

Welcome to Microsoft TCP/IP-32 Help

▼ Expand



Overview Of Microsoft TCP/IP-32 for Windows for Workgroups



Installing and Configuring Microsoft TCP/IP-32



Utilities and Services Reference



More Information On...

Welcome to Microsoft TCP/IP-32 Help

▼ Expand



Overview of Microsoft TCP/IP-32

- [-] Introduction to TCP/IP-32 for Windows for Workgroups
- [-] What Is TCP/IP-32 for Windows for Workgroups?
- [-] What Does Microsoft TCP/IP-32 Include?
- [-] Windows Solutions in TCP/IP Networks
- [-] Networking Concepts for TCP/IP
- [-] Internet Protocol Suite
- [-] IP Addressing
- [-] Dynamic Host Configuration Protocol (DHCP)
- [-] WINS and Name Resolution for Windows Networking
- [-] TCP/IP and Windows Networking



Installing and Configuring Microsoft TCP/IP-32



Utilities and Services Reference



More Information On...

Welcome to Microsoft TCP/IP-32 Help



[Overview Of Microsoft TCP/IP-32 for Windows](#)



[Installing and Configuring Microsoft TCP/IP-32](#)

[Introduction to TCP/IP Installation](#)

[Before Installing Microsoft TCP/IP-32](#)

[Installing TCP/IP](#)

[Configuring TCP/IP](#)

[Configuring TCP/IP to Use DNS](#)








[Configuring Advanced TCP/IP Options](#)



[Utilities and Services Reference](#)

[More Information On...](#)


Welcome to Microsoft TCP/IP-32 Help






























-  [Overview Of Microsoft TCP/IP-32 for Windows](#)
-  [Installing and Configuring Microsoft TCP/IP-32](#)
-  [C: > Utilities and Services Reference](#)
-  [Introduction to TCP/IP Utilities](#)
-  [TCP/IP Utilities](#)
-  [Troubleshooting with TCP/IP Diagnostic Utilities](#)
-  [More Information On...](#)

Welcome to Microsoft TCP/IP-32 Help

- [!\[\]\(3da2b303d29c1ea489bbe26a3f5ac664_img.jpg\) Overview Of Microsoft TCP/IP-32 for Windows](#)
- [!\[\]\(9421cea5a5b5319f79b58962509475ab_img.jpg\) Installing and Configuring Microsoft TCP/IP-32](#)
- [!\[\]\(17cce402a0380c36f25e02ecf91578f5_img.jpg\) Utilities and Services Reference](#)
- [!\[\]\(1086da34995924f924c8e8e23387d139_img.jpg\) More Information On...](#)
- [!\[\]\(ffa6dd4cd8800071ccc1a355540c540c_img.jpg\) Configuration Settings in SYSTEM.INI and PROTOCOL.INI](#)
- [!\[\]\(dfba61b58454dd961d978e324a1fb5e5_img.jpg\) Finding More Information](#)
- [!\[\]\(9580d03b8c5bd7e23dc602a02886460d_img.jpg\) Setting Up LMHOSTS](#)
- [!\[\]\(406c76dc95713637836155a54c3b56d5_img.jpg\) What's New in This Release](#)
- [!\[\]\(b950fe96ed6737d8544db83990032195_img.jpg\) Microsoft TCP/IP-32 Glossary](#)
- [!\[\]\(ec7b82925343491880a39b127070bd34_img.jpg\) Internet Sources for Windows Sockets Applications](#)

Welcome to Microsoft TCP/IP-32 Help

 Collapse

-  [Overview Of Microsoft TCP/IP-32 for Windows](#)
-  [Introduction to TCP/IP-32 for Windows](#)
-  [What Is TCP/IP-32 for Windows?](#)
-  [What Does Microsoft TCP/IP-32 Include?](#)
-  [Windows Solutions in TCP/IP Networks](#)
-  [Networking Concepts for TCP/IP](#)
-  [Internet Protocol Suite](#)
-  [IP Addressing](#)
-  [Dynamic Host Configuration Protocol \(DHCP\)](#)
-  [WINS and Name Resolution for Windows Networking](#)
-  [TCP/IP and Windows Networking](#)
-  [Installing and Configuring Microsoft TCP/IP-32](#)
-  [Introduction to TCP/IP Installation](#)
-  [Before Installing Microsoft TCP/IP-32](#)
-  [Installing TCP/IP](#)
-  [Configuring TCP/IP](#)
-  [Configuring TCP/IP to Use DNS](#)
-  [Configuring Advanced TCP/IP Options](#)
-  [Utilities and Services Reference](#)
-  [Introduction to TCP/IP Utilities](#)
-  [TCP/IP Utilities](#)
-  [Troubleshooting with TCP/IP Diagnostic Utilities](#)
-  [More Information On...](#)
-  [Configuration Settings in SYSTEM.INI and PROTOCOL.INI](#)
-  [Finding More Information](#)
-  [Setting Up LMHOSTS](#)
-  [What's New in This Release](#)
-  [Microsoft TCP/IP-32 Glossary](#)
-  [Internet Sources for Windows Sockets Applications](#)

Introduction to TCP/IP Utilities

The Windows for Workgroups TCP/IP utilities, which are Windows Sockets-based, provide diagnostic and connectivity utilities for connectivity and network administration.

To get help on TCP/IP utilities

- ▶ At the DOSPrompt command line, type a TCP/IP utility name and **-?**. For example, type **ping -?**
Or, see the related command name in this Help file.

Important: The **ftp** and **telnet** utilities all rely on password authentication by the remote computer, and they pass the user's account name and password over the network in cleartext. Because a user equipped with a network analyzer on the same network to steal a user's remote account password, users of these utilities should not use the same password to log onto Windows for Workgroups as the one they use to log onto networks that are not secure. Microsoft Windows, Windows NT, and LAN Manager networking never permit the transmission of Microsoft logon credentials in cleartext.

See Also

[TCP/IP Utilities](#)

[Troubleshooting with TCP/IP Diagnostic Utilities](#)

Introduction to TCP/IP-32 for Windows for Workgroups

Welcome to Microsoft TCP/IP-32 for Windows for Workgroups. Microsoft TCP/IP-32 is a 32-bit implementation of the industry-standard TCP/IP protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP) is a networking protocol that provides communication across interconnected networks. This Help file describes how to install, configure, and troubleshoot Microsoft TCP/IP-32 on a computer running the Microsoft Windows for Workgroups 3.11 operating system.

