NetBIOS and NetBEUI

SOME OF THE MAIN TOPICS IN THIS CHAPTER ARE

Before TCP/IP became popular for use in small LANs, the only choice most administrators had was to use a proprietary protocol (such as DECnet or IPX/SPX). When PCs started appearing on the desktop in the early 1980s, the NetBIOS interface and NetBEUI were developed to make networking PCs an easy task. This became the basis of Microsoft's LAN Manager products. An *interface* is not a network transfer protocol. The NetBIOS interface doesn't send or receive data. Instead, NetBEUI provided a means for transmitting data packets on the local network, whereas NetBIOS gave the programmer an application programming interface that could be used to easily access network functions from within applications. With Windows 2000 and the family of Windows 2003 servers, NetBIOS can be used over any of the transport protocols installed on the computer, not just NetBEUI. The underlying transport protocol will be transparent to the applications that make calls to NetBIOS functions. Because the interface stands between the actual network transmission and the application, the programmer does not need to know what transport protocol is used to get data from one place to another. It is simply a matter of making calls to routines in the NetBIOS interface and letting it take care of things from there. Also, because the applications are written to use the NetBIOS interface, you don't have to buy a different application to use for each transport protocol that transports data on the LAN.

Note

This chapter is meant to provide you with historical information about the NetBIOS namespace that was necessary in earlier Windows operating systems. Today you will find that after you adopt Windows 2000 or the Windows Server 2003 family of servers, NetBIOS will be supported only if you have legacy operating systems and applications that depend on this namespace—or the protocol that was developed to work with it, NetBEUI. For all practical purposes, you should consider upgrading your clients to newer operating systems, such as Windows 2000 Professional or Windows XP Professional. Your server operating systems, likewise, are due for an overhaul if you are still using Windows NT.

If you extend the name *NetBIOS* into its original full-length name, you get *Network Basic I/O System*. Just as the BIOS (Basic Input/Output System) you find on your computer motherboard (Basic I/O System) is a collection of software routines that make writing code easier for operating-system and application developers, NetBIOS makes it easier to write applications that require network communications. NetBIOS was created as an extension to the original BIOS, making it easy for programmers to write applications that could make use of network services. By using an API, the programmer doesn't have to get down to the bits and bytes of the underlying hardware or network protocols.

This chapter explains the NetBIOS interface and then looks at the other protocols that were developed to work with or extend NetBIOS. Today NetBIOS support is generally provided only for legacy applications and operating systems—for example, Windows 95/98 clients. If at all possible in your network, you should get rid of these clients and bite the bullet and upgrade to at least Windows 2000 Professional, or Windows XP Professional.

For most major applications, you'll find that newer versions are available that no longer require NetBIOS.

A Brief Historical Look at NetBIOS

Sytec, Inc. originally developed NetBIOS for IBM in 1983. It was designed for use in small departmental LANs of about 20–200 computers and provided peer-to-peer networking capabilities. Peer-to-peer means that there is no central controlling computer or device. Any computer can talk to any other computer connected to the same LAN. It was first employed on IBM's PC Network. At that time, the PC revolution had barely begun, and the capability to network a few hundred computers was quite a task. Even larger networks could be built using gateways or other devices to join these smaller LANs.

In 1985, the *NetBIOS Extended User Interface (NetBEUI)* was released. This allowed more functionality for networking with NetBIOS. NetBEUI, which was used extensively by Microsoft, made a clearer distinction between the network transport protocol function and the programming interface. This was in line with trends in the computer networking industry toward standards, in particular the OSI seven-layer reference model.

Novell released its Advanced NetWare 2.0 in 1986 with a NetBIOS emulator. This emulator enabled programmers to write applications that used NetBIOS calls, yet the underlying transport mechanisms were Novell's own IPX and SPX protocols. Subsequent versions of NetWare have continued to provide a NetBIOS interface, even with NetWare 5, which uses TCP/IP as its core network transport rather than the company's own protocols.

◀◀ The legacy protocols IPX/SPX are discussed in detail in Chapter 32, "Overview of Novell NetWare IPX/SPX."

When IBM developed Token-Ring networks, an emulator for NetBIOS was created. NetBIOS could now be used on both Ethernet and Token-Ring networks. The underlying transport, again, is not really relevant. Providing a common programming interface to top-level applications enabled PC networking to grow faster than it had when each vendor employed proprietary solutions.

