client2	192.168. 22.102	Unique, tombstone, H-node, dynamic	WINS-A	657	1/6/00 9:50:53 AM
Client3	192.168. 55.103	Unique, active, H-node, dynamic	WINS-B	78F	1/25/00 8:30:45 AM
Client4	192.168. 55.104	Unique, active, H-node, dynamic	WINS-B	79C	1/25/00 8:30:45 AM

Note that Client2 has entered the tombstone state and that both its time stamp and its version ID have changed. The time stamp is now the sum of the current time and the extinction timeout, and the new version ID means that this entry is replicated at the next replication. Note also that Client1 has a new time stamp while retaining its version ID. It has been renewed throughout the last four days. The renewal rate depends the client stack.

After replication at 10:00:23 A.M., the database on WINS-B contains the entries shown (note that Client3 and 4 were renewed) as shown in Table 7.13.

Table 7.13 WINS-B Database After Replication

Name	Address	Flags	Owner	Version ID	Time stamp
Client1	192.168. 22.101	Unique, active, H-node, dynamic	WINS-A	4B3	1/25/00 8:31:15 AM
Client2	192.168. 22.102	Unique, tombstone, H-node, dynamic	WINS-A	657	1/6/00 10:00:23 AM
Client3	192.168. 55.103	Unique, active, H-node, dynamic	WINS-B	78F	1/9/00 8:11:12 AM
Client4	192.168. 55.104	Unique, active, H-node, dynamic	WINS-B	79C	1/9/00 8:12:21 AM

If Client2 remains down for one more day, exceeding the extinction timeout, it will be removed from the databases when it is next scavenged.

Once Client2 is removed, the database on WINS-A contains the entries shown in Table 7.14.

Table 7.14 WINS-A Database After Client 2 Is Removed

Name	Address	Flags	Owner	Version ID	Time stamp
Client1	192.168. 22.101	Unique, active, H-node, dynamic	WINS-A	4B3	1/11/00 9:45:56 AM
Client3	192.168. 55.103	Unique, active, H-node, dynamic	WINS-B	78F	1/25/00 8:30:45 AM
Client4	192.168. 55.104	Unique, active, H-node, dynamic	WINS-B	79C	1/25/00 8:30:45 AM

After Client2 is removed, the database on WINS-B contains the entries shown in Table 7.15.

Table 7.15 WINS-B Database After Client 2 Is Removed

Name	Address	Flags	Owner	Version ID	Time stamp
Client1	192.168. 22.101	Unique, active, H-node, dynamic	WINS-A	4B3	1/25/00 8:31:15 AM
Client3	192.168. 55.103	Unique, active, H-node, dynamic	WINS-B	78F	1/11/00 9:44:27 AM
Client4	192.168. 55.104	Unique, active, H-node, dynamic	WINS-B	79C	1/11/00 9:46:44 AM

During the first scavenging after 8:30 A.M. on January 25, 2000, WINS-A verifies with WINS-B that Client3 and Client4 are still valid active names. WINS-B does the same for Client1 with WINS-A.

Pulling WINS Database Entries by Version Number

The WINS server database maintains a table that stores the IP addresses and owner IDs of remote WINS servers that own entries in the local database.

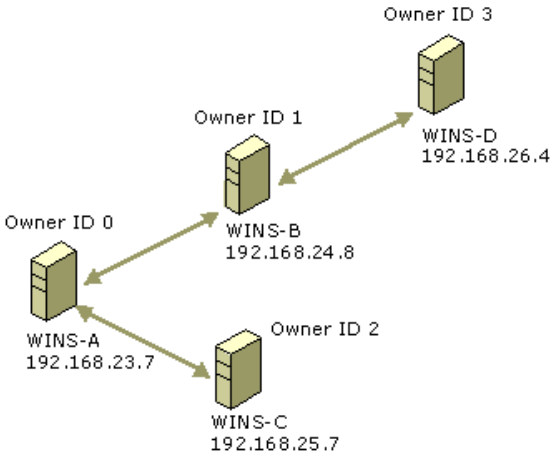


Figure 7.13 A Sample Replication Pattern

Based on an example replication pattern like that shown in Figure 7.13, a table mapping IP addresses to owner IDs for a sample server — again called WINS-A—would contain entries like those shown in Table 7.16.

Table 7.16 Example Remote WINS Server IP Addresses and Owner IDs

IP address	Owner ID
192.168.23.7	0
192.168.24.8	1
192.168.25.7	2
192.168.26.4	3

During WINS initialization, the WINS server scans the NetBIOS name–to–IP address mapping table to determine the maximum version number corresponding to each owner registered in its database. Initializing it with the information retrieved, the WINS server creates a table in memory that maps push partner IDs to version numbers. This table is never committed to the database. Example WINS server WINS-A creates a table like that shown in Table 7.16; other servers will have different tables.

As you might expect, the push partner–to–version number table contains an entry for each push partner, and each entry contains the maximum version ID found for all owners in the local database of the push partner. For example, if the local WINS server has an owner ID of 0, then given the preceding example IP address–to–owner ID mapping table, the push partner–to–version number mapping table might look like that shown in Figure 7.14.

	0	1	2
0	100	900	630
1			
2			

Figure 7.14 Example Push Partner–to–Version Number Table

Figure 7.14 shows that the local database of the WINS server, identified by owner ID 0, contains entries owned by three WINS servers with owner IDs of 0, 1, 2. The highest version numbers for the entries are 100, 900, and 630, respectively.

WINS server WINS-A is now ready to determine whether it needs to update its database. It sends a message to each of its push partners, asking it to respond with the highest version numbers pertaining to IP addresses in its local database. As push partners respond, the WINS server fills and expands its own table. The table might expand to fill more columns, each corresponding to another server.

For example, if WINS server WINS-B at IP address 192.168.24.8 (owner ID 1) responds with a record for WINS server WINS-D at IP address 192.168.26.4, WINS-A adds a column to the local push partner–version number mapping table for indirect push partner WINS-D with an owner ID 3. At the same time, IP address and owner ID WINS-D are stored in the IP address–to–owner ID mapping table. The relevant cells in the new table are initialized. After WINS-B, at 192.168.24.8, responds with the following three records:

```
192.168.24.8 999
192.168.26.4 700
192.168.23.7 89
```

the WINS server adds a record containing IP address 192.168.26.4 and owner ID 3 to the IP address–to–owner ID mapping table and updates the local push partner–to–version number mapping table to resemble that shown in Figure 7.15.

	0	1	2	3
0	100	900	630	0
1	89	999	0	700
2				

Figure 7.15 Example Push Partner–to–Version Number Table After Response from WINS-B

After all push partners have responded, the IP address–to–version number mapping table contains the information shown in Figure 7.16.

	0	1	2	3
0	100	900	630	0
1	89	999	0	700
2	93	879	820	0

Figure 7.16 Example Push Partner–to–Version Number Table After All Responses

The WINS server examines this table to determine which push partner has the latest data for each owner. A WINS server always has the highest version ID for entries it owns. For example, in Figure 7.16, the entry with the ID 0 is recorded in three databases: 0, 1, and 2. Because the entry is owned by WINS-A, the entry for 0,0 has the highest version ID number (100) for that entry.

However, some WINS servers might not be partners of the requesting pull partner. The WINS server determines the starting version ID

required to synchronize the local database, and requests that the push partner send the database records with version IDs that are equal or greater. If a push partner has the latest data for more than one owner, a single request can be sent to retrieve the records for all. Of course, in this simple example, the WINS server that has the most current data for a database never changes. When WINS-A pulls data, WINS-B has the latest data for itself and WINS-D; WINS-C has the latest data for itself. In a more complex model, the replication paths might form loops, and replication takes place at differing intervals.

When the push partner receives a request from another WINS server—a pull partner—it retrieves the required records from its local database and sends them to the requesting server. The push partner retrieves records by seeking the record that starts the range and moving sequentially over the records, retrieving them, until the push partner retrieves the last record in the range. When the pull partner receives the data from the push partners, the pull partner updates its database.

All entries with version IDs greater than those in the pulling database are replicated. However, not every change to a database increments a record's version ID.

How Records Change and Update

A WINS server always enters name registrations in its database in an active state and time stamped with the sum of the current time and the renewal interval. The version ID is taken from the version ID counter, and the counter is then incremented.

If a name is explicitly released or not refreshed during the renewal interval, the name enters the released state. The WINS server gives the database entry a time stamp using the sum of the current time and the extinction interval, and leaves the version ID unchanged. Thus, released records are not replicated. If a record remains released past the extinction interval, the WINS server changes the state of the record to tombstone, gives the record a time stamp using the sum of the current time and the extinction timeout, and increments the version ID of the record so that the record will be replicated. If a record remains in the tombstone state for a period longer than the extinction timeout, it is deleted from the database.

WINS replicates only records in the active and tombstone states. In the WINS database, WINS enters these replica records with the fields received from the owner database, with the exception of owner ID and time stamp. (The owner ID comes from the local IP address–to–owner ID mapping table because the value used locally to represent a particular WINS server differs from server to server. For example, WINS-D might be represented by a 2 on WINS-B and by a 3 on WINS-A.) WINS gives an active record a time stamp that is the sum of the local current time and the verification interval. WINS gives a tombstone record a time stamp that is the sum of the local current time and the extinction timeout.

Conflicts Detected During Replication

Although name conflicts are normally handled at the time of name registration (see "Client Conflicts Detected During Registration" earlier in this chapter), it is possible for the same name to be registered at two different WINS servers. This would happen if a WINS client registered the same name at a second WINS server before the database from the first WINS server replicated to the second. In this case, WINS resolves the conflict at replication time.

Conflict at replication can be between two unique entries, between a unique entry and a group entry, or between two group entries.

Conflict Between Unique Entries WINS resolves conflicting unique entries according to three factors:

- **State of the entries.** The database entry can be in the active, released, or tombstone state; the replica can be either in the active or tombstone state.
- **Ownership of entries.** The WINS server might or might not own the database entry.
- **Addresses of the entries.** The addresses of the entries might or might not be the same.

Conflict Between Two Replicas When two replicas conflict, the new replica overwrites the replica in the database, regardless of whether the addresses match or not. The only exception to this rule is if the replica in the database is active and the new replica is a tombstone. If the new replica is a tombstone, the replica in the database does not overwrite the new replica, unless they are both owned by the same WINS server.

Conflict Between an Owned Entry and a Replica with the Same IP Address The replica replaces the database record, unless the database record is active and the replica is a tombstone. In that case, WINS increments the version ID of the database record so that the record is propagated at replication time.

Conflict Between an Owned Entry and a Replica with Different IP Addresses The replica replaces the database record unless the database record is active. If the record is active and the replica is a tombstone, WINS increments the version ID of the database record so that the record can be propagated by replication. If the replica is also active, the server receiving the replica challenges the client that owns the name in the local database to determine whether the client still uses the name. If it does, WINS sends the client node designated in the replica record a *name conflict demand*, a message that puts the client in a conflict state. This forces the node to place the name in the conflict state. A name in the conflict state is marked and the name is no longer used.

Conflict Between a Unique Entry and a Group Entry When a unique entry and a group entry conflict, WINS keeps the group entry. If the WINS server owns the unique entry and the entry is not in the released or tombstone state, the WINS server asks the client named in the unique entry to release the name.

Conflict Between Two Special Group Entries The replica replaces the database record unless the database record is active. If the record is active, WINS increments its version ID so that the record is propagated at replication time. If the replica is also active, the WINS server updates the member list of the database record with any new members from the replica. If the list of active members grows to more than 25, the extra members are not added but are dropped silently.

Conflict Involving a Multihomed Record If a multihomed replica conflicts with a tombstone or released entry in the database, WINS replaces the entry in the database with the replica, unless the entry is a normal group and is in the released state. This is no different from the other scenarios in which a single-address entry conflicts with a released normal group entry.