It is assumed that you are familiar with the Microsoft Windows for Workgroups operating system. If you are not familiar with this operating system, refer to your Microsoft Windows for Workgroups documentation set.

TCP/IP is a networking protocol that provides communication across interconnected networks made up of computers with diverse hardware architectures and various operating systems. TCP/IP can be used with Windows for Workgroups alone, to communicate with devices using other Microsoft networking products, or to communicate with non-Microsoft systems, such as UNIX.

The following topics introduce Microsoft TCP/IP-32 for Windows for Workgroups, describing the elements that make up TCP/IP and presenting an overview of how TCP/IP can be used to provide networking solutions.

- [!\[\]\(750841ae7100dc832cb0a4b3af4492f3_img.jpg\) What is TCP/IP for Windows for Workgroups?](#)
- [!\[\]\(78e449f8a1164b81ecbd00cd97498e27_img.jpg\) What does Microsoft TCP/IP-32 include?](#)
- [!\[\]\(9931ff4a747d4e6edc8cfe9a6d936949_img.jpg\) Windows solutions in TCP/IP networks](#)

What Is TCP/IP-32 for Windows for Workgroups?

The TCP/IP protocol family is a standard set of networking protocols, or rules, that govern how data is passed between computers on a network. TCP/IP is used to connect the Internet, the worldwide internetwork connecting over two million universities, research labs, U.S. defense installations, and corporations. (By convention, "Internet" is capitalized when referring to the worldwide internetwork.) The same protocols can be used in private internetworks that connect several local area networks.

Microsoft TCP/IP-32 for Windows for Workgroups is a 32-bit implementation of the industry-standard TCP/IP protocol that enables enterprise networking and connectivity on Windows for Workgroups-based computers. Adding TCP/IP to a Windows for Workgroups configuration offers the following advantages:

- A standard enterprise networking protocol that is the most complete and accepted protocol available. All modern operating systems offer TCP/IP support, and most large networks rely on TCP/IP for much of their network traffic.
- A technology for connecting to non-Microsoft systems. Many standard connectivity utilities are available to access and transfer data between non-Microsoft systems, including FTP and Terminal Emulation Protocol (Telnet). Several of these standard utilities are included with Windows for Workgroups.
- A robust, scalable, cross-platform client-server framework. Microsoft TCP/IP-32 offers the Windows Sockets interface, which is ideal for developing client-server applications that can run with Windows Sockets-compliant stacks from other vendors and also take advantage of other networking protocols such as Microsoft NWLink. Many public-domain Internet tools also use Windows Sockets.

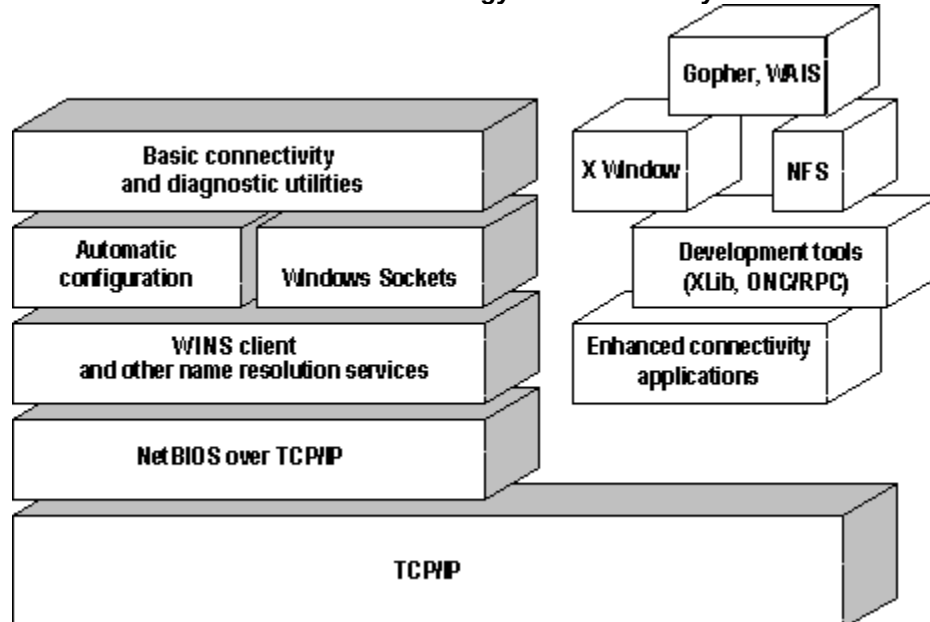
What Does Microsoft TCP/IP-32 Include?

Microsoft TCP/IP-32 provides all the elements necessary to implement these protocols for networking, including:

- Core TCP/IP protocols, including Transmission Control Protocol (TCP), Internet Protocol (IP), User Datagram Protocol (UDP), Address Resolution Protocol (ARP), and Internet Control Message Protocol (ICMP). This suite of Internet protocols provides a set of standards for how computers communicate and how networks are interconnected.
- Support for application interfaces, including Windows Sockets for network programming and NetBIOS for establishing logical names and sessions on the network.
- Basic TCP/IP connectivity applications, including **ftp** and **telnet**. These utilities allow Windows for Workgroups users to interact with and use resources on non-Microsoft hosts, such as UNIX workstations.
- TCP/IP diagnostic tools, including **arp**, **ipconfig**, **nbtstat**, **netstat**, **ping**, **route**, and **tracert**. These utilities can be used to detect and resolve TCP/IP networking problems.

The following diagram shows the basic elements of Microsoft TCP/IP-32 along side the variety of additional applications and connectivity utilities provided by Microsoft and other developers.

Microsoft TCP/IP-32: Core Technology and Third-Party Add-ons:



TCP/IP standards are defined in *Requests for Comments* (RFCs), which are published by the Internet Engineering Task Force (IETF) and other working groups. The key RFCs supported in this version are described in the following list.

<u>RFC</u>	<u>Title</u>
768	User Datagram Protocol (UDP)
791	Internet Protocol (IP)
792	Internet Control Message Protocol (ICMP)
793	Transmission Control Protocol (TCP)
826	Address Resolution Protocol (ARP)
854	Telnet Protocol (TELNET)
894	IP over Ethernet


919, 922	IP Broadcast Datagrams (broadcasting with subnets)
959	File Transfer Protocol (FTP)
1001, 1002	NetBIOS Service Protocols
1034, 1035	Domain Name System (DOMAIN)
1042	IP over Token Ring
1112	Internet Gateway Multicast Protocol (IGMP)
1122, 1123	Host Requirements (communications and applications)
1188	IP over FDDI
1191	Path MTU Discovery
1201	IP over ARCNET
1533	DHCP Options and BOOTP Vendor Extensions
1534	Interoperation Between DHCP and BOOTP
1541	Dynamic Host Configuration Protocol (DHCP)
1542	Clarifications and Extensions for the Bootstrap Protocol

All RFCs can be found on the Internet via ds.internic.net.