Many other implementations of PC networking software have used the NetBIOS interface. Pathworks (Digital Equipment Corporation, now part of Compaq Computer Corporation) was an implementation of a LAN Manager 2.x network running on Digital's DECnet networks. Furthermore, Microsoft has used NetBEUI in LAN Manager since 1987, and it has been integral to networking in Windows for Workgroups and Windows NT to version 4.0. Although Windows 2000 includes support for NetBEUI, it is no longer required and is not the preferred network protocol. To maintain backward compatibility with previous versions of Windows, the newer Windows Server 2003 servers still support a WINS service that can be used to translate between NetBIOS names and IP addresses. The `LMHOSTS` file, which served that purpose in the past for small LANs, is still present and can still be used on these newer servers.

RFC 1001, "Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods," was finished in 1987 and delineates a method of using NetBIOS as an interface with TCP/IP as the network transport. Another, RFC 1002, details the specifics for this proposal, including descriptions of the contents of packets, the format of names, and pseudocode that shows different mechanisms that can be used to resolve names in this environment.

Many other networking products have used NetBIOS and NetBEUI. Because the specifications were not exacting—as they are in other established network protocols, such as TCP/IP—many LAN products were released from various vendors that could not interoperate. Still, until the Internet became a dominant force in both LANs and WANs, NetBIOS and NetBEUI were used by almost every major PC network product.

The important point to gather from this historical perspective is that NetBIOS was the first widely adopted attempt at a networking standard, even though it might not have been as "clean" as the standards that are being developed and used today. The advantages that came from using NetBIOS and NetBEUI helped formulate a desire for network components that could work together instead of the proprietary traps that major computer vendors used before PCs came along.

Note

If you are operating in an all–Windows 2000 environment, you might find that you don't need to use NetBIOS or any of the utilities, such as WINS, that were developed around it. The same goes if you are using one of the Windows Server 2003 servers, which still support WINS. As long as none of your applications requires NetBIOS, then in a true Windows

2000/2003 environment, using DNS for name resolution and the Active Directory for user and computer services information, it is possible to get by without NetBIOS. Note, however, that you should check your applications. For example, the browser service that is usually used to locate computers or services will not function without NetBIOS. If you have older legacy machines, such as Windows 95, Windows 98, or Windows NT on your network, you will have to continue to use NetBIOS. If this is the case, however, you should consider an upgrade for your clients to Windows XP Professional.

NetBIOS Names

Most network protocols require a network address—usually a numeric value—to identify the different computers and processes that run on them. For example, in the TCP/IP protocol suite, IP addresses are used to identify computers and other devices, and ports are used to address specific applications or processes on those computers. NetBIOS, however, is used to establish a logical communication path based on names.

NetBIOS names can be classified into two categories: *unique names* and *groupnames*. A unique name can be used by only one workstation in the local broadcast network, whereas many computers can share groupnames. A NetBIOS name is 16 bytes in length, and if a shorter name is used, it is padded so that it is 16 bytes long. In some implementations (such as in Microsoft products), the 16th byte is given a special meaning. Unlike DNS names, NetBIOS names can be made up of almost any type of character, but cannot start with the asterisk character (*).

Each computer that participates in the network has a unique name that identifies it. A computer can hold more than one unique name. Each service that a computer offers will need to register a name for that service. The most common ones you'll see are for file and print services. For a computer to lay claim to a name, it must usually broadcast that desire to the rest of the network (or use a WINS server) and wait to see whether another computer challenges the name.

The 16th Character

Although NetBIOS names can be 16 characters in length, in many cases (as with Microsoft and IBM products), they are limited to 15 actual characters. The first 15 characters can be anything the administrator wants to use, but the 16th character is used to differentiate between different types of names. This character is often called the NetBIOS suffix, and it qualifies the function of the resource represented by the name. Table 1 is a listing of most of the NetBIOS name types employed by Microsoft. Note that the last character in the name does not have to be in the range of printable ASCII characters. For this reason, the hexadecimal value appears instead. In the Type column, the letter *U* indicates that the name is a unique name, and *G* indicates that the name is a groupname (called Internet groupname in Windows 2000) that can be registered by multiple computers.

Table 1 NetBIOS Names

NetBIOS Name	Type	Suffix	Description
<*computername*>	U	00	Workstation Service
<*computername*>	U	01	Messenger Service
<\\--__MSBROWSE__>	G	01	Master Browser
<*computername*>	U	03	Messenger Service
<*computername*>	U	06	RAS Server Service
<*computername*>	U	1F	NetDDE Service
<*computername*>	U	20	File Server Service
<*computername*>	U	21	RAS Client Service