If a multihomed replica in the tombstone state conflicts with an active database entry owned by the same server as that of the multihomed replica, the database entry is replaced. If the active database entry is a replica owned by a different owner, WINS does not replace the database entry. If the active database entry is owned by the local WINS server and is a unique entry, the WINS server increments the version ID of the database record to prompt propagation.

If an active multihomed replica conflicts with an active, unique, multihomed replica in the local database with the same owner, the database entry is replaced. If the owner is different, it is not replaced. If the entry in the database is owned by the local WINS server, and if the members of the record (a single member in the case of a unique record) is a subset of the members in the replica, WINS changes the time stamp of the database record and increments its version ID to force propagation. If the members of the replica are not a subset, the addresses in the database record are challenged. If all challenges succeed—that is, the clients challenged do not respond to any challenge—the database record is replaced. If at least one challenge fails, WINS tells the client to release the name from all addresses prior to replacing the database record with the replica.

If a multihomed replica conflicts with an active group entry in the database, WINS increments the version ID of the entry in the database to cause propagation.

If a single-address replica conflicts with an inactive multihomed record in the database, WINS replaces the database record with the replica. If the replica conflicts with an active multihomed entry in the database owned by the same owner, WINS replaces the database record with the replica. If the multihomed entry in the database is a replica owned by a different server, WINS does not replace it. If,

however, the multihomed entry is owned by the local WINS server, and the pulled replica is a unique record, then WINS issues a challenge to the clients using the addresses in the multihomed record. If all challenges succeed, WINS replaces the database record. If at least one challenge fails, WINS sends the addresses in the database record requests for the name, and then the database record is updated. Note that the address in the unique replica, if present in the member list, is ignored in the above situation.

Persistent Connections

Windows 2000 WINS introduces persistent connections between WINS server replication partners. Earlier versions of WINS required servers to establish a new connection whenever they replicated databases. Because establishing and terminating each connection required a modest number of CPU cycles and the sending of network packets, network managers set their systems to accumulate a configurable number of records before establishing connections with replication partners. Waiting for records to accumulate introduces a delay to the updating of the entire database—perhaps as long as several minutes—which can cause windows of inconsistency with replication partners.

A Windows 2000 WINS server can be configured in the WINS management console to request a persistent connection with one or more replication partners; this eliminates the overhead of opening and terminating connections. Persistent connections increase the speed of replication because a server sends records to its partners immediately, without establishing temporary connections each time. This immediately updates every record across the network, making records more consistent. The bandwidth required is minimal because the connection is usually idle.

It is also possible to configure a persistent connection to replicate only when it reaches a certain update count threshold. Normally the minimum update count threshold is 20 records. However, when persistent connections are employed, that minimum is waived.

Autodiscovery of WINS Partners

The autodiscovery feature enables a WINS server to discover its replication partners automatically, rather than being manually configured with a predetermined set of replication partners. To turn this feature on from the Microsoft Management Console, on the **Replication Partners Properties** page, check the **Enable automatic partner configuration** check box.

Periodically, WINS servers announce their presence on the network. The WINS announcements are sent on a multicast address reserved for WINS (224.0.1.24). WINS servers with autodiscovery enabled listen for these announcements and learn about other WINS servers on the network. Any WINS servers discovered this way are automatically added to the partners list as both a push and pull partner. This feature should be used only when you are sure that no unauthorized WINS servers will be placed on the network. Otherwise, the unauthorized servers are picked up as partners.

Best Practices for WINS Replication

Configuring replication correctly can avert many problems, and doing so enables a group of WINS servers to function more effectively.

Configure Push/Pull Replication Partners

In general, push/pull replication is the simplest and most effective way to ensure full WINS replication between partners. This also ensures that the primary and secondary WINS servers for any particular WINS client are push and pull partners of each other, a requirement for proper WINS functioning in the event of a failure of the primary server of the client.

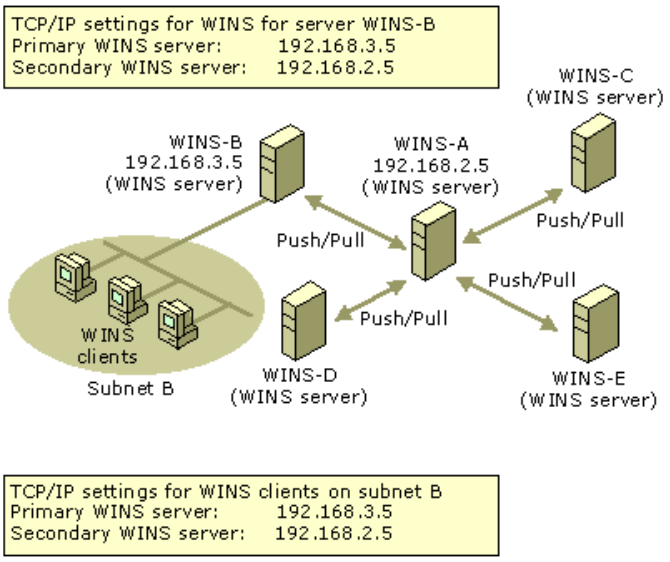
For most WINS installations, avoid the use of limited replication partnerships (push only or pull only) between WINS servers. In some large enterprise WINS networks, limited replication partnering can effectively support replication over slow network links. However, when you plan limited WINS replication, pay attention to the design and configuration. Each server must still have at least one replication partner, and each slow link that employs a unidirectional link should be balanced by a unidirectional link elsewhere in the network that carries updated entries in the opposite direction.

Use a Hub-and-Spoke Design for WINS Replication and Convergence

Convergence is a critical part of WINS planning. The central question of convergence time for a WINS network design is "How long does it take for a change in WINS data at one WINS server to replicate and appear at other WINS servers on the network?" The answer is the sum of the replication periods from one server to the next over the path containing the longest replication periods. For more information on convergence, see "Detailed Replication Example" in this chapter.

In most cases, the hub-and-spoke model provides a simple and effective planning method for organizations that require full and speedy convergence with minimal administrative intervention. For example, this model works well for organizations with centralized headquarters or a corporate data center (the hub) and several branch offices (the spokes). Also, a second or redundant hub (that is, a second WINS server in the central location) can increase the fault tolerance for WINS.

For an example of a simple hub-and-spoke configuration, see Figure 7.17.



If your browser does not support inline frames, click here to view on a separate page.

Figure 7.17 A Hub-and-Spoke Deployment of WINS Servers

The convergence time for the system shown in Figure 7.17 is the sum of the two longest convergence times to the hub. For instance, if WINS-B and WINS-D replicate with WINS-A every 30 minutes, and WINS-C and WINS-E are configured to replicate every 4 hours, the convergence time is 8 hours.

Replication Across a Firewall

In some large networks, WINS replication is desirable across a firewall. WINS replication occurs over TCP port 42, so this port must not be blocked on any intervening network device between two WINS replication partners when configuring replication across network firewalls.

Managing WINS Servers

Windows 2000 provides an updated version of WINS Manager, a graphical administrative utility that you can use to manage WINS on your network. The updated version of WINS Manager is a Microsoft Management Console snap-in, giving you the ability to further integrate and customize WINS administration to your network management needs.

The WINS management console contains significant enhancements, as compared to earlier versions, many of which were suggested by network managers. These new features include:

- Persistent connections
- Manual tombstoning
- Improved management utilities
- Enhanced filtering and record searching
- Dynamic record deletion and multiple selection
- Record verification and version number validation
- Export function
- Increased fault tolerance
- Dynamic renewal of clients

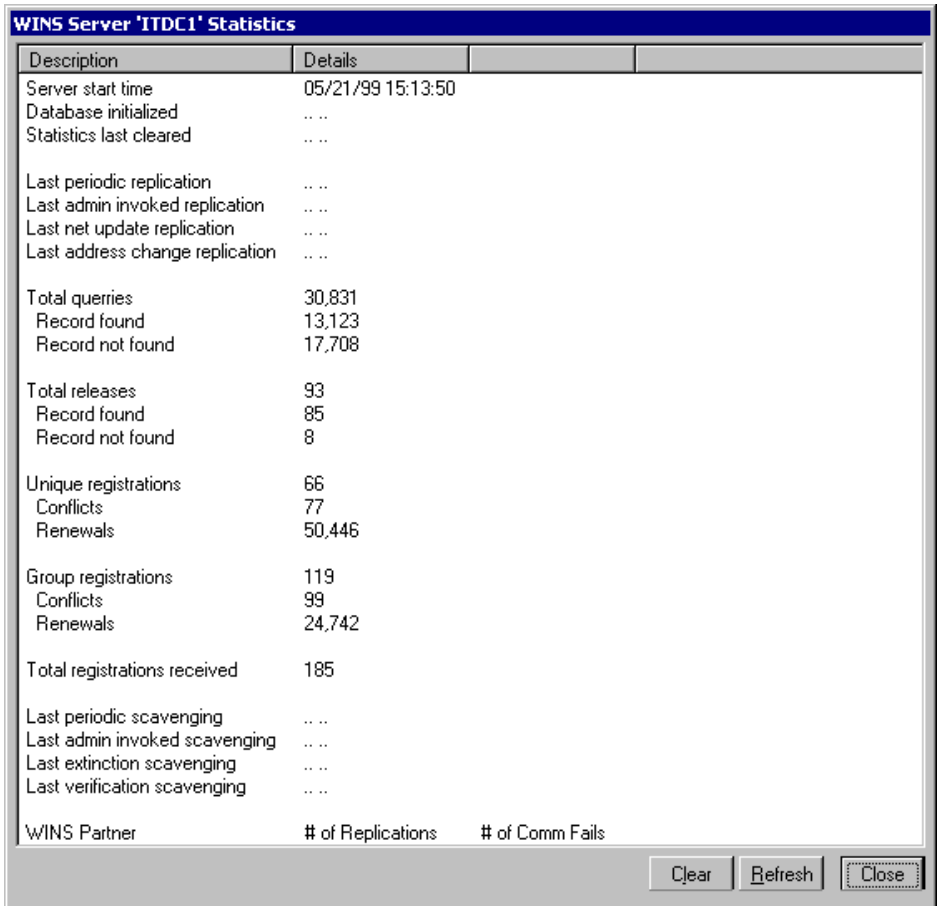
The WINS management console provides the utilities you need to maintain, view, back up, and restore the WINS server database. To view and change parameters for WINS servers, use the WINS management console. For more information about specific administration and configuration tasks, see WINS management console Help.

System Monitor and SNMP agent service are also valuable tools for managing a WINS server. You can use System Monitor to monitor WINS server performance.

You can use the SNMP service to monitor and configure WINS servers by using third-party SNMP manager utilities. When using a third-party SNMP manager utility, some WINS queries may time out; if so, you should increase the timeout on the SNMP utility you are using. Microsoft Information Base objects are supported by Windows 2000 SNMP Service. MIB objects are formally described objects that provide support for SNMP to allow the monitoring of processes, such as error counts, status records, and the contents of the IP routing table of a computer. For more information about MIB object types see "MIB Object Types" in this book. For more information about System Monitor and SNMP agent service, see "Simple Network Management Protocol" in this book and "Monitoring Network Performance" in the *Microsoft® Windows® 2000 Server Resource Kit Server Operations Guide*.

Viewing WINS Server Operational Status

The WINS management console displays administrative and operational information about WINS servers. To display basic statistics about a specific WINS server, open the WINS management console, highlight that server, on the pull-down menu, select **Action**, and then click **Show Server Statistics**. This provides WINS server information similar to that shown in Figure 7.18.



If your browser does not support inline frames, click here to view on a separate page.