Microsoft TCP/IP-32 does not include a complete suite of TCP/IP connectivity utilities, Network File System (NFS) support, or some TCP/IP server services (daemons) such as **routed** and **telnetd**. Many such applications and utilities are available through third-party vendors or are in the public domain.

Windows Solutions in TCP/IP Networks

When TCP/IP is used as a transport protocol, Windows for Workgroups computers can communicate with other kinds of systems without additional networking software. These topics summarize how TCP/IP works with Windows for Workgroups to provide enterprise networking solutions.

 [Using TCP/IP for Scalability in Windows Networks](#)

 [Using TCP/IP for Connectivity to the Internet](#)

 [Using TCP/IP with Third-Party Software](#)

Using TCP/IP for Scalability in Windows Networks

Microsoft TCP/IP-32 provides Windows networking with a set of open transport protocols. TCP/IP delivers a scalable internetworking technology widely supported by hardware and software vendors.

When TCP/IP is used as the enterprise networking protocol, the Windows networking solutions from Microsoft can be used on an existing internetwork. Each of these solutions uses a transport-independent architecture to provide client and server support for TCP/IP and connectivity utilities. These solutions include:



Microsoft Windows for Workgroups version 3.11, with Windows Sockets support. Microsoft TCP/IP-32 for Windows for Workgroups can be used in larger networks to provide access for Windows for Workgroups computers through TCP/IP to Windows NT, LAN Manager, and other TCP/IP systems.



Microsoft Windows NT Workstation 3.5, with enhancements for wide area network (WAN) support, FTP Server service software, extended LMHOSTS support, Windows Sockets support, and DHCP and WINS client software.



Microsoft Windows NT Server 3.5, including the enhancements to Windows NT Workstation 3.5, plus DHCP server and WINS server software to support implementation of these new protocols.



Microsoft LAN Manager, including both client and server, support for Windows Sockets, and MS-DOS-based connectivity utilities.

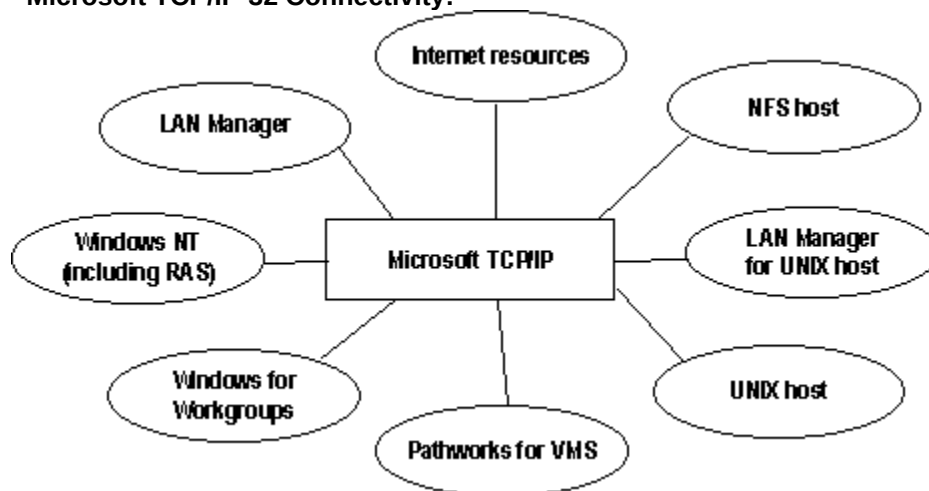
Using TCP/IP for Connectivity to the Internet

Microsoft TCP/IP-32 for Windows for Workgroups includes **ftp** and **telnet** to support file transfer and terminal emulation for communication on the Internet and between foreign systems.





Other TCP/IP applications created by researchers and other users, such as Gopher and NCSA Mosaic, are in the public domain or are available through other vendors as both 16-bit and 32-bit Windows-based applications. Any of these applications that follow the Windows Sockets standard are compatible with Windows for Workgroups. Such applications allow a Windows for Workgroups computer to act as a powerful Internet client while using only basic networking components and public-domain viewers and applications to access Internet resources.

Public-domain Windows-based utilities such as LPR and Gopher can be obtained on the Internet via [ftp.cica.indiana.edu](ftp://cica.indiana.edu/pub/win3/nt) in the /pub/win3/nt or /pub/win3/winsock directory, or via the same directories on [ftp.cdrom.com](ftp://cdrom.com).

Microsoft TCP/IP-32 Connectivity:



Because most modern operating systems (in addition to Windows for Workgroups) support TCP/IP protocols, an internetwork with mixed system types can share information using simple networking applications and utilities. With TCP/IP as a connectivity protocol, Windows for Workgroups can communicate with many foreign systems, including:

-  Internet hosts
-  UNIX® systems
-  Apple® Macintosh® systems
-  Open VMS® systems

Using TCP/IP with Third-Party Software

TCP/IP is a common denominator for heterogeneous networking, and Windows Sockets is a standard used by application developers. Together they provide a framework for cross-platform client-server development. TCP/IP-aware applications from vendors that comply with the Windows Sockets standards can run over virtually any TCP/IP implementation.

The Windows Sockets standard ensures compatibility with Windows-based TCP/IP utilities developed by more than 30 vendors.

This includes third-party applications for the X Window System, sophisticated terminal emulation software, NFS (client and server), electronic mail packages, and more. Because Windows for Workgroups offers compatibility with 16-bit Windows Sockets, applications created for Windows 3.x Windows Sockets will run over Windows for Workgroups without modification or recompilation.

For example, third-party applications for X Window provide strong connectivity solutions by means of X Window servers, database servers, and terminal emulation. With such applications, a Windows for Workgroups computer can work as an X Window server platform while retaining compatibility with applications created for Windows for Workgroups, Windows 3.1, and MS-DOS® on the same system. Other third-party software includes X Window client libraries for Windows for Workgroups, which allow developers to write X Window client applications on Windows for Workgroups that can be run and displayed remotely on X Window server systems.

What's New in This Release

Microsoft TCP/IP capabilities have been expanded to include automatic TCP/IP configuration and powerful name resolution capabilities through the addition of new protocols and supporting administrative tools. New TCP/IP utilities plus the addition of performance counters for TCP/IP and related services will also help make administrative tasks easier.