Table 1 Continued

NetBIOS Name	Type	Suffix	Description
<computername>	U	22	Microsoft Exchange Interchange
<computername>	U	23	Microsoft Exchange Store
<computername>	U	24	Microsoft Exchange Directory
<computername>	U	30	Modem Sharing Server Service
<computername>	U	31	Modem Sharing Client Service
<computername>	U	43	SMS Clients Remote Control
<computername>	U	44	SMS Administrators Remote Control Tool
<computername>	U	45	SMS Clients Remote Chat
<computername>	U	46	SMS Clients Remote Transfer
<computername>	U	4C	DEC Pathworks TCP/IP Service on Windows NT
<computername>	U	42	McAfee Anti-Virus
<computername>	U	52	DEC Pathworks TCP/IP Service on Windows NT
<computername>	U	87	Microsoft Exchange MTA
<computername>	U	6A	Microsoft Exchange IMC
<computername>	U	BE	Network Monitor Agent
<computername>	U	BF	Network Monitor Application
<username>	U	03	Messenger Service
<domain>	U	00	Domain Name
<domain>	U	1B	Domain Master Browser
<domain>	U	1C	Domain Controllers
<domain>	U	1D	Master Browser
<domain>	U	1E	Browser Service Elections
<Inet~Services>	U	1C	IIS
<IS~computername>	U	00	IIS
<computername>	U	2B	Lotus Notes Server Service
IRISMULTICAST	G	2F	Lotus Notes
Irisnameserver	g	23	Lotus Notes
Forte_$ND800ZA	U	20	DCA IrmaLan Gateway Server Service

This is not an exhaustive listing of all possible types of NetBIOS names that can exist. Other applications, such as Lotus Notes, can also register NetBIOS names.

The NetBIOS Scope ID

When you use NetBIOS over TCP/IP, the NetBIOS Scope identifier is used. This identifier is composed of a string of characters that conform to the rules used to construct DNS names. By using a scope identifier, it is possible to use the same unique NetBIOS name on a network more than once (which is strongly discouraged). The different systems that use the same unique name are differentiated by the scope ID.

For example, the NetBIOS name popeye can be used to uniquely identify a computer on a LAN in the accounting department in the San Francisco office. The same name also can be used to identify a

computer in the New York office. Each is qualified by its scope ID, so the names `popeye.sf.acme.com` will not be confused with `popeye.ny.acme.com`.

However, use the NetBIOS scope ID with reservation because workstations that have a scope ID can communicate only with other nodes that have the same scope ID. The actual definition of a NetBIOS scope in RFC 1001 is "the population of computers across which a registered NetBIOS name is known."

Note that you cannot specify the NetBIOS scope easily. It can be set in the options offered by a DHCP server in a Windows NT or Windows 2000 environment, or editing the Registry can do it. Because Microsoft strongly advises against this practice, I will not go into the actual Registry entries here. Instead, consult the online Knowledge Base at `www.supportmicrosoft.com` if you want to pursue this matter further.

Node Types

RFC 1001 defined three end-node types, based on the method used to register and resolve NetBIOS names: b-node, p-node, and m-node. An additional h-node has been formalized and is used in Microsoft Windows networks.

B-Node (Broadcast)

This type of node uses broadcasts on the local network to register and resolve names. This mode of operation has two major drawbacks. First, in anything but a small network, a lot of bandwidth can be taken up by broadcast messages. Second, most routers do not forward broadcast messages by default. Even if you have a router that can be configured to forward broadcast messages, you probably don't want to do so except in a special case in which there is no easier solution. You will end up loading multiple network segments with a lot of broadcast messages.

This type of node might work well in a small network, such as a home network. In such a situation, where the volume of network traffic is quite low, broadcast messages are an insignificant matter. In any other type of networking environment, where having a large number of computers and high network bandwidth is a commodity to be monitored and used wisely, setting up a computer to operate as a b-node is unacceptable.

To register a name, a b-node computer simply broadcasts a datagram on the local segment indicating its desire to use the name. If no other computer challenges the name with a Negative Name Registration Reply, the computer can use the name.

By default, Windows 2000/2003 computers run in b-mode when it comes to NetBIOS. If you configure the computer to use a WINS server, however, it will run as an h-node. You also can edit the Registry to manually configure your Windows 2000/2003 computer to make it run as a p-node or m-node, or by using a DHCP server that has the capability to set the node type.

P-Node (Point-to-Point)

A p-node communicates directly with a *NetBIOS Name Service (NBNS)* to register and resolve names. RFCs 1001 and 1002 describe the functions performed by an NBNS. Microsoft's implementation of the name server is called WINS, for Windows Internet Name Server. WINS operates much like a DNS server, except that it maps NetBIOS names to IP addresses, whereas DNS maps TCP/IP names to IP addresses. WINS also differs from the traditional implementation of DNS in that it is a dynamic database (modern DNS servers include functionality for dynamic name registration). Nodes register unique and groupnames when they boot up by sending directed datagrams to the WINS server. Dynamic name registration techniques for DNS servers is described in quite a few RFCs and is implemented in Windows 2000/2003 DNS servers, but you should check to see whether the DNS server you use has this feature if you want to use it in a Windows 2000 environment.