Figure 7.18 WINS Server Statistics

Table 7.17 describes the basic WINS server statistics shown in Figure 7.18. These include both the basic and detailed statistics from Windows NT 4.0.

Table 7.17 WINS Server Statistics

Statistic	Description
Database initialized	The last time static mappings were imported into the WINS database.
Statistics last cleared	The last time the administrator cleared statistics for the WINS server with the **Clear Statistics** command on the **View** menu.
Last replication times	The times at which the WINS database was last replicated.
Periodic	The last time the WINS database was replicated based on the replication interval specified in the **Preferences** dialog box.
Admin trigger	The last time the WINS database was replicated because the administrator clicked the **Replicate Now** button in the **Replication Partners** dialog box.
Net update	The last time the WINS database was replicated as a result of a network request, which is a push notification message that requests propagation.
Total queries	The number of name query request messages received by this WINS server. "Record found" indicates the number of names that were successfully matched in the database, and "Record not found" indicates the number of names the server could not resolve.
Total releases	The number of messages received that indicate a NetBIOS program has stopped. "Record found" indicates how many names were successfully released, and "Record not found" indicates how many names this WINS server could not release.
Total registrations	The number of name registration messages received from clients.
Last address change replication	Indicates when the last WINS database change was replicated.
Last scavenging times	Indicates the last times the database was cleaned for specific types of entries.
Periodic	Indicates when the database was cleaned based on the renewal interval specified in the **WINS Server Properties** dialog box on the **Name Record** tab.
Admin trigger	Indicates when the database was last cleaned because the administrator chose **Initiate Scavenging**.
Extinction	Indicates when the database was last cleaned based on the extinction interval.
Verification	Indicates when the database was last cleaned based on the verification interval
Unique registrations	Indicates the number of name registration requests accepted by this WINS server.

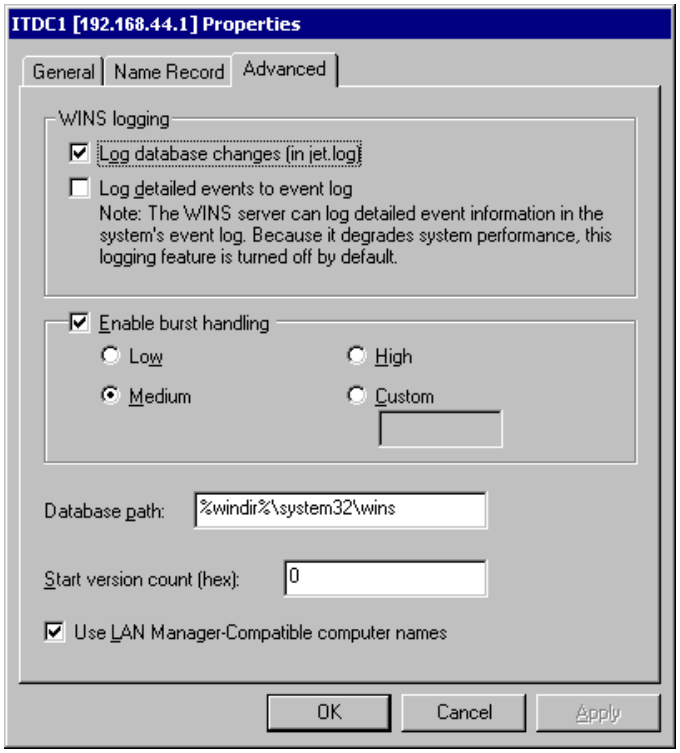
Unique conflicts	The number of conflicts encountered during registration of unique names owned by this WINS server.
Unique renewals	The number of renewals received for unique names.
Group registrations	The number of registration requests for groups that have been accepted by this WINS server.
Group conflicts	The number of conflicts encountered during registration of group names.

Configuring Server and Client Behavior

You can use the configuration options of the WINS management console to change how a WINS server manages its WINS client mappings.

The timer options are found on the **Name Record** tab of the **WINS Server Properties** dialog box shown in Figure 7.8 in the "WINS Replication" section of this chapter. Using these options, you can specify the various timers that govern WINS client behavior: the renewal interval, the extinction interval, the extinction timeout, and the verification interval. All of these are described in "Timers" in this chapter.

In addition, you can change the frequency with which the statistics are updated, and change the backup path for the database, under the **General** tab of the WINS server **Properties** page. Finally, you can change the advanced properties of the server, as shown in Figure 7.19.



If your browser does not support inline frames, click here to view on a separate page.

Figure 7.19 Advanced Configuration Options for WINS Server

To more finely tune a WINS server, configure the options shown in Figure 7.19, which shows the advanced logging and burst handling dialog box. Changing the values of the parameters described in Table 7.18 allows you to alter the most advanced features of your WINS server.

Table 7.18 Advanced WINS Server Configuration Options

Configuration Option	Description
Log database changes	Specifies whether logging of database changes to J50.log files should be turned on.
Log detailed events to event log	Specifies whether events are logged using verbose mode, typically used when troubleshooting. This requires considerable computer resources and should be turned off if you are tuning for performance.
Replicate only with partners	Specifies that replication occur only with configured WINS pull or push partners. If this option is not selected, an administrator can ask a WINS server to pull from or push to an unlisted WINS server partner. By default, this option is selected.
Backup on termination	Automatically backs up the database when the WINS management console stops, except when the computer is shut down.
Migrate	Static unique and multihomed records in the database are treated as dynamic when they conflict with a new registration or replica. If they are no longer valid, they are overwritten by the new registration or replica. Check this option if you are migrating non-Windows NT-based computers to Windows NT. By default, this option is not checked.
Start version count	Specifies the highest version ID number for the database. Usually, you do not need to change this value unless the database becomes corrupted. In this case, set this value to a number higher than the version number counter for this WINS server on all the remote partners that earlier replicated the records to the local WINS server. WINS might increase the value you specify to ensure that database records are quickly replicated to other WINS servers. The maximum allowable value is 2^{31} –1. This value can be seen in the **View Database** dialog box in

	the WINS management console.
Database backup path	Specifies the directory that stores the WINS database backup. If you specify a backup path, WINS automatically performs a full backup of its database to this directory every 24 hours. WINS uses this directory to perform an automatic restoration of the database if the database is found to be corrupted when WINS is started. Do not specify a network directory.

Managing Static Address Mappings

Static mappings are non-dynamic database entries of NetBIOS computer names and IP addresses for computers on the network that are not WINS-enabled or for special groups of network devices.

To view, add, edit, import, or delete static mappings in the WINS management console, on the **Mappings** menu, click **Static Mappings**.

Once entered to the WINS server database, static name–to–IP address mappings cannot be challenged or removed, except by an administrator who removes the specific mapping using the WINS management console. All changes made to the WINS server database with the WINS management console take effect immediately. Note that a DHCP reserved (or a static) IP address for a unique name in a multihomed computer overrides an obsolete WINS static mapping if the WINS server advanced configuration option **Migration On/Off** is checked.

Managing Multihomed Servers

For all computers that use WINS and/or NetBIOS over TCP/IP (NetBT), a single IP address is bound and used. In default configurations, the IP address used to bind NetBT is the primary IP address configured for the first network adapter installed and recognized by Windows 2000.

The order of binding of the adapters can be changed. To do so, open **Network and Dial-up Connections** in **Control Panel**, and then select **Advanced** on the menu bar. Next, select the **Advanced Settings...** command, and then choose the **Adapters and Bindings** tab.

However, you can modify the order of adapter bindings from the **Adapter and Bindings** tab of the **Advanced Settings** screen in the **Network and Dial-up Connections** folder. This dialog screen is located off of the **Advanced** menu. To modify bindings order, use the up and down arrows to re-order the list present in the **Connections list box.**

Because of the reliance of NetBIOS on the first adapter installed and bound in the system, you must verify the IP address of the adapter when using a multihomed WINS server. Once this address is known, assign only this IP address to WINS clients (either dynamically using a DHCP server or by manually configuring clients).

In addition, all WINS push and pull replication partners should be configured through this bound IP address and its physical network adapter. You may need to verify that the bound IP address is also configured at other partner WINS servers.

Administering WINS Through a Firewall

When you administer WINS remotely, an initial session is established to TCP port 135. This is followed by another session to a random TCP port above 1024. These two sessions to specific ports are established because the WINS Administrator uses dynamic endpoints in the remote procedure call (RPC) protocol. Internet firewalls cannot be configured to pass WINS remote administration traffic when the port is not consistent. To solve this problem, in Windows 2000, the default system settings for dynamic port allocation can be changed, in the registry, to a fixed port assignment.

Caution Do not use a registry editor to edit the registry directly unless you have no alternative. The registry editors bypass the standard safeguards provided by administrative tools. These safeguards prevent you from entering conflicting settings or settings that are likely to degrade performance or damage your system. Editing the registry directly can have serious, unexpected consequences that can prevent the system from starting and require that you reinstall Windows 2000. To configure or customize Windows 2000, use the programs in Control Panel or Microsoft Management Console (MMC) whenever possible.

To allow remote administration of WINS through a firewall, you must define a list of all ports available (or not available) from the Internet in the registry in the following entries. These entries are located in the following registry path:

HKEY_LOCAL_MACHINE \SOFTWARE \Microsoft \Rpc \Internet

In particular, the three entries are Ports, PortsInternetAvailable, and UseInternetPorts. Each of these is described in more detail here.

Name: Ports

Data Type: REG_MULTI_SZ — Set of IP port ranges

Description: Specifies a set of IP port ranges consisting of either all of the ports available from the Internet or all of the ports not available from the Internet. Each string represents a single port or an inclusive set of ports (for example, "1000-1050" or "1984"). If any entries are outside the range of zero to 65,535, or if any string cannot be interpreted, the RPC run time will treat the entire configuration as invalid.

Name: PortsInternetAvailable

Data Type: REG_SZ — Y or N (not case sensitive)

Description: If Y, the ports listed in the Ports key are all the Internet-available ports on that computer. If N, the ports listed in the Ports key are all those ports that are not Internet-available.

Name: UseInternetPorts

Data Type: REG_SZ — Y or N (not case sensitive)

Description: Specifies the system default policy. If Y, processes using the default are assigned ports from the set of Internet-available ports, as defined above. If N, processes using the default are assigned ports from the intranet-only ports.

Best Practices for WINS Management Console

The WINS management console provides flexibility and control over your WINS network, but in many cases the default settings are appropriate. Logging and migration are two of the most common trouble spots.

Use Default Configuration Settings The WINS default settings provide the optimal configuration for most conditions and can be used without modification in the majority of WINS network installations. When you modify default settings, be sure that the need to modify is clear and necessary and that you understand the implications.

Do Not Modify the Migrate Setting If you use static WINS entries only to support temporary changes on your network, keep the default **Migrate (Overwrite unique static record with dynamic record)** setting selected in the WINS management console.

When **Migrate (Overwrite unique static record with dynamic record)** is checked, any temporary static entries that are unique or

multihomed can be challenged and dynamically updated by clients. Any later attempt by a WINS client to register a dynamic name that is unique or multihomed over an existing static entry of the same name results in a challenge.

In the challenge, the WINS server compares the IP address in the static mapping to any IP address that the named client attempts to dynamically register in WINS. If the two addresses are different and the static IP address is no longer active, the IP mapping can be changed from static to dynamic and the IP address updated in WINS.

If you use static WINS entries on a permanent basis, you should disable **Migrate (Overwrite unique static entry with dynamic entry).** This prevents a dynamic WINS entry from overriding a static WINS entry that maps to the name and address of a critical server on your network. This is primarily necessary in environments that include many UNIX systems, which do not register with WINS.

Leave WINS Database Logging Enabled When logging is enabled, WINS logs database update activity temporarily to a log file before writing changes back to the server database file. By enabling logging, WINS can process a bulk set of updates that are logged, and then write back the updates to the server database file at periodic intervals. If logging is not enabled, the WINS server database file is written back to disk every time an individual record is changed or updated.