Enhanced speed and performance

Graphical Telnet and FTP applications

Dynamic Host Configuration Protocol (DHCP) client

Microsoft TCP/IP-32 includes DHCP client software to take advantage of automatic TCP/IP configuration through the new DHCP service. When DHCP servers are installed on the network, users can take advantage of dynamic IP address allocation and management.

Windows Internet Name Service (WINS) client

Microsoft TCP/IP-32 includes the WINS client, to take advantage of provides a new powerful name resolution service for easy, centralized management of computer name-to-IP address resolution in medium and large internetworks where WINS servers are installed.

New TCP/IP utilities and commands

This version includes a new Windows Sockets-based Telnet accessory for connecting to remote systems. The utilities provided with Microsoft TCP/IP-32 have been expanded to include **ipconfig**, a diagnostic tool that displays current TCP/IP network configuration values, and **tracert**, a diagnostic tool for determining the route taken to a destination.

Support for Windows Sockets 1.1

Windows Sockets is the standard networking application programming interface (API) used by programmers to create networking applications.

Finding More Information

For detailed technical information, refer to the following texts and articles:

Allard, J., K. Moore, and D. Treadwell. "Plug into Serious Network Programming with the Windows Sockets API," *Microsoft Systems Journal*, July: 35 - 40, 1993.

Comer, D. *Internetworking with TCP/IP Volume 1: Principles, Protocols, and Architecture*. Second edition. Englewood Cliffs, NJ: Prentice Hall, 1991.

Comer, D. and D. Stevens. *Internetworking with TCP/IP Volume II: Design, Implementation, and Internals*. Englewood Cliffs, NJ: Prentice Hall, 1991.

Comer, D. and D. Stevens. *Internetworking with TCP/IP Volume III: Client-Server Programming and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1991.

Hall, M., et al. *Windows Sockets: An Open Interface for Network Programming Under Microsoft Windows*, Version 1.1, Revision A, 1993.

Krol, E. *The Whole Internet User's Guide and Catalog*. O'Reilly and Associates, 1992.

Glossary

address classes Predefined groupings of Internet addresses, with each class defining networks of a certain size. The range of numbers that can be assigned for the first octet in the IP address is based on the address class. Class A networks (values 1-126) are the largest, with over 16 million hosts per network. Class B networks (128-191) have up to 65,534 hosts per network, and Class C networks (192-223) can have up to 254 hosts per network.

Address Resolution Protocol (ARP) A protocol in the TCP/IP suite that provides IP address-to-media access control (MAC) address resolution for IP packets.

binding A process that establishes the communication channel between a protocol driver and a network adapter driver.

b-node A NetBIOS over TCP/IP mode that uses broadcasts to resolve computer names as addresses.

BOOTP See *Bootstrap Protocol*.

Bootstrap Protocol (BOOTP) An internetworking protocol used to configure systems across internetworks. DHCP is an extension of BOOTP.

Broadcast name resolution A mechanism defined in RFC 1001/1002 that uses broadcasts to resolve names to IP addresses through a process of registration, resolution, and name release.

checksum The mathematical computation used to verify the accuracy of data in TCP/IP packets.

computer name The unique name to which the computer responds. In Windows for Workgroups, the computer name is set by choosing the Network icon in Control Panel, and it is a name of up to 15 uppercase characters that cannot contain spaces. See also *host name*.

daemon A networking program that runs in the background.

datagram A packet of data and other delivery information that is routed through a packet-switched network or transmitted on a local area network.

default gateway The intermediate network device on the local network that has knowledge of the network IDs of the other networks in the internetwork, so it can forward the packets to other gateways until the packet is eventually delivered to a gateway connected to the specified destination. Gateways are usually dedicated computers called routers.

DHCP See *Dynamic Host Configuration Protocol*.

DNS See *Domain Name System*.

DNS name servers In the DNS client-server model, the servers containing information about a portion of the DNS database, which makes computer names available to client resolvers querying for name resolution across the internetwork.

domain name space The database structure used by the *Domain Name System* (DNS).

Domain Name System (DNS) Sometimes referred to as the BIND service in BSD UNIX, DNS offers a static, hierarchical name service for TCP/IP hosts. The network administrator configures the DNS with a list of *hostnames* and IP addresses, allowing users of workstations configured to query the DNS to specify remote systems by *hostnames* rather than IP addresses. For example, a workstation configured to use DNS name resolution could use the command **ping remotehost** rather than **ping 11.102.3.4** if the mapping for the system named remotehost was contained in the DNS database. DNS domains should not be confused with LAN Manager or Windows NT networking domains.

Dynamic Host Configuration Protocol A protocol for automatic TCP/IP configuration that provides static and dynamic address allocation and management.

file sharing The ability for a Windows for Workgroups computer to share parts (or all) of its local file system(s) with remote computers. An administrator creates share points by using either File Manager or by using the **net share** command from the command line.

File Transfer Protocol (FTP) A service that supports file transfers between local and remote systems that support this protocol. FTP supports several commands that allow bidirectional transfer of binary and ASCII files between systems. The FTP client is installed with the TCP/IP connectivity utilities.

FQDN See *fully qualified domain name*.

FTP See *File Transfer Protocol*.

fully qualified domain name (FQDN) *Host names* with their *domain names* appended to them. For example, a host with host name **corp001** and DNS domain name **trey.com** has an FQDN of **corp001.trey.com**. (DNS domains should not be confused with Windows NT networking domains.)

gateway Used interchangeably with *IP router* to describe a system connected to multiple physical TCP/IP networks, capable of routing or delivering IP packets between them.

header The data inserted at the beginning of a *packet* that contains control information. For a TCP packet, the header contains the *port ID*, *checksum*, *sequence number*, and other information.

heterogeneous environment An internetwork with servers and workstations from different vendors, using a mix of different operating systems and transport protocols.

h-node A NetBIOS over TCP/IP implementation that uses *p-node* first for name queries, then *bnode* if the name service is unavailable to resolve computer names as addresses.

host Any device that is attached to the internetwork and uses TCP/IP.

host ID The portion of the IP address that identifies a computer within a particular network ID.

host name The name of a device on an internetwork. For a device on a Windows network, this can be the same as the computer name, but it may not be. The host name must be in the host table or be known by a DNS server for that host to be found by another computer attempting to communicate with it.

host table The HOSTS and LMHOSTS files, which contain mappings of known IP addresses mapped to host names.