To register a name, a p-node client will send the name to the WINS server. If another client has already registered the name, the WINS server will send a challenge to that computer. If that computer is still using the name, the WINS server will return a Negative Name Registration Reply to the client trying to register the name. Otherwise, it will send a Positive Name Registration Reply, acknowledging the name registration. Another possibility is that the WINS server will send a Wait Acknowledgment to the client requesting the name registration. In this case, the WINS server is still trying to contact the computer that has previously registered the name and has not yet determined whether it can release this old name registration.

The main advantages that the p-node has over the broadcast method are obvious: No broadcast messages propagate through the network, and by using the IP address or point-to-point communication with a name server, this type of node can talk with a name server on the other side of a router. Thus, it reduces network traffic and scales better for a larger network.

There are drawbacks, however. You must configure each client computer to know the address of the name server because this is the computer it needs to contact both to register its own names and to resolve other names. If the name server is down, the client computers cannot register any new names or resolve names for nodes with which they want to communicate.

Microsoft has addressed both of these issues in Windows NT networks by adopting DHCP so that client nodes can be configured automatically when they boot, and by allowing for multiple WINS servers to replicate data so that if one goes down the others can fill the void.

M-Node (Mixed: Broadcast and Point-to-Point)

The m-node was designed to address the problems inherent to both b-node and p-node methods. A computer configured as an m-node first attempts to use broadcast messages to register or resolve NetBIOS names. If no computer on the local subnet objects to the name registration, the m-node computer then tries to use point-to-point communication with a WINS name server.

The advantage this has over the previous nodes is that if the name server is down, computers within the same broadcast domain can still communicate to resolve names among themselves. Resolving names for computers or resources in other subnets, however, is hampered until the name server returns to service.

H-Node (Hybrid)

The h-node was not included in the RFC 1001 specification, but has been adopted by Microsoft in its Windows operating systems. This node is basically the opposite of the m-node. It first tries point-to-point communication with a name server, and if that fails, it operates as a b-node and attempts name registration and resolution using broadcast methods.

The advantage this node type has over the p-node is the same advantage that the m-node has: It can continue to register and resolve names in the local broadcast domain if communication with the name server fails. The advantage this type has over the m-node, however, is that it limits the use of broadcast messages. As long as the name server is available and can answer the queries the client submits, broadcast messages do not consume network bandwidth.

Another advantage the h-node has over the other three types is that it can be configured to consult the `lmhosts` (LAN Manager Hosts) file. This file is similar to the `hosts` file used by TCP/IP clients to translate TCP/IP hostnames to IP addresses. The `LMHOSTS` file, however, is used to translate NetBIOS names to IP addresses. This modified version also can consult the NetBIOS name cache. When configured in this manner, the node is known as a Microsoft-Modified B-Node. The `lmhosts.sam` example file can be found in the directory `%systemroot%\system32\drivers\etc`, for Windows NT, Windows 2000, and Windows Server 2003 servers. For Windows 95 or Windows 98 clients, the file is located in

the system directory, which by default is `\windows`. To use this file, you should rename it, or copy it to a new file, and use the name `lmhosts` without the `.sam` file extension. In Chapter 30, "Network Name Resolution," you can find examples of how to use this file.

Note

It is important to understand the differences between b-nodes and p-nodes. Because one uses only a broadcast method to register and resolve names and the other uses only a name server, these two node types cannot interact with each other in the name resolution process. There is no common point of reference between the two. It is possible for m-nodes and h-nodes to interact with nodes that use only broadcast methods.

The NetBIOS Namespace

Unlike the TCP/IP namespace, the NetBIOS namespace is flat, with no hierarchical organization. *Flat* means that employing a name is not much different from using the MAC address. There can be only one computer in the network that uses the particular name. Nothing in the MAC address or a NetBIOS name enables the administrator to organize networked computers and resources into a meaningful structure. TCP/IP names represent a hierarchical structure. Computer names are qualified by the domain or subdomain in which they exist. For example, `yoko.ny.acme.com` is immediately identified as a unique computer that is in the subdomain `ny` that is part of the `acme.com` domain. The same Acme company can have many locations and can address computers or resources in each location by the subdomain in which they reside, making use of the same computer name, as long as they reside in different DNS domains.

The use of scope identifiers seems to overcome this problem with NetBIOS names. However, this is not part of the original NetBIOS implementation; rather, it uses the organization inherent in TCP/IP and the domain name system.