If logging is disabled, registrations are much faster, but this configuration introduces the risk of losing the last few updates to the WINS database when a failure occurs.

Checking the **Log detailed events to event log** box provides a verbose mode with even more detail, and is typically used when troubleshooting. It requires considerable computer resources and should be turned off if you are tuning for performance.

Deploying Microsoft WINS Service

Before you install WINS servers on your network, you must consider the following issues. Each issue is described in more detail later in this section.

Determine the Number of WINS Servers Needed One WINS server can handle NetBIOS name resolution requests for 10,000 computers. However, when deciding how many WINS servers you need, you must consider the location of routers on your network and the distribution of clients in each subnet. For more information, see "How Many Servers to Use" in this section.

Design the WINS Replication Partners Planning WINS replication involves determining whether WINS servers are configured as pull or push partners and setting partner preferences for each server. For more information about how to decide between push or pull replication, see "Configuring WINS Replication" in this section.

Assess the Impact of WINS Traffic on Slow Links Although WINS helps reduce broadcast traffic within and between local subnets, it does create some traffic between servers and clients. Estimate this traffic, particularly on routed TCP/IP networks. In addition to routing traffic, consider the effects of low-speed links (such as those typically used for wide area networking) upon replication traffic between WINS servers and WINS clients registering and renewing NetBIOS names. For more information, see "Network Performance" later in this section.

Assess the WINS Fault Tolerance Within a Network To plan a successful WINS installation, you must consider the effect of a WINS server being shut down or temporarily disconnected from the network. Use additional WINS servers for disaster recovery, backup, and redundancy. For more information on planning a fault-tolerant WINS installation, see "Fault Tolerance" later in this section.

Test and Revise Your Planned WINS Installation By testing the performance of your installation of WINS, you can better identify the source of potential problems before they occur.

WINS Configuration Examples

In the example illustrated in Figure 7.20, a medium-sized company has two main sites (labeled Site 1 and Site 3) with 500 computers each, all connected through relatively high-speed links. The company also has more than 160 small branches. To save on the costs of the links, some branches act as concentrators for a region (such as Site 2).

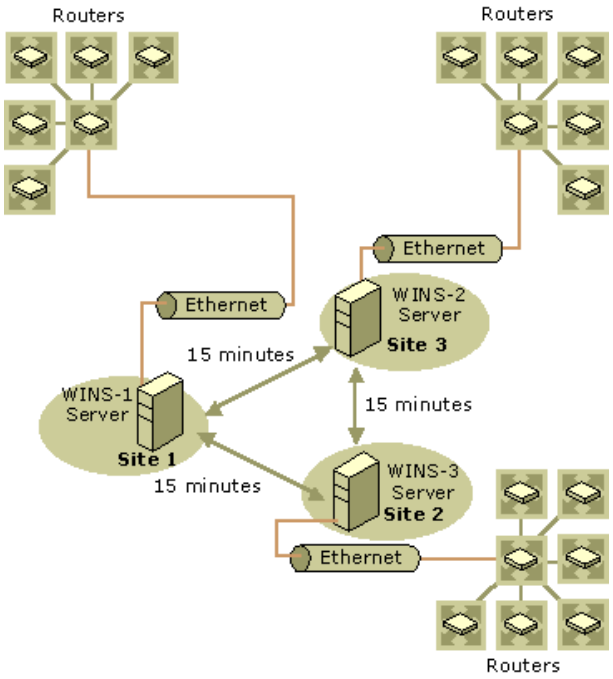


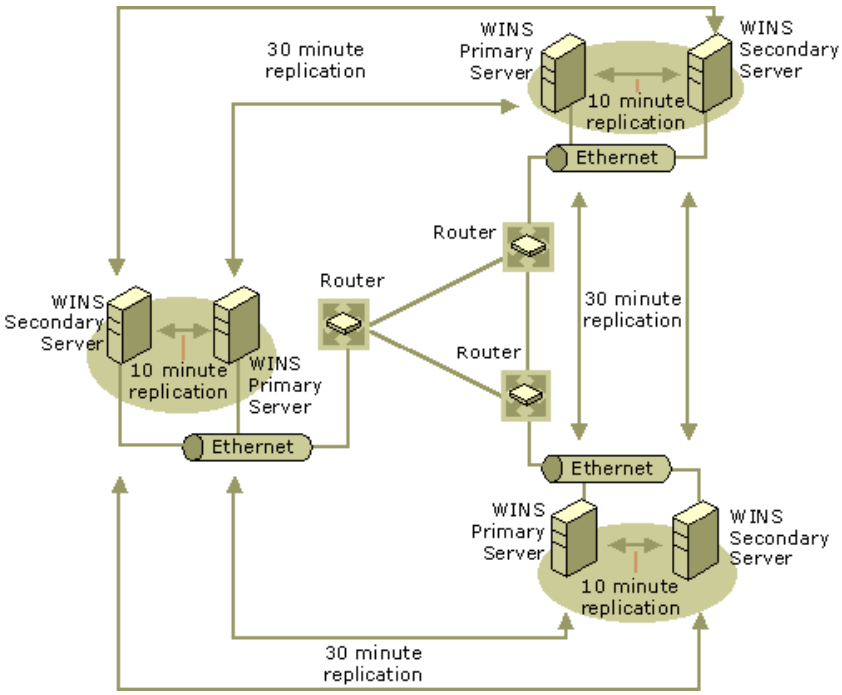
Figure 7.20 A Typical WINS Deployment

The branches might have local WINS servers, but in most cases, they do not—there is simply no need for a separate server for each branch. Instead, the company adds regional WINS servers when the costs of registration and query traffic increase above the cost of deploying the additional server. When the link to a regional WINS server fails, local names can still be resolved by the broadcast mechanism.

The regional WINS servers are not required for this configuration to function correctly, but they do provide a cost optimization. From a network efficiency point of view, the company's system administrators should avoid deploying the regional servers whenever possible because they increase the convergence time. Administrators configure regional WINS servers (such as the one at site 2) as replication partners of the WINS servers in the main sites (sites 1 and 3). Clients in the main site are configured with the IP address of their local WINS server as primary and the IP address of the WINS server in the other main site as secondary. Clients in the regional branches are

configured with the IP address of the regional WINS server as primary and the address of the closest main site WINS server as secondary.

Figure 7.21 shows the network configuration of another example company that is very different. The network serves a larger company with three sites, each with 5,000 users. The sites are connected with multiple T1 links. The number of users justifies a primary and a secondary WINS server at each site. The clients are configured with a local primary and secondary WINS server. Half of the clients have one local WINS server as primary and the other as secondary. The other half have exactly the opposite configuration. This balances the registration and query load over both WINS servers, and it provides a hot backup for maintenance purposes and in case of a calamity.



If your browser does not support inline frames, click here to view on a separate page.

Figure 7.21 A Triangular WINS Deployment

The local WINS servers use a very short pull replication interval of 10 minutes, so all computers within the same building are reachable within 10 minutes of an address registration or change. The replication interval between the sites can be longer—about 30 minutes—because most users work with resources on their local servers.

Assessing Network Traffic

The performance of a WINS system depends on other traffic in the network. When the WINS server is not on the local subnet but is somewhere on the WAN, server requests and responses must go through router queues, causing delays at peak times. Even during replication, bulk transport gets its fair share of the network bandwidth.

The messages described later in this section are UDP/TCP messages. Before they can be sent, the physical address, or *media access control (MAC) address* must be known. If the IP address is already in the ARP cache, the WINS message generates no new ARP message. Otherwise, the client machine will send an ARP packet to resolve the MAC address from the destination IP address, if the destination is on the same local subnet, or from the IP address of the gateway, if the destination is on a remote subnet. Registrations are usually done in groups. Only one ARP message is required per group; this message is not WAN traffic.

All message sizes given in this section are for the messages without a scope ID. The message sizes described here are for Ethernet. On token ring, FDDI, WAN, and so on, the headers (and therefore the total length) vary.

WINS clients generate the following four basic messages:

- Name registration
- Name refresh
- Name release
- Name query

These four messages are discussed in more detail in "WINS Clients" in this chapter.

A Windows 2000–based WINS client usually registers more NetBIOS names than other WINS-enabled clients. The name registration requests generated by a Windows 2000–based computer include the following:

- Workstation component
- Server component
- Messenger service name
- Domain name or names
- Replicator service name
- Browser service name
- Additional network program and service names

When a WINS-enabled client starts on the network, it sends a name registration request for the Workstation service, the Server service, the Messenger service, and any additional Microsoft network services running on the computer. In other words, when a WINS client starts on the network, it generates a minimum of three name registration requests and three entries in the WINS database.

Typical Network Traffic

A name registration request is sent for every NetBIOS name that an application uses. The application makes the request when it (often implemented as a service) starts, which is usually when the computer starts. For a client, the minimum number of names is the two

computer names (one for the workstation component with a last byte of <00> and one for the messenger with a last byte of <03>); the domain name; and the user name (messenger name, last byte <03>).

A server usually has additional names, including server name (the same as the computer name but with a last byte of <20>); more variants of the domain name (<1B> and <1D> for browsing, and <1C>for the domain controllers); a replicator account; a Systems Management Server account; and so forth. The name registration request packet is 110 bytes. A positive name registration response is 104 bytes.

A name release request is sent for a name when its service stops, typically when the system shuts down. The name release request is 110 bytes and the name release response is 104 bytes.

A name refresh request (also called a renewal) is sent regularly while the name is registered. The request is 110 bytes, the response is 104 bytes. The time between renewals depends on the client implementation and the renewal interval. The implementation of WINS in Windows 2000 sends name refresh requests after half the renewal interval to the primary WINS server. When the primary goes down, the renewals are sent at the rate as specified by the renewal interval of the secondary. Only half the renewal attempts are actually done at the secondary; the other attempts are sent to the primary. If the WINS service at the primary WINS server is stopped, then the renewal attempt fails. However, the attempt still generates three packets.

Name query traffic depends on the application and the server. The application might disconnect from the server regularly to release the NetBIOS session. The file server might disconnect idle sessions. Different applications might connect to different servers. This all results in name query traffic; the name query request is 92 bytes, the response is 104 bytes.

Replication and Verification Traffic

Replication and verification traffic is slightly more complicated than typical network traffic because WINS servers perform replication and verification in batches to reduce traffic. In addition, replication and verification sometimes trigger challenge traffic when entries are verified at their owner.

Implementing replication and verification also requires the basic load of connecting and disconnecting with TCP. Each unique name entry requires WINS servers to exchange between 12 and 50 bytes; other types of name entries, such as a group name, which creates a load that depends on the number of clients in the group, might require servers to exchange more data. You can configure the replication interval to reduce the connection overhead, and configuring a persistent connection reduces the overhead to zero.

WINS Client Traffic on Routed Networks

When planning for WINS client traffic on large routed networks, consider the effect of name query, registration, and response traffic routed between subnets. Name requests and responses that occur at the daily startup of computers must pass through the traffic queues on the routers and might cause delays at peak times.

Traffic and Topology

You can estimate WINS client traffic based on the behavior of the WINS clients as described in the preceding sections. However, when estimating WINS client traffic, you must also consider the network topology and the design or configuration of the routers in the network—it might not be possible to predict the traffic load on a specific network router because the routers might be designed or configured to autonomously route traffic based on factors other than traffic load.

How Many Servers To Use

The number of Windows NT–based WINS servers an enterprise requires depends on two factors: the number of WINS clients per server and the network topology. The number of users each server can support depends on usage patterns, data storage, and the processing capabilities of the server. You might need to upgrade your server hardware to handle WINS service.