HOSTS file A local text file in the same format as the 4.3 Berkeley Software Distribution (BSD) UNIX */etc/hosts* file. This file maps *host names* to IP addresses. In Windows for Workgroups, this file is stored in the \WINDOWS\SYSTEM directory.

ICMP See *Internet Control Message Protocol*.

IETF See *Internet Engineering Task Force*.

Internet Control Message Protocol (ICMP) A maintenance protocol in the TCP/IP suite, required in every TCP/IP implementation, that allows two nodes on an IP network to share IP status and error information. ICMP is used by the **ping** utility to determine the readability of a remote system.

Internet Engineering Task Force (IETF) A consortium that introduces procedures for new technology on the Internet. IETF specifications are released in documents called *Requests for Comments* (RFCs).

Internet group names A name known by a DNS server that includes a list of the specific addresses of systems that have registered the name.

Internet Protocol (IP) The messenger protocol of TCP/IP, responsible for addressing and sending TCP packets over the network.

IP See *Internet Protocol*.

IP address Used to identify a node on a network and to specify routing information on an internetwork. Each node on the internetwork must be assigned a unique IP address, which is made up of the *network ID*, plus a unique *host ID* assigned by the network administrator. In Windows for

Workgroups, the IP address can be configured statically on the workstation or configured dynamically through DHCP.

IP router A system connected to multiple physical TCP/IP networks that can route or deliver IP packets between the networks. See also *Gateway*.

LMHOSTS file A local text file that maps IP addresses to the computer names of Windows networking computers outside the local subnet. In Windows for Workgroups, this file is stored in the WINDOWS\SYSTEM directory.

MAC address The address for a device as it is identified at the media access control layer in the network architecture.

m-node A NetBIOS over TCP/IP mode that uses *bnode* first (broadcasts), then *pnode* (name queries) if the broadcast fails to resolve computer names as addresses.

multihomed system A system with multiple network adapters attached to separate physical networks.

name registration The method by which a computer registers its unique name with a name server on the network. name resolution. In a Windows network, a WINS server can provide name registration services.

name resolution The service provided by a DNS name server or a NetBIOS name server to map DNS or NetBIOS computer names to IP addresses. In a Windows network, a WINS server is an NBNS server.

NBNS See *NetBIOS Name Server*.

NDIS See *network driver interface specification*.

NetBIOS Name Server (NBNS) The server implemented under RFC 1001/1002 to provide name resolution services for NetBIOS computer names. WINS is an NBNS.

NetBIOS over TCP/IP The networking module that provides the functionality to support NetBIOS name registration and resolution.

network basic input/output system (NetBIOS) A software interface for network communication.

network driver interface specification (NDIS) In Windows networking, the interface for network adapter drivers. All transport drivers call the NDIS interface to access network adapter cards.

Network File System (NFS) A service for distributed computing systems that provides a distributed file system, eliminating the need for keeping multiple copies of files on separate computers.

network ID The portion of the IP address that identifies a group of computers and devices located on the same logical network.

Network Information Service (NIS) A service for distributed computing systems that provides a distributed database system for common configuration files.

NFS See *Network File System*.

NIS See *Network Information Service*.

packet A transmission unit of fixed maximum size that consists of binary information representing both data and a header containing an ID number, source and destination addresses, and error-control data.

p-node A NetBIOS over TCP/IP implementation that uses point-to-point communications with a name server to resolve computer names as addresses.

port ID The method TCP and UDP use to specify which application running on the system is sending or receiving the data.

protocol A set of rules and conventions by which two computers pass messages across a network.

proxy A computer that listens to name query broadcasts and responds for those names not on the local subnet. The proxy communicates with the name server to resolve names and then caches them for a time period.

Requests for Comments (RFCs) The official documents of the IETF (Internet Engineering Task Force) that specify the details for protocols included in the TCP/IP family.

resolvers DNS clients that query DNS servers for name resolution on networks.

RFC See *Requests for Comments*.

routing The process of forwarding packets to other gateways until the packet is eventually delivered to a gateway connected to the specified destination.

socket A bidirectional pipe for incoming and outgoing data between networked computers. The Windows Sockets API is a networking API used by programmers creating TCP/IP-based sockets applications.

subnet On the Internet, a subnet is any lower network that is part of the logical network identified by the *network ID*.

subnet mask A 32-bit value that allows the recipient of IP packets to distinguish the network ID portion of the IP address from the host ID.

TCP See *Transmission Control Protocol*.

TCP/IP See *Transmission Control Protocol/Internet Protocol*.

TDI See *Transport Driver Interface*.

Transmission Control Protocol (TCP) A connection-based Internet protocol responsible for breaking data into packets, which the IP protocol sends over the network. This protocol provides a reliable, sequenced communication stream for internetwork communication.

Transmission Control Protocol/Internet Protocol (TCP/IP) The Internet protocols used to connect a world-wide internetwork of universities, research laboratories, military installations, organizations, and corporations. TCP/IP includes standards for how computers communicate and conventions for connecting networks and routing traffic.

Transport Driver Interface (TDI) In Windows networking, the common interface for network components that communicate at the Session layer.

User Datagram Protocol (UDP) A TCP complement offering a connectionless datagram service that guarantees neither delivery nor correct sequencing of delivered packets. Optional UDP data checksums validate header and data but do not enforce acknowledgments, leaving this to the application.










Windows Internet Name Service (WINS) A name resolution service that resolves Windows networking computer names to IP addresses in a routed environment. A WINS server handles name registrations, queries, and releases.

WINS See *Windows Internet Name Service*.

TCP/IP Utilities

Select a utility name to get more information

TCP/IP Utilities

 <u>arp</u>	 <u>netstat</u>
 <u>ftp</u>	 <u>ping</u>
 <u>ipconfig</u>	 <u>route</u>
 <u>nbtstat</u>	 <u>telnet</u>
	 <u>tracert</u>

The TCP/IP utilities offer network connections to non-Microsoft hosts such as UNIX workstations. You must have the TCP/IP network protocol installed to use the TCP/IP utilities.

Arp

Displays and modifies the IP-to-Ethernet or token ring address translation tables used by address resolution protocol (ARP). This command is available only if the TCP/IP protocol has been installed.

arp -a [*inet_addr*] [-N [*if_addr*]]

arp -d *inet_addr* [*if_addr*]

arp -s *inet_addr ether_addr* [*if_addr*]

Parameters

inet_addr

Specifies an IP address in dotted decimal notation.