Representing NetBIOS Names in the Domain Name System

Because NetBIOS names can consist of characters that are not used in the *Domain Name System (DNS)*, a method needs to be used to construct a name that is acceptable to DNS when using NetBIOS with an underlying TCP/IP transport mechanism. RFC 1001 defines such a method, which takes the 16-character name and transforms it into a 32-character name that consists of all uppercase ASCII characters. This process is termed *reversible, half-ASCII,* or *biased encoding*. After the name has undergone this first-level encoding, it is subject to the same compression techniques used by DNS (as described in RFC 883).

To perform first-level encoding, each byte of the NetBIOS name is split into two 4-bit values. Each of these 4-bit values is right-filled with zeros to produce a full byte. The hexadecimal representation of the ASCII value for the uppercase letter *A* is then added to each of these new bytes to produce the final value. This produces uppercase ASCII characters in the range of A–P, all of which are valid characters in a DNS name.

For example:

1. The space character (ASCII value 32 decimal, 20 hexadecimal) is not valid in a DNS name. The binary representation of this ASCII value is "100000".
2. This value is split into 2 half-bytes (sometimes called *nibbles*) of "0010" and "0000".
3. These nibbles are reconstructed into 2 separate bytes by right-filling them with zeros. The resulting bytes are "00000010" and "00000000".

4. Finally, add 41 (hexadecimal) to each of these bytes to get the final byte value for each of these characters:

```
"00000010" + "01000001" = "01000011" or 43 hexadecimal
"00000000" + "01000001" = "00000001" or 41 hexadecimal
```

5. The ASCII character representation of a byte that has a hexadecimal value of 43 is C. The ASCII character represented by the hexadecimal value 41, of course, is A. Thus, the space character of a NetBIOS name is represented as CA when it is transformed into the 32-byte DNS-compatible string.

Before the NetBIOS name can be stored in a DNS database, however, it must have its scope ID appended to it to form a valid DNS fully qualified domain name (FQDN). For example, if the NetBIOS name is `The NetBIOS name`, and the scope ID is `ACME.COM`, the fully qualified DNS name, after encoding, is `FEEIEFCAEOEFFEECEJEPFDCAEOEBENEF.ACME.COM`. It might not make a lot of sense when you look at it, but this encoding method does get around the limitations imposed by DNS name rules, allowing DNS servers to be used to resolve NetBIOS names.

NetBIOS Name Renewal and Release

When a NetBIOS name has been registered with a WINS server, it is granted the right to use the name only for a short time, configurable on the WINS server. The default lease time for names on a WINS server is 6 days, but the administrator can change this (from 1 minute up to 365 days). After a successful name registration, the reply message that the WINS server sends back to the client contains the time value (Time-To-Live, or TTL) that tells the client how long it can use the name.

Renewing and releasing NetBIOS names is handled in the following manner:

- **Renewing**—After half of the lease time has expired, or if the client computer is rebooted, the client will attempt to renew the name registration with the WINS server.
- **Releasing**—Names are released in two ways. First, the client can explicitly release the name. This can be done by using the `nbtstat` command. For example, the command `nbtstat -RR` will cause all names registered by the client to be released and then reregistered. This is used mostly for diagnostic purposes to check and see whether name registration is working properly. Second, if a client fails to renew the name (that is, the client is down or has been moved to a different subnet), the name will eventually timeout or will be challenged by another computer.

NetBIOS Services

NetBIOS provides services to the programmer. The name services have already been discussed. But if all NetBIOS provided was the capability to register and resolve a name on the network, it would not be of much use. Fortunately, NetBIOS (and NetBEUI and other implementations of NetBIOS) provides communications services between NetBIOS names, which can be of three basic types:

- **Datagram**—This is an unreliable connectionless service. A datagram can be sent to a unique name or a groupname. Each datagram is considered independent of others. Because there is no ongoing exchange of data in a logical sequence between computers, it is considered to be "connectionless." Because there is no acknowledgment, the sending computer does not know whether the message is ever received (hence, it is unreliable). The datagram method is the fastest method of sending information by NetBIOS. The size of the message is limited to 512 bytes in most implementations.
- **Broadcast**—Similar to the datagram service, the broadcast service also provides an unreliable connectionless service. The main difference is that the broadcast message can be picked up by all computers in the broadcast domain and is not limited to a specific unique or groupname.

- **Session**—The session service provides a connection-oriented service in full duplex mode. A session ID is used to identify the session, and communications can flow in both directions. Because NetBIOS was intended for small networks, there are no provisions for flow control. Messages using the session service can be up to 64KB in length.

For any of these services to work, both nodes must cooperate. That is, when an application wants to send data, it issues a send or call command. Computers that want to receive these messages must have outstanding receive commands to process any incoming messages.

Locating Network Resources: Browsing the Network

When a Windows client uses the Network Neighborhood icon on the desktop to view the resources available on the network, the display that appears is constructed from a list of servers that make up the *browse list*. You also can use the `net view` command from the MS-DOS command prompt to display this list. Other `net` command options can be used for troubleshooting NetBIOS problems as well.