Clients Per Server

A single WINS server can adequately service up to 10,000 clients for NetBIOS name resolution requests and for WINS service, which is enough for a small network. To provide additional fault tolerance, you should configure a second computer running Windows 2000 and use it as a secondary (or backup) WINS server for clients.

If your network uses only two WINS servers, they should be configured as each other's replication partners. For simple replication between two servers, you should configure one server as a pull partner and the other server as a push partner. You can configure replication manually, or you can set it to be performed automatically by selecting the **Enable Automatic Partner Configuration** check box, which is on the **Advanced** property tab under **Replication Partner** properties.

WINS Server Performance

A WINS server should always be a dedicated device; it should not also be a domain controller, a mail server, or anything else. It should also have a high-performance disk subsystem, such as a RAID array. In general, avoid deploying WINS on domain controllers or on servers that perform other tasks unless absolutely necessary.

A WINS server can typically register 1,500 names per minute or answer 4,500 queries per minute. A conservative recommendation is to install one WINS server and a backup server for every 10,000 computers on the network, which is based on these query response rates. You should plan for the worst cases, such as large-scale power outages that force many computers to restart simultaneously.

Two factors enhance WINS server performance:

1. A dual-processor WINS server increases performance almost 25 percent.
2. A dedicated disk drive measurably improves WINS server name replication response time.

After you establish WINS servers on an intranet, you can also adjust the renewal interval. Setting this interval to reduce the numbers of registrations can improve server response time. You can set the renewal interval when you configure the server, and you can change the interval later in the **WINS Replication Partner** property sheet.

Configuring Replication

Configuring WINS replication correctly is essential to an efficient WINS-capable network. The most important features of a proper WINS configuration are described below.

Automatic Partner Configuration

A WINS server can be configured to automatically accept other WINS servers as its replication partners. When a server uses automatic partner configuration, it finds other WINS servers as they join the network and adds them to its list of replication partners.

Automatic configuration is possible because each WINS server announces its presence on the network through periodic multicast announcements. These announcements are sent as IGMP messages for the multicast group address of 224.0.1.24 (the well-known multicast IP address reserved for use by WINS servers).

When WINS uses automatic replication configuration, it monitors the traffic for these multicast announcements. When it detects a new server, it automatically:

- Adds IP addresses for discovered servers to its list of replication partners.
- Configures any discovered servers to be both push and pull partners.
- Configures pull replication with discovered servers to occur every two hours.

If a remote server is discovered and added as a partner through multicasting, it is removed as a replication partner when WINS is shut down properly. To allow automatic partner information to persist when WINS is restarted, you must use manual partner configuration instead.

To manually configure replication with other WINS servers, configure each partner server using the WINS management console.

Automatic partner configuration is most useful in single-subnet environments. It can also be useful for situations in which the reachable network for WINS multicast traffic is extended by configuring routers between subnets to forward WINS multicast traffic between routed subnets.

Because periodic multicast announcements between WINS servers add traffic to your network, automatic partner configuration is only recommended for use if you have three or fewer installed WINS servers on the reachable network.

Replication Between Untrusted Domains

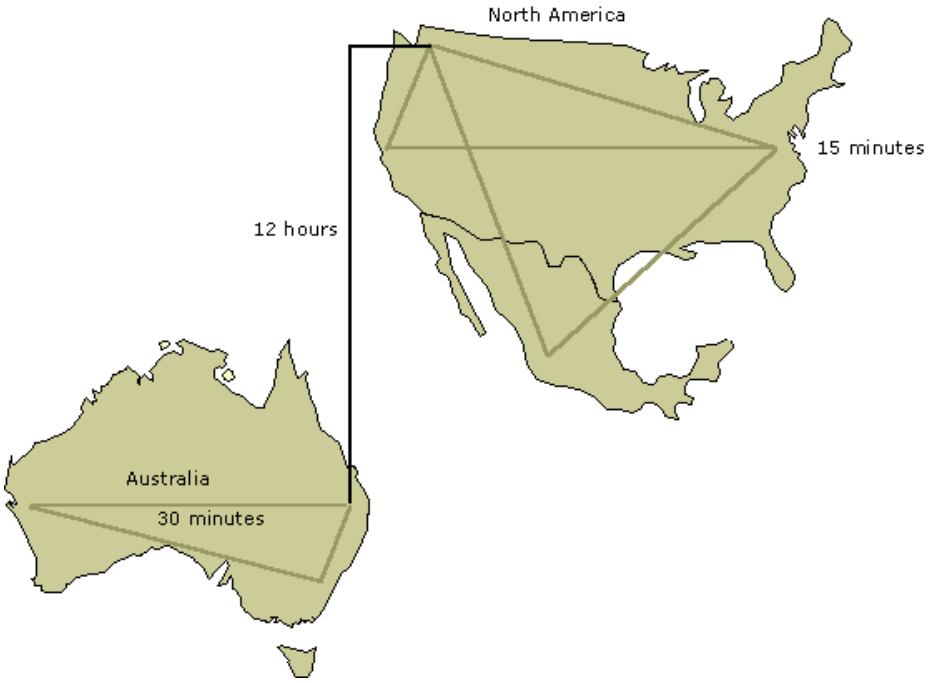
WINS replication can be set up between WINS servers in untrusting domains without requiring a valid user account in the untrusting domain. To configure replication, administrators for each WINS server must use the WINS management console to configure their respective server to allow replication with the WINS server in the remote domain.

Replication Across Wide Area Networks

Selecting the right replication interval requires careful consideration. The WINS server database should be replicated frequently enough that the down time of a single WINS server does not affect the reliability of the mapping information in the database of other WINS servers. However, you do not want the frequency of database replication to interfere with network throughput, which can occur if the replication interval is short.

You also need to consider the topology of your network . If your network has multiple hubs connected by relatively slow WAN links, configure replication between WINS servers on the slow links to occur less frequently than replication between WINS servers on fast links. This reduces traffic across the slow links and reduces contention between replication traffic and client name queries.

Consider an example network in which WINS servers at a central LAN site replicate every 15 minutes, while WINS servers in different WAN hubs replicate every 30 minutes, and WINS servers on different continents replicate just twice each day. Figure 7.22 illustrates this variation in replication frequency.



If your browser does not support inline frames, click here to view on a separate page.

Figure 7.22 Replication over Enterprise Network Configuration

Configurations of other enterprise networks might involve even more zones, each replicating internally on a constant basis with persistent connections, or replicating in short cycles (every 10–30 minutes) to keep the time to convergence short. The servers connecting any two of these zones might replicate daily or hourly.

Replication Convergence Time

When deploying WINS servers, you must choose an acceptable convergence time for your network. Figure 7.23 illustrates a network with WINS servers and the database replication interval between them. This sample network configuration shows how the replication interval between WINS servers affects the convergence time.

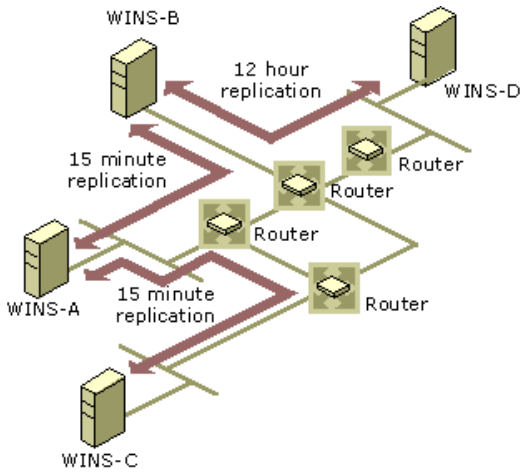


Figure 7.23 Replication Intervals in a Routed TCP/IP Network

If a WINS client registers its name with the WINS server WINS-C, other WINS clients can query WINS-C for this name and get the name–to–IP address mapping. WINS clients that query any of the other WINS servers do not get a positive response until the entry is replicated from WINS-C to WINS servers to WINS-A, WINS-B, and WINS-D.

WINS-C is configured to start replication when the update count exceeds the push threshold or when the pull replication interval expires on its WINS pull partner, WINS-A. The update count is the number of changes to database entries required to trigger push replication. (In this example, WINS-A is configured with a pull replication interval of 15 minutes, rather than an update count threshold.)

In this example, the entry is replicated only when the pull replication interval expires, but queries for the new name to WINS servers WINS-B and WINS-D might still fail. The interval for replication to WINS-B is 15 minutes; to WINS-D, it is 12 hours. Calculate the convergence time as follows:

```
12 hours + ( 2 * 15 minutes ) = 12.5 hours
```

However, name query requests sometimes succeed before the convergence time has passed. For example, this happens when the entries are replicated over a shorter path than the worst-case path. It also happens when an update count threshold is passed before the replication interval expires; this results in earlier replication of the new entry. The longer the replication path, the longer the convergence time.

Example of WINS Server Fault Tolerance

The WINS server database inherently provides fault-tolerant service because it is replicated among multiple WINS servers in a LAN or WAN. This replicated database design prevents users from registering duplicate NetBIOS computer names on the network. In general, even small networks should have more than one WINS server to distribute the load of processing NetBIOS name queries and registration, and to provide WINS database redundancy, backup, and disaster recovery.

WINS server failures come in two basic types:

- **Server failure.** A WINS server might crash, or it might be stopped for maintenance.
- **Network failure.** Routers or link stations might fail.

The failure of an individual WINS server within a network affects multiple WINS servers. An example routed network, shown in Figure 7.23, contains four separate physical segments separated by routers and four WINS servers. Each segment has a single WINS server to provide primary service to local clients on its own segment. Three of the servers (WINS-A, WINS-B, and WINS-C) are linked by routers contained within a single high-speed LAN link topology. The fourth server, WINS-D, is located on a remote segment that uses a low-speed WAN link.

In this example, a failure of WINS-A or WINS-B would segment the distribution of NetBIOS names. Entries would no longer be replicated from WINS-C to WINS-D, and vice versa. Because the IP address and name would no longer match for updated clients, other clients would not be able to connect to the updated computers. Adding replication between WINS-B and WINS-C would improve the configuration for cases in which WINS-A fails. Adding replication between WINS-D and WINS-C would improve fault tolerance in a case where the WINS-B server fails.

Failures of a single link between A, B, and C would not disable the WINS configuration because the underlying router network would reroute the traffic. Although this is not very efficient compared with a fully operational network, WINS replication does continue relatively undisturbed. Failure of the link between WINS-B and WINS-D, however, segments the WINS configuration. Because this makes other network traffic impossible, the network needs an on-demand backup link between WINS-D and WINS-C. This link would allow the underlying router infrastructure to reroute the WINS replication traffic.

In Figure 7.23, the routers are all single points of failure. When one of them fails, it segments the WINS configuration.

Segmented Configurations

When a link or router between two subnets fails, replication between two WINS servers may well be interrupted or prevented by the link failure. However, a segmented WINS configuration can provide many of the services of a fully functional system. Clients can usually resolve addresses from names. Local WINS servers and/or broadcasts resolve most name queries. The only names that cannot be resolved are new entries that were registered remotely, or those that have been updated since the network was segmented. Entries are not dropped at scavenging time when the owning WINS server cannot be reached. To restore the segmented network to full function, install WINS service on another computer when the hardware of the regular WINS server fails, and restore the database by forcing replication from a replication partner.

Improving Fault Tolerance

Windows 2000 and Windows 98 provide an extra measure of fault tolerance by allowing a client to specify more than two WINS servers (up to a maximum of 12) per interface through either the **DHCP** or the **WINS** option under Administrative Tools. The additional WINS servers resolve names only if the primary and secondary WINS servers fail to respond. If one of the additional WINS servers answers a query, the client caches the address of the WINS server that responded and uses it the next time the primary and secondary WINS servers fail to resolve the name. This feature is enabled by default in NetBT. However, if this feature is activated for too many computers, the result is excessive duplication of name queries, resulting in performance degradation.