-N [*if_addr*]

Displays the ARP entries for the network interface specified by *if_addr*.

-d

Deletes the entry specified by *in_addr*.

-a

Displays current ARP entries by querying TCP/IP. If *in_addr* is specified, only the physical address for the specified system is displayed.

-s

Adds an entry in the ARP cache associating the IP address *in_addr* with the physical address *ether_addr*. The physical address is given as 6 hexadecimal bytes separated by hyphens. The IP address is specified using dotted decimal notation. The entry is permanent, that is, it will not be automatically removed from the cache after the timeout expires.

ether_addr

Specifies a physical address.

if_addr

Specifies, if present, the IP address of the interface whose address translation table should be modified. If not present, the first applicable interface will be used.

Ftp

Transfers files to and from a computer running an FTP server service (sometimes called a daemon). **Ftp** can be used interactively or with ASCII text files.

To run ftp



In the Microsoft TCP/IP program group, double-click the FTP icon, then use the FTP command prompt to transfer files.

Or, choose Run from the File menu in Program Manager, and then type **ftp** plus any command-line switches and press ENTER. For example, **ftp -s:myfile.scr**

Syntax

```
ftp [-v] [-n] [-i] [-d] [-g] [host] [-s: filename]
```

Parameters

-v

Suppresses display of remote server responses.

-n

Suppresses autologon upon initial connection.

-i

Turns off interactive prompting during multiple file transfers.

-d

Enables debugging, displaying all **ftp** commands passed between the client and server.

-g

Disables filename globbing, which permits the use of wildcard characters in local file and path names. (See the **glob** command.)

host

Specifies the host name or IP address of the remote host to connect to.

-s: filename

Specifies a text file containing **ftp** commands; the commands will automatically run after **ftp** starts. Use this switch instead of redirection (>).











































More Information About FTP



[Ftp Commands](#)

Ftp Commands

Select an **ftp** command to get more information

- | | | |
|---|---|---|
|  <u>Ftp: !</u> |  <u>Ftp: glob</u> |  <u>Ftp: put</u> |
|  <u>Ftp: ?</u> |  <u>Ftp: hash</u> |  <u>Ftp: pwd</u> |
|  <u>Ftp: append</u> |  <u>Ftp: help</u> |  <u>Ftp: quit</u> |
|  <u>Ftp: ascii</u> |  <u>Ftp: lcd</u> |  <u>Ftp: quote</u> |
|  <u>Ftp: bell</u> |  <u>Ftp: literal</u> |  <u>Ftp: recv</u> |
|  <u>Ftp: binary</u> |  <u>Ftp: ls</u> |  <u>Ftp:</u>
remotehelp |
|  <u>Ftp: bye</u> |  <u>Ftp: mdelete</u> |  <u>Ftp: rename</u> |
|  <u>Ftp: cd</u> |  <u>Ftp: mdir</u> |  <u>Ftp: rmdir</u> |
|  <u>Ftp: close</u> |  <u>Ftp: mget</u> |  <u>Ftp: send</u> |
|  <u>Ftp: debug</u> |  <u>Ftp: mkdir</u> |  <u>Ftp: status</u> |
|  <u>Ftp: delete</u> |  <u>Ftp: mls</u> |  <u>Ftp: trace</u> |
|  <u>Ftp: dir</u> |  <u>Ftp: mput</u> |  <u>Ftp: type</u> |
|  <u>Ftp:</u>
disconnect |  <u>Ftp: open</u> |  <u>Ftp: user</u> |
|  <u>Ftp: get</u> |  <u>Ftp: prompt</u> |  <u>Ftp: verbose</u> |

Ftp: !

Runs the specified command on the local computer.

! command

Parameter

command

Specifies the command to run on the local computer. If *command* is omitted, the local command prompt is displayed; type **exit** to return to **ftp**.

Ftp: ?

Displays descriptions for **ftp** commands. **?** is identical to **help**.

? [*command*]

Parameter

command

Specifies the name of the command about which you want a description. If *command* is not specified, **ftp** displays a list of all commands.

Ftp: append

Appends a local file to a file on the remote computer using the current file type setting.

append *local-file* [*remote-file*]

Parameters

local-file

Specifies the local file to add.

remote-file

Specifies the file on the remote computer to which *local-file* will be added. If *remote-file* is omitted, the local filename is used for the remote filename.

Ftp: ascii

Sets the file transfer type to ASCII, the default.

ascii

Notes

FTP supports two file transfer types, ASCII and binary image. ASCII should be used when transferring text files. See also **binary**.

In ASCII mode, character conversions to and from the network standard character set are performed. For example, end-of-line characters are converted as necessary, based on the target operating system.

Ftp: bell

Toggles a bell to ring after each file transfer command is completed. By default, the bell is off.

bell

Ftp: binary

Sets the file transfer type to binary.

binary

Notes

FTP supports two file transfer types, ASCII and binary image. Binary should be used when transferring executable files. In binary mode, the file is moved byte-by-byte. See also [ascii](#).

Ftp: bye

Ends the FTP session with the remote computer and exits **ftp**.

bye

Ftp: cd

Changes the working directory on the remote computer.

cd *remote-directory*

Parameter

remote-directory

Specifies the directory on the remote computer to change to.

Ftp: close

Ends the FTP session with the remote server and returns to the command interpreter.

close

Ftp: delete

Deletes files on remote computers.

delete *remote-file*

Parameter

remote-file

Specifies the file to delete.

Ftp: debug

Toggles debugging. When debugging is on, each command sent to the remote computer is printed, preceded by the string --->. By default, debugging is off.

debug

Ftp: dir

Displays a list of a remote directory's files and subdirectories.

dir [*remote-directory*] [*local-file*]

Parameters

remote-directory

Specifies the directory for which you want to see a listing. If no directory is specified, the current working directory on the remote computer is used.

local-file

Specifies a local file to store the listing. If not specified, output is displayed on the screen.

Ftp: disconnect

Disconnects from the remote computer, retaining the **ftp** prompt.

disconnect

Ftp: get

Copies a remote file to the local computer using the current file transfer type.

get *remote-file* [*local-file*]

Parameters

remote-file

Specifies the remote file to copy.

local-file

Specifies the name to use on the local computer. If not specified, the file is given the *remote-file* name.

Ftp: glob

Toggles filename globbing. Globbing permits use of wildcard characters in local file or path names. By default, globbing is on.

glob

Ftp: hash

Toggles hash-sign (#) printing for each data block transferred. The size of a data block is 2048 bytes. By default, hash mark printing is off.

hash

Ftp: help

Displays descriptions for **ftp** commands.

help [*command*]

Parameter

command

Specifies the name of the command about which you want a description. If *command* is not specified, **ftp** displays a list of all commands.