◀◀ For more information on the `net` command, see Chapter 35, "File Server Protocols."

Microsoft networks use a method called browsing to enumerate (list) resources on the network and make a list available to clients that need to locate these resources. The browser was first created for use in Windows for Workgroups but has continued in other Microsoft products including LAN Manager and the Windows 95, Windows 98, and Windows NT operating systems. This functionality is also included in Windows 2000 and the Windows Server 2003 family of servers, though it is not necessary if your network is composed of only Windows 2000/2003 server computers and your applications can use the Active Directory.

Computers that offer services to the network use NetBIOS names to announce themselves. The computer does not have to be a Windows NT Server computer to offer services. For example, a Windows 98 computer will announce the file and print services it can offer if you allow it to do so when you are configuring its network properties. When the computer boots, it sends a *server announcement* to the master browser or the domain master browser.

The browser system has three main components, as summarized here:

- **Master browsers**—The computer that is the master browser is responsible for compiling the master browse list, which contains a list of servers, workgroups, and domains. The master browser is an elected position, though certain types of computers are more likely to become the master browser than others. If the network contains domains and the domain extends over more than one subnet, the master browser keeps the browse list for the domain members on its subnet.
- **Backup browsers**—Computers that operate as backup browsers poll the master browser every 15 minutes to obtain an updated copy of the browse list. If for any reason a backup browser cannot communicate with the master browser, it forces an *election*, which is the process by which a new master browser is selected.
- **Browse list**—The list of servers that the master or backup browser can maintain is limited to 64KB of data. This limits the number of entries to between 2,000 and 3,000, more or less. If a computer is elected to be the master browser and finds that its browse list is empty, it sends out a request datagram asking servers on the network to send it a server announcement. Computers that receive this request respond during a random interval within 30 seconds. The random delay factor is used to minimize network traffic that occurs if a large number of computers respond at the same time.

The list of servers that is maintained in the browse list does not mean just Windows NT Servers. A server is any node that provides a network and uses NetBIOS names for sharing them.

Client Computers

A client computer first contacts the master browser for a list of backup browsers when an application first makes a `NetServerEnum` API call. It uses the `QueryBrowserServers` directed datagram to the NetBIOS name `<domain name>\0x1d` to do this.

When the master browser on the client's subnet detects this datagram, it sends the client a list of browsers for the workgroup or domain that the client requests. The client selects three names from this list, randomly selects one of these browsers, and sends a request to it for the browse list.

The Domain Master Browser

A special type of master browser is the domain master browser. This browser service runs *only* on the domain's primary domain controller. When the domain spans more than one subnet, each master browser that is responsible for a subnet portion of the domain announces itself to the domain master browser. The domain master browser obtains a list of servers from each master browser and compiles the domain browser list. This list is updated every 15 minutes. This domainwide browse list is then solicited by each master browser so that its clients can browse the entire domain.

If the network is composed of only Windows for Workgroups computers and no domain mechanism is in place, each subnet functions as a separate browsing entity. It makes no difference if you use the same workgroup name on each subnet. It is the domain master browser that provides the capability of maintaining a browse list that extends across subnets.

When Servers or Browsers Fail

When a node that provides a service or a computer that is operating as a browser fails, the names of the servers or services are not immediately removed from the browse list. This is because the updates that are made to the list are not done in real-time, but rather at regular intervals.

Backup browsers and computers that provide a NetBIOS service but that are not browsers announce themselves every 12 minutes on the network. If the master browser does not receive an announcement for three consecutive time periods from a server, it removes the server's name from the browse list. Because backup browsers receive updates from the master browser every 15 minutes, it can take up to 51 minutes for a server to be removed from the list of resources ($3 \times 12 + 15 = 51$).

Because backup browsers expect to receive updates every 15 minutes from the master browser, the failure of a master browser will be noticed more quickly. If any backup browser cannot contact the master browser, it begins the election process. If a client computer detects the failure (that is, in its first attempt to get a list of backup browsers from the master browser), it begins the election process.

When the domain master browser fails, each master browser can only maintain a list for the servers on its subnet. Thus, if the domain master browser is not restored, the domain resources on other subnets are removed from the browse list after a short period.

Browser Elections

Whether a computer can become the master browser depends on several factors. It is an "elected" position. Almost any computer running a Windows-based operating system can become a browser. Unless the browser service is configured not to start, it automatically does so when the computer boots. Some computers serve as master browsers, others as backup browsers. If there is more than one subnet, there is at least one master browser for each subnet, and there is a domain master browser if domains are in use.