To plan for fault tolerance, determine the maximum period you ever expect any given WINS server to be out of service. Remember to factor into your assessment the length of both planned and unplanned outages. Also, consider the effect on your WINS clients when their primary WINS server is shut down. By maintaining and assigning secondary WINS servers for clients, the effects of a single WINS

server being offline can often be reduced, if not fully eliminated. In addition, clustering can provide further fault tolerance. For more information, see "WINS Clustering" in this chapter.

Duplicate Replication Traffic

Finely tuning your replication intervals might conserve some bandwidth on WAN links. Improving the example network in Figure 7.23 results in a network like that shown in Figure 7.24. This new configuration is not only more fault tolerant, but it also has a shorter convergence time. In Figure 7.23, the longest path was from WINS-C over WINS-A and WINS-B to WINS-D. Now the longest path is from WINS-A or WINS-C over WINS-B to WINS-D, with a convergence time of 12 hours and 15 minutes.

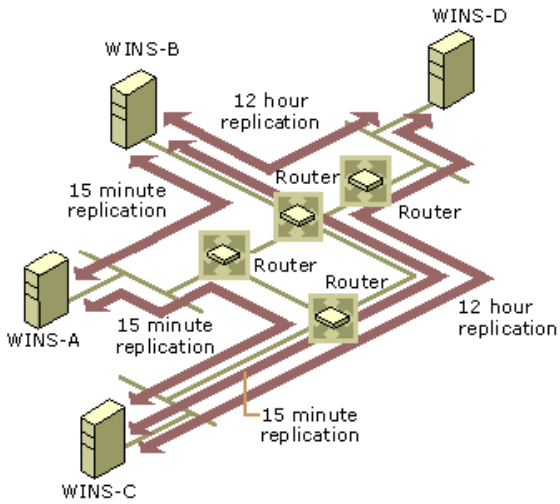


Figure 7.24 An Improved Replication Configuration

By keeping the pull replication intervals between WINS-C and WINS-B short (15 minutes), WINS servers WINS-A, WINS-B, and WINS-C are always reasonably well synchronized. Replicas are never pulled more than once, and only replicas with higher version IDs are copied. When WINS-B pulls an entry directly from WINS-C, it does not pull that replica again from WINS-A.

WINS-D and WINS-B might pull replicas from WINS-C over the link between WINS-B and WINS-C, if WINS-B pulls the replicas from WINS-C, and WINS-D then pulls replicas from WINS-C. This increases the load on the link between WINS-B and WINS-C. To avoid this problem, configure WINS-D to pull from WINS-B first and then check WINS-C. The pull replication interval between servers WINS-D and WINS-C remains 12 hours.

To ensure that the replication is triggered by the pull replication interval and not the update count threshold, you must configure push update counts on WINS servers WINS-D and WINS-C that are high enough to exceed the 12 hours pull replication interval. If these counts are too low, the update count threshold triggers unexpected replication.

Replication Partners and Network Configuration

Choosing whether to configure another WINS server as a push partner or pull partner depends on several considerations, including the specific configuration of servers at your site, whether the partner is across a wide area network (WAN), and how important it is to distribute changes throughout the network immediately.

In the hub-and-spoke configuration, you can configure one WINS server as the central server and all other WINS servers as both push partners and pull partners of this central server. Such a configuration ensures that the WINS database on each server contains addresses for every node on the WAN.

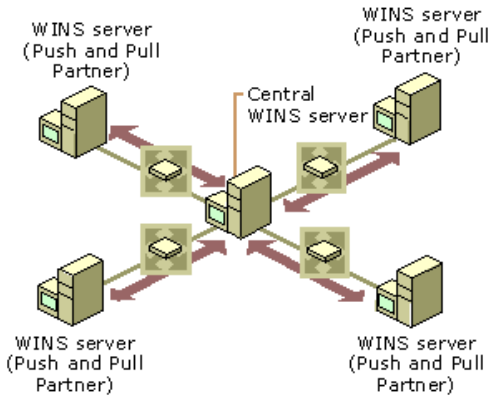


Figure 7.25 Replication using a Central WINS Server

You can select other configurations for replication partner configurations to meet the particular needs of your site. For example, in Figure 7.26, Server1 has only Server2 as a partner, but Server2 has three partners. So Server1 gets all the replicated information from Server2, but Server2 gets information from Server1, Server3, and Server4.

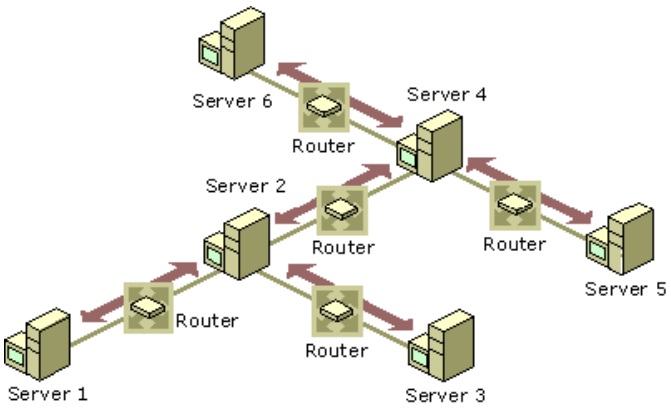


Figure 7.26 Replication in a T Network Configuration

If, for example, Server2 needs to perform pull replications with Server3, make sure it is a push partner of Server3. If Server2 needs to push replications to Server3, it should be a pull partner of Server3.

Decommissioning WINS

For Windows 2000 environments, you might want to reduce or eliminate the need to use WINS on your network. The process of removing installed WINS servers from your network is referred to as decommissioning. For this process to succeed, first consider the following:

- Are any Windows-based computers on your network running earlier versions of Windows or Windows NT?
- Do any client computers in your enterprise still use older Windows-based or MS-DOS-based applications, such as NET command-line utilities, that still require NetBIOS name service?

If the answer to either question is "yes," you still need WINS servers on your network to provide compatibility with older clients and applications. If the answer is "no," you can proceed with designing and implementing a process for removing WINS from your network.

To decommission WINS, you must also implement DNS as your primary naming service for all Windows-based computers active on your network. For more information about implementing DNS on your network, see "Windows 2000 DNS" in this guide. Once DNS is implemented, you can decommission WINS as described in the following sections.

Reconfigure Client Computers' Use of WINS

When you want to remove servers from your network, you must first reconfigure their clients to stop registering and renewing their names with WINS servers. Clients register in WINS on the basis of their being configured for TCP/IP. This reconfiguration is done in one of two ways:

- For clients manually configured to use TCP/IP, remove IP addresses for any WINS servers from the TCP/IP properties for each network connection used by the client computer.
- For clients dynamically configured by DHCP, reconfigure options at your DHCP server (including any options configured by server, scope, or client) not to distribute option code 44 to clients. This option provides a list of WINS server IP addresses to clients.

Verify DNS Configuration

While reconfiguring WINS clients to stop registering of their names and using WINS servers, verify that DNS is fully configured for all clients. Once you are sure that DNS is active for all your clients, you can remove WINS servers from your network.

Clients register in DNS on the basis of their TCP/IP configuration. You can ensure this configuration in two ways:

- For clients that are manually configured to use TCP/IP, add IP addresses for either primary or secondary DNS servers in the TCP/IP properties for each network connection used by the client computer.
- For clients that are dynamically configured by DHCP, reconfigure options at your DHCP server (including any options configured by server, scope, or client) to distribute option code 6 to clients. This option provides to clients a list of IP addresses of DNS servers.

Decommission WINS Servers

Once clients are configured to use DNS rather than WINS for name service, you can decommission individual WINS servers.

Use the WINS management console to mark all records as released for the owner server that you are decommissioning. This tombstone status information is then passed to other WINS servers on your network during the next replication, when they update their local database copies of these records.

Once the records for this server are tombstoned on the other WINS servers on your network, they are automatically removed from other WINS server databases when they have aged to the point of extinction.

To decommission the server, click **Start**, point to **Programs** and then to **Administrative Tools**, and then click **WINS**. If the WINS server you want to decommission does not appear in the console tree, you can add it.

In the console tree, click the WINS server you want to decommission, and then click **Active Registrations**. On the **Action** menu, click **Delete Owner**. In **Delete Owner**, for **Delete this owner**, click the IP address for the WINS server you want to decommission. If the WINS server is not running locally on this computer, it might take a while to load the records for the selected server.

For **Use this operation to delete the selected owner and its records**, click **Replicate deletion to other WINS servers (tombstone)**, and then click **OK**. When prompted to confirm tombstoning, click **Yes**.

In the console tree, click **Replication Partners**. On the **Action** menu, click **Replicate Now**. Once you have verified that records tombstoned in the previous step have been replicated to other partner servers, stop and remove WINS on the decommissioned server.

Important Before decommissioning a WINS server, make sure that any computers previously configured as WINS clients of this server are reconfigured to use other servers as their primary or secondary WINS servers. Reconfiguration is necessary only if these clients will use WINS to register and resolve network names.

Tombstoning ensures that the WINS servers that are replication partners with the decommissioned WINS server are updated properly so that they release the records. If tombstone status is not properly replicated, you can manually delete the records at each WINS server on which tombstone replication failed.

WINS supports remote record deletion for servers running Windows NT Server 4.0, if updated to Service Pack 4 (SP4) or later.

Reducing and Redirecting WINS Traffic

Even after deploying a majority of Windows 2000 computers, most networks must continue using WINS for some time. Once you have begun decommissioning servers, several additional reconfiguration options can reduce both the number of WINS servers on your network and the amount of WINS traffic.

One recommended server-side option is to enable WINS lookup for each of your DNS zones where you are using Microsoft DNS servers. This allows the DNS servers to use WINS to look up names for clients and also to cache frequently requested WINS names. For more information about how to configure WINS lookup for DNS zones, see "WINS Lookup" in this chapter.

As a final step in decommissioning WINS, Windows 2000–based computers allow you to perform client-side configuration changes to disable NetBIOS over TCP/IP. You only need do this where you want to prevent NetBIOS name query and registration traffic from being sent on the network at its source—that is, at each client computer. However, in most networks some limited use of WINS will remain necessary for the foreseeable future. Therefore, disabling NetBIOS over TCP/IP is not recommended for most installations.

Interoperability

The WINS service can share information and functions with DHCP and DNS. The most important interoperability issues are described in this section.

Using DHCP with WINS

When using DHCP and WINS together on your network, consider using additional DHCP scope options to assign WINS node types and to identify WINS primary and secondary servers for DHCP clients. Adjust the options for each physical subnet where DHCP and WINS are implemented on your network.

Assign lease durations of comparable length for both DHCP and WINS. If lease lengths for WINS and DHCP differ widely, the effect on network service is an overall increase in lease management traffic for both services. This is significant only if you do not use the default lease lengths for both services, and lease durations have been changed for either DHCP or WINS individually.

Create DHCP Reservations for Windows 2000 Hosts

Statically mapped Windows 2000–based computers can be problematic when these computers are not periodically stopped and restarted and their initial registration record in WINS becomes damaged. You can have a more reliable and more manageable network by creating DHCP reservations for Windows 2000–based computers. Configure Windows 2000–based domain controllers and domain member servers as DHCP clients with reserved TCP/IP addresses.

You can enter a DHCP reservation at the DHCP server using the media access control address of the network adapter installed in the computer. This reservation ensures that the Windows 2000–based computer gets the same IP address from the DHCP server each time it starts on the network. You can renew WINS registrations for a DHCP client by typing **ipconfig /renew** at a command prompt or by restarting the computer; either procedure corrects the offending WINS registration record.