Ftp: lcd

Changes the working directory on the local computer. By default, the working directory is the directory in which **ftp** was started.

lcd [*directory*]

Parameter

directory

Specifies the directory on the local computer to change to. If *directory* is not specified, the current working directory on the local computer is displayed.

Ftp: literal

Sends arguments, verbatim, to the remote FTP server. A single FTP reply code is expected in return.

literal *argument* [...]

Parameter

argument

Specifies the argument to send to the FTP server.

Ftp: Is

Displays an abbreviated list of a remote directory's files and subdirectories.

Is [*remote-directory*] [*local-file*]

Parameters

remote-directory

Specifies the directory for which you want to see a listing. If no directory is specified, the current working directory on the remote computer is used.

local-file

Specifies a local file to store the listing. If not specified, output is displayed on the screen.

Ftp: mdelete

Deletes files on remote computers.

mdelete *remote-files* [...]

Parameter

remote-files

Specifies the remote files to delete.

Ftp: mdir

Displays a list of a remote directory's files and subdirectories. **Mdir** allows you to specify multiple files.

mdir *remote-files* [...] *local-file*

Parameters

remote-files

Specifies the directory for which you want to see a listing. *Remote-files* must be specified; type - to use the current working directory on the remote computer.

local-file

Specifies a local file to store the listing. Type - to display the listing on the screen.

Ftp: mget

Copies remote files to the local computer using the current file transfer type.

mget *remote-files* [...]

Parameter

remote-files

Specifies the remote files to copy to the local computer.

Ftp: mkdir

Creates a remote directory.

mkdir *directory*

Parameter

directory

Specifies the name of the new remote directory.

Ftp: mls

Displays an abbreviated list of a remote directory's files and subdirectories.

mls *remote-files* [...] *local-file*

Parameters

remote-files

Specifies the files for which you want to see a listing. *Remote-files* must be specified; type - to use the current working directory on the remote computer.

local-file

Specifies a local file to store the listing. Type - to display the listing on the screen.

Ftp: mput

Copies local files to the remote computer using the current file transfer type.

mput *local-files* [...]

Parameter

local-files

Specifies the local files to copy to the remote computer.

Ftp: open

Connects to the specified FTP server.

open *host* [*port*]

Parameters

host

Specifies the remote computer to connect to. Host can be specified by IP address or host name (a DNS or HOST file must be available). If auto-login is on (default), FTP also attempts to automatically log the user in to the FTP server (see [Ftp](#) to disable auto-login).

port

Specifies a port number to use to contact an FTP server.

Ftp: prompt

Toggles prompting. **Ftp** prompts during multiple file transfers to allow you to selectively retrieve or store files; **mget** and **mput** transfer all files if prompting is turned off. By default, prompting is on.

prompt

Ftp: put

Copies a local file to the remote computer using the current file transfer type.

put *local-file* [*remote-file*]

Parameters

local-file

Specifies the local file to copy.

remote-file

Specifies the name to use on the remote computer. If not specified, the file is given the *local-file* name.

Ftp: pwd

Displays the current directory on the remote computer.

pwd

Ftp: quit

Ends the FTP session with the remote computer and exits **ftp**.
quit

Ftp: quote

Sends arguments, verbatim, to the remote FTP server. A single FTP reply code is expected in return. **Quote** is identical to **literal**.

quote *argument* [...]

Parameter

argument

Specifies the argument to send to the FTP server.

Ftp: recv

Copies a remote file to the local computer using the current file transfer type. **Recv** is identical to **get**.
recv remote-file [local-file]

Parameters

remote-file

Specifies the remote file to copy.

local-file

Specifies the name to use on the local computer. If not specified, the file is given the *remote-file* name.

Ftp: remotehelp

Displays help for remote commands.

remotehelp [*command*]

Parameter

command

Specifies the name of the command about which you want help. If *command* is not specified, **ftp** displays a list of all remote commands.

Ftp: rename

Renames remote files.

rename *filename newfilename*

Parameters

filename

Specifies the file you want to rename.

newfilename

Specifies the new filename.

Ftp: rmdir

Deletes a remote directory.

rmdir *directory*

Parameter

directory

Specifies the name of the remote directory to delete.

Ftp: send

Copies a local file to the remote computer using the current file transfer type. **Send** is identical to **put**.
send *local-file* [*remote-file*]

Parameters

local-file

Specifies the local file to copy.

remote-file

Specifies the name to use on the remote computer. If not specified, the file is given the *local-file* name.

Ftp: status

Displays the current status of FTP connections and toggles.

status

Ftp: trace

Toggles packet tracing; **trace** displays the route of each packet when running an **ftp** command.

trace

Ftp: type

Sets or displays the file transfer type.

type [*type-name*]

Parameter

type-name

Specifies the file transfer type; the default is ASCII. If *type-name* is not specified, the current type is displayed.

Notes

FTP supports two file transfer types, ASCII and binary image.

ASCII should be used when transferring text files. In ASCII mode, character conversions to and from the network standard character set are performed. For example, end-of-line characters are converted as necessary, based on the destination's operating system.

Binary should be used when transferring executable files. In binary mode, the file is moved byte-by-byte.

See Also



[ascii](#)



[binary](#)

Ftp: user

Specifies a user to the remote computer.

user *user-name* [*password*] [*account*]

Parameters

user-name

Specifies a user name with which to log in to the remote computer.

password

Specifies the password for *user-name*. If not specified, but required, **ftp** prompts for the password.

account

Specifies an account with which to log on to the remote computer. If *account* is not specified, but required, **ftp** prompts for the account.

Ftp: verbose

Toggles verbose mode. If on, all **ftp** responses are displayed; when a file transfer completes, statistics regarding the efficiency of the transfer are also displayed. By default, verbose is on.

verbose

Ipconfig

This diagnostic command displays all current TCP/IP network configuration values. This utility is of particular use on systems running DHCP for automatic TCP/IP configuration that do not have user-specified values, but this command can be used on any computer that has Microsoft TCP/IP-32 installed.

With no arguments, the **ipconfig** utility presents all of the current TCP/IP configuration values to the user, including IP address, subnet mask, and default gateway.

ipconfig [/all | renew | /release]

Parameters

all

Produces a full display. Without this switch, **ipconfig** displays only the IP address, subnet mask, and default gateway values for each network card.

renew

Renews DHCP configuration parameters for all network cards on the computer. This option is available only on systems running DHCP.

release

Releases a DHCP configuration for all network cards on the computer. This option disables TCP/IP on the local system, and is available only on systems running DHCP.