The election of a master browser occurs when any of the following happens:

- A computer boots and finds that it cannot locate the master browser.
- A computer that has already booted loses communication with the master browser. A client tries to contact the list of backup browsers it knows about first, and then the master browser. When it fails to find any of them, it forces an election.
- A computer acting as a domain controller comes online. These computers are *preferred* master browsers.

When any of these events occurs, the computer that is involved sends out an *election datagram*. This datagram contains two important pieces of information that are used to evaluate which computer will become the master browser: election version and election criteria.

Election version data consists of a fixed value that is 16 bits in length. It specifies the version of the browser election protocol running on the computer. The election criteria value is a 32-bit value, which is divided into four hexadecimal values of 2 bytes each (as shown in Figure 1).

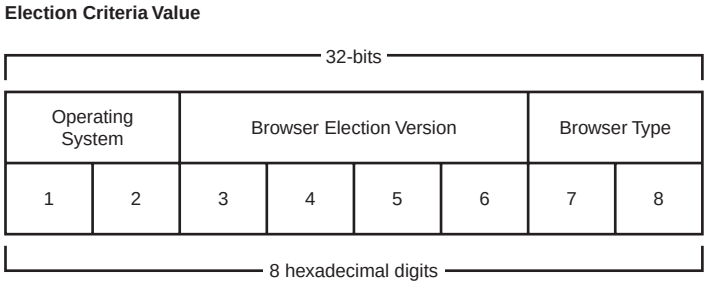

Figure 1 Election criteria for master browser elections.

The first 2 bytes indicate which version of the operating system the computer is running. This value (in hex) is 20 for Windows NT Server and Windows 2000 Server, 10 for Windows NT Workstation and Windows 2000 Professional, and 01 for Windows for Workgroups or Windows 95.

The last 2 bytes are used to obtain further information about the computer's suitability to become the master browser:

- **80**—This computer is a primary domain controller.
- **20**—This computer is a WINS client.
- **08**—This computer is a preferred master browser.
- **04**—This computer is already a running master browser.
- **02**—The Registry for this computer indicates that it has the `MaintainServerList` value or the `BrowseMaster` value set to `Yes`.
- **01**—This computer is a running backup browser.

When a computer receives the election datagram, it compares the datagram to its own values. The first comparison is made of the election version. If the computer has an election version higher than that found in the election datagram, it wins the election at this point and does not bother to process the remaining election criteria.

Otherwise, a comparison of the election criteria is made. If the computer that receives the datagram has a higher value for the election criteria, it joins the election by sending out an election datagram itself. Otherwise, it attempts to determine which computer will become the master browser.

There can be a tie when evaluating the election criteria. The ties are resolved this way:

- The computer that has been running the longest wins, or
- The computer with the lower lexical name wins.

When a computer determines that it has won the election based on the datagram it has evaluated, it enters the running state and broadcasts up to four election datagrams. If it receives no election datagram from any other computer indicating that it should not be the winner, it promotes itself to become the master browser. If it receives an election datagram indicating that another computer will win, it demotes itself to become a backup browser. The delay for sending out the election datagrams is different, depending on the current status of the computer:

- 100 milliseconds for a computer that is a master browser or a primary domain controller.
- 200–600 milliseconds for a computer that is a backup browser or a backup domain controller (value randomly chosen).
- 800–3000 milliseconds for all others.

If a computer that is running in the election receives an election datagram that shows it cannot possibly win the election, it does not continue to send out the remainder of the four election datagrams because doing so would not change the outcome of the election.

The Server Message Block Protocol (SMB) and the Common Internet File System (CIFS)

So far I have discussed the construction of NetBIOS names and how Microsoft clients that use them locate resources on the network. After a client locates a resource, however, the client must be able to use the service the resource provides. The *Server Message Block* protocol *(SMB)* was originally developed at IBM, with later development done at Microsoft. SMB is used to create file and print services, among others, and is an important protocol to understand because it also is used to provide non-Microsoft and non-IBM clients with connectivity to these networks.

For example, SAMBA is a suite of applications that allow SMB clients to use resources that reside on Unix systems. Banyan VINES networking protocols are another example of using SMB to implement network files and print sharing.

Tip

SAMBA is an open-source implementation of the SMB/CIFS protocols that can be used on Unix/Linux and other operating-system platforms. This enables non-Windows systems the capability of locating and sharing files with Windows computers. You can download the current source code, binaries, and documentation for SAMBA at the URL `www.samba.org`.

SMB is a request-response type of protocol. An application formats its message into a data structure called a *Network Control Block (NCB)* and sends the message to the server. SMB messages can be grouped into four basic categories:

- **Session Control**—Messages used to create or delete connections to a network resource.
- **File**—Messages that control accessing file system resources on a network resource.