Configure WINS-Reliant Computers for Fault Tolerance

For fault tolerance in the case of link failure, configure computers that depend on the WINS service located on other subnets as follows. For their primary WINS server, these clients should point to a local WINS server. For their secondary WINS server, these clients should point to the secondary WINS hub. Computers running Windows 95 or Windows NT Workstation send a directed message to the secondary WINS server when the primary WINS server does not contain the requested NetBIOS name. Ideally, this secondary WINS server is located in a separate building and on a separate power grid from the primary WINS server.

Using DNS with WINS

WINS works with the Windows 2000 implementation of Domain Name System (DNS), which is an Internet and TCP/IP networking protocol that provides a scalable and dynamic database service. DNS in Windows 2000 registers and resolves DNS domain names used on private networks and on the Internet. It can provide DNS name service for networked clients, as described in the DNS standard. For more information about DNS, see "Introduction to DNS" and "Windows 2000 DNS" in this book.

In Windows 2000, as with Windows NT 4.0, implementation of DNS is tightly integrated with WINS. This allows non-WINS clients to resolve NetBIOS names by querying a DNS server. Administrators can now remove static entries for Microsoft-based clients in older DNS server zone files in favor of the dynamic integration of WINS and DNS. For example, if a third-party client wants to access a Web page on a WWW server that is enabled for DHCP and WINS, the client can query the DNS server, the DNS server queries WINS, and the name is resolved and returned to the client. Before the integration of WINS and DNS, dynamic IP addressing would have made it impossible to reliably resolve the name in such a situation.

WINS Interoperability Options for DNS

If most of your clients use NetBIOS and you are using Windows 2000 DNS, consider enabling WINS lookup on your DNS servers. When WINS lookup is enabled on DNS servers, WINS resolves any names that DNS resolution does not find. The WINS forward lookup and WINS-R reverse lookup records are supported by Windows 2000 DNS only. If you use third-party DNS servers, use DNS Manager to prevent these WINS records from propagating to the third-party DNS servers that do not support WINS lookup.

If most of your networked computers run Windows 2000, consider upgrading older WINS clients to Windows 2000 and establishing DNS as your only method of name resolution. Support issues involving network name service are simplified if you use a single naming and resource locator service on your network. For more information on moving from an environment combining WINS and DNS to an environment using only Windows 2000 DNS, see "Decommissioning WINS" in this chapter.

Best Practices

Keeping various services working well together is often a matter of dealing with their internal problems rather than their shared elements. This section provides basic practices to help interoperability.

Consolidate Subnets

When you have multiple subnets in a small remote office, consider consolidating the office to one subnet address. You can do this using asynchronous transfer mode (ATM) switching or a virtual private network (VPN) configuration. By consolidating to one subnet address, a local broadcast can be used to resolve names before a request must traverse the WAN to contact a WINS server.

Changing the client to M-node allows it to broadcast locally for resources before contacting a WINS server for NetBIOS name resolution. This can help to reduce the overall amount of WINS-associated traffic, especially WAN traffic.

Update Older Clients

Update client computers running Windows for Workgroups that use the Microsoft TCP/IP-32 protocol stack to the latest Vredir and Vserver files. These files are located on the Windows NT Server 4.0 compact disc (revision 3.11b).

Troubleshooting WINS

This section describes some basic troubleshooting steps for common problems. It also describes how to restore or rebuild the WINS database.

The following conditions can indicate basic problems with WINS:

- Administrator cannot connect to a WINS server using the WINS console. A message appears stating "The RPC server is unavailable."
- TCP/IP NetBIOS Helper service on the WINS client is down and cannot be restarted.
- WINS service is down and cannot be restarted.

First, make sure the appropriate services are running. To do so, complete the following steps at both the WINS server and WINS client:

1. Verify that the WINS services are running.
2. If a necessary service is not started on either computer, start the service.

If services do not start properly, you can use **Computer Management,** available in **Administrative Tools** in Control Panel, to check the status column of the services and try to start them manually. If the service cannot be started, use Event Viewer to check the system event log and determine the cause of failure.

For WINS clients, "Started" should appear in the status column for **TCP/IP NetBIOS Helper Service**. For WINS servers, "Started" should appear in the status column for **Windows Internet Name Service (WINS)**.

Common problems

Following are common WINS problems and steps to solve them.

How can I locate the cause of "duplicate name" error messages?

Check the WINS database for the name. If you find a static record, remove it from the database of the primary WINS server for the client where the duplicate name was detected.

Alternatively, select the **Migrate (Overwrite unique static record with dynamic record)** check box in **Replication Partners Properties** for the WINS server. Now the static mappings in the database can be updated by dynamic registrations (after WINS successfully challenges the old address).

How can I locate the cause of "Network path not found" error messages on a WINS client?

Check the WINS database for the name. If the name is not present in the database, check whether the computer uses B-node name resolution. If so, add a static mapping for it in the WINS database.

If the computer is configured as a P-node, M-node, or H-node, and if its IP address is different from the one in the WINS database, then its address might have changed recently, and the new address has not yet replicated to the local WINS server. To get the latest records, you can start replication at the WINS server that registered the record with the changed address to perform a push replication with propagation to the local WINS server.

Why can't the WINS server pull or push replications to another WINS server?

If the servers are located across routers, confirm that the problem is not a loss of network connectivity or router failure on an intermediate link.

Ensure that each server is correctly configured as either a pull or push partner.

For example, suppose the two WINS servers are named WINS-A and WINS-B. If WINS-A needs to perform pull replications with WINS-B, make sure it is a push partner of WINS-B. Likewise, if WINS-A needs to push replications to WINS-B, it should be a pull partner of WINS-B.

To determine the current configuration of a replication partner, using the WINS console, check the **Type** column for the list of replication partners at each WINS server. If necessary, you can change the replication partner type. Also, make sure that TCP port 42 is not blocked on an intervening network device, such as a router or firewall.

Why are WINS backups failing consistently?

Make sure the path for the WINS backup directory is on a local disk on the WINS server. WINS cannot back up its database files to a remote drive.

Troubleshooting WINS Clients

The most common WINS client problem is failed name resolution. When name resolution fails at a client, answer the following questions to identify the source of the problem.

Was the name that failed to resolve a NetBIOS or DNS name?

NetBIOS names are 15 characters or less and not structured like DNS names, which are generally longer and use periods to delimit each domain level within a name. For example, the short NetBIOS name "PRINT-SRV1" and the longer DNS name "print-srv1.example.microsoft.com" might both refer to the same Windows 2000 resource computer—a network print server—configured to use either name.

In the previous example, if the short name was used at the client, Windows 2000 would first involve NetBIOS name services, such as WINS or NetBT broadcasts, in its initial attempts to resolve the name. If a longer DNS name (or a name that uses dots) was involved in the failure, DNS is more likely the cause of the failed name resolution.

Is the client using an application or version of Windows that requires WINS to resolve names?

Not all Windows computers or applications require WINS or NetBIOS over TCP/IP (NetBT). For example, if the failed name resolution was a URL entered in a Web browser or FTP program, or if it was part of an address entered in an Internet e-mail program, a more likely explanation for the problem is a DNS failure.

In pure Windows 2000 environments, DNS can replace WINS as a naming service. For a pure environment to exist, both the client and the resource server (the computer the client has targeted for locating by name) must both be running Windows 2000; Active Directory must be in use as well. For all other cases involving either the client or resource server running an earlier version of Windows or MS-DOS, a mixed environment exists.

In mixed environments, name resolution could fail when any clients need access to shared resources not published via Active Directory, such as older file and print servers, or to complete logon or browsing of Windows NT domains. Some examples of applications that a client might use and need WINS to assist in name resolution include My Network Places, the **Map Network Drive** feature in Windows Explorer, or the **net** command (Net.exe) and any of its supported options, such as **net use** or **net view**.

Is the client computer able to use WINS, and is it correctly configured?

First, check that the client is configured to use both TCP/IP and WINS. Client configuration of WINS-related settings can be done manually by an administrator setting the TCP/IP configuration of the client, or it can be done dynamically by a DHCP server providing the client its TCP/IP configuration.

In most cases, computers running earlier versions of Microsoft operating systems are already able to use WINS once TCP/IP is installed and configured at the client. For Windows 2000, administrators can optionally disable NetBIOS over TCP/IP (NetBT) for each client. If you disable NetBT, WINS cannot be used at the client.

Also, check that the client computer has valid IP addresses. To check the IP configuration of a client computer, use the **ipconfig /all** command. (To slow or pause the output, use **ipconfig /all | more** for screen-by-screen review.)

In the command output, verify that the client computer has a valid IP address, a valid subnet mask, a default gateway, and both a primary and secondary WINS server.

If the client has an invalid configuration, you can either use the **ipconfig /renew** command to force the client to renew its IP configuration with the DHCP server or you can update the TCP/IP configuration for the client manually.

Does the client have basic connectivity with its configured WINS servers?

To verify that a client has basic TCP/IP access to the WINS server, first try pinging the IP address of the WINS server.

For example, if the client uses a primary WINS server at IP address 10.0.0.1, type **ping 10.0.0.1** at the command prompt on the client computer. If you are not sure of the IP address of the WINS server, you can usually learn it typing **ipconfig /all | more** at the command prompt.

If the WINS server responds to a direct ping of the IP address, use the **nbtstat –RR** command at both the client and the resource server that the client seeks to locate by name. This command forces the WINS client services on each computer to send name release and refresh requests to the WINS server and reregister their names.

If the WINS server does not respond to a direct ping, the source of the problem is likely to be a network connectivity problem between the client and the WINS server. Follow basic TCP/IP network troubleshooting steps to fix the problem. For more information, see "TCP/IP Troubleshooting" in this book.

Is the primary or secondary WINS server able to service the client?

At the primary or secondary WINS server for the client, use Event Viewer or the WINS management console to see if WINS is currently running. If WINS is running on the server, search for the name previously requested by the client to see if it is in the WINS server database.

If the name does not appear in the server database, check that replication is configured correctly and is operational between your WINS servers. For more information, see "Troubleshooting WINS Replication" in this chapter.

Troubleshooting WINS Servers

The most common WINS server problem is the inability to resolve names for clients. When a server fails to resolve a name for its clients, the failure most often is discovered by clients in one of two ways:

- The server sends a negative query response back to the client, such as an error message indicating "Name not found."
- The server sends a positive response back to the client, but the information contained in the response is incorrect.

Many WINS problems involve incorrect or missing configuration details. To help prevent the most common types of problems, review WINS best practices for deploying and managing your WINS servers.

Success in fixing WINS problems nearly always follows if you use an orderly approach to troubleshooting. Most WINS-related problems start as failed queries at a client, so it is a good practice to start with examination of the client. For more information, see "Troubleshooting WINS Clients" in this section.

If you determine that a WINS-related problem does not originate at the client, answer the following questions to further troubleshoot the source of the problem at the WINS server of the client.

Is the WINS server able to service the client?

At the primary or secondary WINS server for the client that cannot locate a name, use Event Viewer or the WINS management console to see if WINS is currently running. If WINS is running on the server, search for the name previously requested by the client to see if it is in the WINS server database.

If the WINS server is failing or registering database corruption errors, you can use WINS database recovery techniques to help restore WINS operations. For more information, see the "Troubleshooting WINS databases" section of this chapter.

If the name does not appear in the server database, verify that replication is configured correctly and is operational between your WINS servers. For more information, see "Troubleshooting WINS replication" in this section.

Is the name entry affected by a static mapping issue?

In general, static mappings are not recommended for clients that can use WINS to dynamically update their name and address information. If the information returned to a client during name resolution is incorrect or stale, check to see if the name entry in the WINS servers database is a static entry. If it is, you can update WINS by performing the following steps:

1. In **Replication Partners Properties,** check the **Enable Migrate box** (shown in Figure 7.10).This enables WINS to overwrite static records with dynamic records.
2. Edit the static mapping to update the mapped address information.
3. Delete the static entry from WINS.