- **Printer**—Messages used to send files to a print resource and to monitor the status of the print job.
- **Message**—Messages, such as unicast or broadcast messages, used to exchange information between network nodes.

Further development of SMB has resulted in a new protocol called the *Common Internet File System (CIFS)*. This enhanced version of SMB was introduced in Service Pack 3 for Windows NT and is now part of the Windows 2000 operating system, as well as the Windows 2003 family of operating systems. It is an open standard because its operations and functions are fully documented and it can be ported to many different operating system platforms and provide a common means for file sharing on the Internet. Because of the large number of applications that use NetBIOS names and the SMB mechanisms, you might see CIFS in your network in the near future, though it will be transparent to you.

◀◀ An in-depth examination of the SMB protocol and CIFS and how they are used can be found in Chapter 35.

Using *nbtstat* for Troubleshooting

The `nbtstat` command can give you a lot of information and assist you in making adjustments to the NetBIOS environment on a computer. This command has a complex syntax, which is justified by the capabilities it gives you. The syntax for `nbtstat` is shown here:

```
nbtstat [ [-a remotename] [-A IP address] [-c] [-n]
       [-r] [-R] [-RR] [-s] [-S] [interval] ]
```

Following is a rundown of the options in this syntax:

- **`-a remotename`**—Adapter status. This lists the remote machine's name table.
- **`-A IP address`**—Adapter status. This lists the remote machine's name table.
- **`-c`**—This lists the names currently in the NetBIOS name cache on the local computer. These are NetBIOS names of remote machines that have been stored in the local machine's cache.
- **`-n`**—This lists the names registered by the local machine.
- **`-r`**—This option lists names that have been resolved by broadcast or by consulting a WINS server.
- **`-R`**—This option purges and reloads the remote cache name table.
- **`-RR`**—This option sends name release packets to WINS and then starts refreshing them.
- **`-s`**—This option lists the NetBIOS Sessions Connection table, converting destination IP addresses to computer NetBIOS names.
- **`-S`**—This option lists the NetBIOS Sessions Connection table, showing the destination IP addresses.
- ***`remotename`***—A remote host machine name.
- ***`IP address`***—Dotted decimal representation of the IP address.
- ***`Interval`***—The number of seconds that will pass before the display is refreshed. Use Ctrl+C to stop the display.

For example, to view the names in the NetBIOS table on a node named Popeye, you could use the following command:

```
C:\>nbtstat -a popeye

Local Area Connection:
Node IpAddress: [140.176.187.185] Scope Id: []

           NetBIOS Remote Machine Name Table

       Name               Type         Status
    ---------------------------------------------
    POPEYE         <00>  UNIQUE      Registered
    ONO            <00>  GROUP       Registered
    POPEYE         <20>  UNIQUE      Registered
    ONO            <1E>  GROUP       Registered
    POPEYE         <03>  UNIQUE      Registered
    ADMINISTRATOR  <03>  UNIQUE      Registered
    ONO            <1D>  UNIQUE      Registered
    INet~Services  <1C>  GROUP       Registered
    .._MSBROWSE_.<01>  GROUP       Registered
    IS~POPEYE......<00>  UNIQUE      Registered

    MAC Address = 00-08-C7-BA-23-7F
```

You can see from this table that the 16th characters are listed after the name so that you can look them up to determine for what purpose this NetBIOS name is being used. You can also tell from this table whether the name is unique for that computer or is a group name. If you only know the remote IP address of the computer whose table you want to look at, you can use the `-A` option with the address instead of the computer name to get the same information.

To determine whether your local workstation is able to register NetBIOS names, use the `-r` and `-R` options. The `-r` version will give you a listing of the NetBIOS names that the local computer has registered, and it will show them categorized as having been resolved by either the broadcast method (as in a b-node) or by consulting a WINS server. The `-R` option can be used to clear and reload the name cache if you think that it is no longer valid. This can be used for troubleshooting to help determine whether a name can still be registered. You might have more than one computer on the LAN that is trying to register the same unique name, for example.

The `-s` and `-S` options show you current sessions that your computer has open with other computers. The lowercase version will list remote computers by name, whereas the uppercase version will list them by IP address. If you are unable to communicate with a remote computer, check the sessions table to be sure that there are no conflicting entries. Attempt to make the connection (say, to a file or printer service), and then examine the table again to see whether the sessions are showing up.

The `nbtstat` command can be used to solve a lot of problems related to the SMB protocol. Practice using the commands so that you will be familiar with the output. You can do before and after views to check the name cache or the sessions table when you are trying to connect to or view a particular network service. If you don't see the correct names (or addresses) in the cache or in the session table, you can begin to check for connectivity problems to make sure you are indeed able to send and receive datagrams from the machine. You can use a simple TCP/IP tool such as `ping` for that purpose.