Is replication occurring between all WINS servers?

In some WINS deployments, the use of one-way replication partnerships, such as push-only or pull-only partners, can create situations where names are not regularly replicated to all servers in the network. For more information, see "Troubleshooting WINS Replication" in this chapter.

The following error conditions can indicate problems with the WINS server:

- Administrator cannot connect to a WINS server with the WINS management console and receives an error message when attempting to do so.
- WINS client service or Windows Internet Name Service is down and cannot be restarted.

Troubleshooting WINS Replication

Many WINS problems can be corrected by troubleshooting WINS replication when a client and servers are involved in a failed name resolution. In some cases, such as for large networks with complex replication designs and a large number of WINS servers in use, problems with the accuracy or availability of names data are related to timely replication of the WINS database throughout the network.

After you have first investigated common problems related to WINS clients and servers, answering the following questions can help to further troubleshoot the source of the problem in a replicated WINS network.

Is the replication pattern of your network correct and appropriate?

In general, deployment of more than 20 WINS servers is strongly discouraged. Also, for best results and simpler administration, follow

a hub-and-spoke replication topology when designing a replicated WINS network that uses push/pull partnerships between each WINS hub server and its member spoke servers.

If a single hub-and-spoke design exceeds the recommended maximum of 20 WINS servers, you should consult with Microsoft Consulting Services or Microsoft Product Support Services about how to revise or reduce your current WINS installation. For larger or enterprise installations, multiple hub-and-spoke designs are effective solutions.

In rare cases, you might need to use push-only and pull-only partner relationships. You should, however, carefully review added WINS administration issues where these configurations are deployed. At a minimum, establish reliable support procedures for occasions when you might need to manually trigger replication between WINS servers configured to operate using these types of limited replication partnering.

Is the version ID incrementing for WINS entries when replicating on all servers?

The version ID is incremented in the WINS database by each server that owns and registers a name record. The version ID is a hexadecimal value stored with each name record in the database, and WINS uses it for version tracking when a record is replicated to other servers.

Version IDs are incremented only for certain types of record changes. For example, when a name is refreshed, WINS typically does not increment the version ID. For other changes, such as a change in IP address, WINS increments the version ID in most cases.

When the version ID is not consistently incrementing for a name record at all servers in the replicated WINS network, you can use either the WINS management console or command-line options to increase the starting version count for the server and correct the problem.

Server Troubleshooting Utilities

Two utilities are useful when troubleshooting server problems: Hotfix.exe and Srvinfo.exe. Hotfix.exe provides specific information on which current hotfixes are installed on a server. This program is on the Microsoft FTP server and is included with posted hotfixes. Srvinfo.exe gives details on a particular server, such as which services or drivers are present; it can also give disk information for a remote server. This utility is found in the *Windows 2000 Resource Kit*. Information on running the utility can be found in online Help.

Other recommendations when preparing for troubleshooting include the following:

- Ensure server partitions have adequate space for Dumpfile.
- Configure critical servers for use of symbols for debug and dumpfile troubleshooting procedures.

Troubleshooting the WINS Server

The WINS database is essential for name resolution with WINS. When a WINS database server suffers a failure, consult "Restoring a WINS Database" earlier in this chapter.

Resources

This section provides reference material for NetBIOS names, including specifics on all unique and group name suffixes, as well as Netshell commands, RFCs, and other WINS documentation.

NetBIOS Names

Microsoft networking components, such as the Workstation service and Server service, allow the first 15 characters of a NetBIOS name to be specified by the user or administrator, but reserve the 16th character of the NetBIOS name (00–FF hex) to indicate a resource type. Following are some examples of NetBIOS names used by Microsoft components.

NetBIOS Names Reference

A user can specify the first 15 characters of a name in all Microsoft operating systems that support and use NetBIOS names. However, the 16th character of the name (00–FF hex) is always reserved to indicate a resource type.

Tables 7.20 and 7.21 contain additional details of the NetBIOS names used by Microsoft networking components when registering unique and group names.

Table 7.19 NetBIOS Unique Names

Format	
Description	
computer_name[00h]	Registered by the Workstation service on the WINS client. In general, this name is called the NetBIOS computer name.
computer_name[03h]	Registered by the Messenger service on the WINS client. The client uses this service when sending and receiving messages. This name is usually appended to the NetBIOS computer name for the WINS client computer and to the name of the user logged on to that computer when sending messages on the network.
computer_name[06h]	Registered by Routing and Remote Access on the WINS client (when the Routing and Remote Access service is started).
domain_name[1Bh]	Registered by each Windows 2000 Server domain controller running as the domain master browser. This name record is used to allow remote browsing of domains. When a WINS server is queried for this name, a WINS server returns the IP address of the computer that registered this name.
computer_name[1Fh]	Registered by the Network Dynamic Data Exchange (NetDDE) services; appears only if the NetDDE services are started on the computer.
computer_name[20h]	Registered by the Server service on the WINS client. This service is used to provide points of service for the WINS client to share its files on the network.
computer_name[21h]	Registered by the Routing and Remote Access Client service on the WINS client (when the Routing and Remote Access Client is started).
computer_name[BEh]	Registered by the Network Monitoring Agent Service and appears only if the service is started on the WINS client computer. If the computer name has fewer than 15 characters, plus symbols (**+**) are added to expand the name to 15 characters.
computer_name[BFh]	Registered by the Network Monitoring Utility (included with Microsoft® Systems Management Server). If the computer name has fewer than 15 characters, plus symbols (**+**) are added to expand the name to 15 characters.

user_name[03h]	User names for the currently logged-on users are registered in the WINS database. Each user name is registered by the Server service component so that the user can receive any **net send** commands sent to that user name. If more than one user logs on with the same user name, only the first computer logged on with that user name registers the name.

Table 7.20 NetBIOS Group Names

Format	Description
domain_name[00h]	Registered by the Workstation service so that it can receive browser broadcasts from LAN Manager–based computers.
domain_name[1Ch]	Registered for use by the domain controllers within the domain, and can contain up to 25 IP addresses.
domain_name[1Dh]	The name *domain_name*[1Dh] is registered for use by a master browser; there is only one master browser per subnet. Backup browsers use this name to communicate with the master browser to retrieve the list of available servers from the master browser. WINS servers always return a positive registration response for *domain_name*[1D], even though the WINS server does not register this name in its database. Therefore, when a WINS server is queried for *domain_name* [1D], the server always responds with a broadcast address, which forces the client to broadcast to resolve the name.
group_name[1Eh]	A normal group name. Any computers configured to be network browsers can broadcast to this name and listen for broadcasts to this name to elect a master browser. A statically mapped group name uses this name to register itself on the network. When a WINS server receives a name query for a name ending with [1E], the WINS server always returns the network broadcast address for the local network of the requesting client. The client can then use this address to broadcast to the group members. These broadcasts are for the local subnet and should not cross routers.
group_name[20h]	A special group name called the Internet Group is registered with WINS servers to identify groups of computers for administrative purposes. For example, "printersg" could be a registered group name used to identify an administrative group of print servers.
[01h][01h] __MSBROWSE__ [01h][01h]	Registered by the master browser for each subnet. When a WINS server receives a name query for this name, the WINS server always returns the network broadcast address for the local network of the requesting client.

NetShell Commands

The NetShell commands for WINS are an alternative to console-based management, and they are especially useful in certain special situations. They offer a fully equivalent command-line utility for administrating WINS servers.

For instance, when managing WINS servers in wide area networks (WANs), you can use NetShell commands in interactive mode at the NetShell command prompt to better manage WINS servers across slow-speed network links.

You can also issue commands as batch processes to script and automate administrative tasks that must be routinely performed for all WINS servers. This is especially useful when you manage a large number of WINS servers.

Table 7.21 lists commands that you can use at the NetShell command prompt—which is not the same as the Windows 2000 command prompt—to manage WINS servers. Each of these commands has additional notes on switches and usage, which can be obtained by typing the command name followed by **/?** At the command prompt.

Table 7.21 Netshell Commands

Command	Description
list	Lists all the available WINS commands.
dump	Dumps WINS server configuration to command output.
add name	Registers a name to the server.
add partner	Adds a replication partner to the server.
add pngserver	Adds a list of persona non grata servers for the current server.
check database	Checks the consistency of the database.
check name	Checks a list of name records against a set of WINS servers.
check version	Checks the consistency of the version number.
delete name	Deletes a registered name from the server database.
delete partner	Deletes a replication partner from the list of replication partners.
delete records	Deletes or tombstones all or a set of records from the server.
delete owner	Deletes a list of owners and their records.
delete pngserver	Deletes all or selected persona non grata (PNG) servers from the list. Replicas from PNG servers are not accepted during replication.
init backup	Initiates backup of WINS database.
init import	Initiates import from an LMHOSTS file.
init pull	Initiates replication and sends a pull trigger to another WINS server.
init pullrange	Initiates replication and pulls a range of records from another WINS server.
init push	Initiates replication and sends a push trigger to another WINS server.
init replicate	Initiates replication of database with replication partners.
init restore	Initiates restoring of database from a file.

init scavenge	Initiates scavenging of WINS database for the server.
init search	Initiates search on the WINS database for the server.
reset counter	Resets the server statistics.
set autopartnerconfig	Sets the automatic replication partner configuration info for the server.
set backuppath	Sets the backup parameters for the server.
set burstparam	Sets the burst handling parameters for the server.
set logparam	Sets the database and event logging options.
set migrateflag	Sets the migration flag for the server.
set namerecord	Sets registration interval and timeout values for the server, determining the rate at which registration records are renewed, deleted, and verified.
set periodicdbchecking	Sets periodic database checking parameters for the server.
set pullpartnerconfig	Sets the configuration parameters for the specified pull partner.
set pushpartnerconfig	Sets the configuration parameter for the specified push partner.
set pullparam	Sets the default pull parameters for the server.
set pushparam	Sets the default push parameters for the server.
set replicateflag	Sets the replication flag for the server.
set startversion	Sets the start version ID for the database.
show browser	Displays all active domain master browser [1Bh] records.
show database	Displays the database records for the specified server.
show info	Displays configuration information.
show name	Displays the detail information for a particular record in the server.
show partner	Displays all or pull or push partners for the server.
show partnerproperties	Displays default partner configuration.
show pullpartnerconfig	Displays configuration information for a pull partner.
show pushpartnerconfig	Displays configuration information for a push partner.
show reccount	Displays the number of records owned by a specific owner server.
show recbyversion	Displays records owned by a specific server.
show server	Displays the currently selected server.
show statistics	Displays the statistics for the WINS server.
show version	Displays the current version counter value for the WINS server.
show versionmap	Displays the mapping of owner IDs to maximum version numbers.

WINS Specifications (RFCs)

Requests for Comments (RFCs) are an evolving series of reports, proposals for protocols, and protocol standards used by the Internet community. Windows Internet Name Service (WINS) specifications are based on approved RFCs published by the Internet Engineering Task Force (IETF) and other working groups.

The following RFCs contain the core specifications used to design WINS:

RFC 1001: Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods

RFC 1002: Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detailed Specifications

Note RFCs 1001 and 1002 define a standard protocol to support NetBIOS services in a TCP/IP environment. These RFCs describe NetBIOS-over-TCP/IP (NetBT) protocols in a general manner, emphasizing the underlying ideas and techniques used by all NetBT implementations.

WINS complies with these RFCs and provides open, standards-based interoperability as a NetBIOS name service. However, because Microsoft has added significant enhancements beyond the protocol specified in the RFCs, WINS servers are more accurately described as enhanced NetBIOS name servers.