Nbtstat

Displays protocol statistics and current TCP/IP connections using NBT (NetBIOS over TCP/IP). This command is available only if the TCP/IP protocol has been installed.

nbtstat [-c] [-n] [-R] [-r] [-S] [-s] [*interval*]

Parameters

-c

Lists the contents of the NetBIOS name cache, giving the IP address of each name.

-n

Lists local NetBIOS names. In this listing, "Registered" indicates that the name has been registered on this network node, either by b-node broadcast or by a WINS server.

-R

Reloads the LMHOSTS file after purging all names from the NetBIOS name cache.

-r

Lists name resolution statistics for Windows networking. On a computer configured to use WINS, this option returns the number of names resolved and registered via broadcast or via WINS.

-S

Displays both workstation and server sessions, listing the remote hosts by IP address only.

-s

Displays both workstation and server sessions. It attempts to convert the remote host IP address to a name using the HOSTS file.

interval

Redisplays selected statistics, pausing *interval* seconds between each display. Press Ctrl+C to stop redisplaying statistics. If this parameter is omitted, **nbtstat** prints the current configuration information once.

More Information About Nbtstat



[Nbtstat--Notes](#)

Nbtstat--Notes

The column headings generated by the **nbtstat** utility have the following meanings.

Input

Number of bytes received.

Output

Number of bytes sent.

In/Out

Whether the connection is from the computer (outbound) or from another system to the local computer (inbound).

Life

The remaining time that a name table cache entry will live before it is purged.

Local Name

The local NetBIOS name associated with the connection.

Remote Host

The name or IP address associated with the remote host.

Type

This refers to the type of name. A name can either be a unique name or a group name.

<03>

Each NetBIOS name is 16 characters long. The last byte often has special significance, because the same name can be present several times on a computer. This notation is simply the last byte converted to hexadecimal. For example, <20> is a space in ASCII.

State

The state of NetBIOS connections. The possible states are shown in the following list:

Accepting = An inbound session is currently being accepted and will be connected shortly.

Associated = A connection endpoint has been created and associated with an IP address.

Connected = The session has been established.

Connecting = The session is in the connecting phase where the name-to-IP address mapping of the destination is being resolved

Disconnected = The local computer has issued a disconnect, and it is waiting for confirmation from the remote system.

Disconnecting = A session is in the process of disconnecting.

Idle = This endpoint has been opened but cannot receive connections.

Inbound = An inbound session is in the connecting phase.

Listening = This endpoint is available for an inbound connection.

Outbound = A session is in the connecting phase where the TCP connection is currently being created.

Reconnecting = A session is trying to reconnect if it failed to connect on the first attempt.

More Information About Nbtstat



[Nbtstat](#)

Netstat

Displays protocol statistics and current TCP/IP network connections. This command is available only if the TCP/IP protocol has been installed.

```
netstat [-a] [-ens] [-p proto] [-r] [interval]
```

Parameters

-a

Displays all connections; server connections are normally not shown.

-e

Displays Ethernet statistics. This may be combined with the **-s** option.

-n

Displays addresses and port numbers in numerical form (rather than attempting name look-ups).

-s

Displays per-protocol statistics. By default, statistics are shown for TCP, UDP, ICMP, and IP; the **-p** option may be used to specify a subset of the default.

-p *proto*

Shows connections for the protocol specified by *proto*; *proto* may be **tcp** or **udp**. If used with the **-s** option to display per-protocol statistics, *proto* may be **tcp**, **udp**, **icmp**, or **ip**.

-r

Displays the contents of the routing table.

interval

Redisplays selected statistics, pausing *interval* seconds between each display. Press **CTRL+C** to stop redisplaying statistics. If this parameter is omitted, **netstat** prints the current configuration information once.

More Information About Netstat



[Netstat--Notes](#)

Netstat--Notes

The **Netstat** utility provides statistics on the following network components.

Proto

The name of the protocol used by the connection.

Recv-Q

The amount of data (in bytes) received but not processed on the connection.

Send-Q

The amount of data (in bytes) waiting to be transmitted (including data sent but not yet acknowledged).

Local Address

The Internet address of the local computer, as well as the port number the connection is using. The name corresponding to the Internet address is shown instead of the number if the HOSTS or LMHOSTS file contains an entry for the Internet address. In cases where the port is not yet established, the port number is shown as an asterisk (*).

Foreign Address

The Internet address and port number of the remote computer to which the socket is connected. The name corresponding to the Internet address is shown instead of the number if the HOSTS or LMHOSTS file contains an entry for the Internet address. In cases where the port is not yet established, the port number is shown as an asterisk (*).

(state)

Indicates the state of TCP connections only. The possible states are

CLOSED	FIN_WAIT_1	SYN_RECEIVED
CLOSE_WAIT	FIN_WAIT_2	SYN_SEND
ESTABLISHED	LISTEN	TIMED_WAIT
LAST_ACK		

More Information About Netstat



[Netstat](#)

Ping

Verifies connections to a remote host or hosts. This command is available only if the TCP/IP protocol has been installed.

```
ping [-t] [-a] [-n count] [-l length] [-f] [-i ttl] [-v tos] [-r count] [-s count]
      [[-j host-list] | [-k host-list]] [-w timeout] destination-list
```

Parameters

-t

Pings the specified host until interrupted.

-a

Resolves addresses to hostnames.

-n count

Sends the number of ECHO packets specified by *count*. The default is 4.

-l length

Sends ECHO packets containing the amount of data specified by *length*. The default is 64 bytes; the maximum is 8192.

-f

Sends a Do Not Fragment flag in the packet. The packet will not be fragmented by gateways on the route.

-i ttl

Sets the Time To Live field to the value specified by *ttl*.

-v tos

Sets the Type Of Service field to the value specified by *tos*.

-r count

Records the route of the outgoing packet and the returning packet in the Record Route field. A minimum of 1 to a maximum of 9 hosts must be specified by *count*.

-s count

Specifies the timestamp for the number of hops specified by *count*.

-j host-list

Routes packets via the list of hosts specified by *host-list*. Consecutive hosts may be separated by intermediate gateways (loose source routed). The maximum number allowed by IP is 9.

-k host-list

Routes packets via the list of hosts specified by *host-list*. Consecutive hosts may not be separated by intermediate gateways (strict source routed). The maximum number allowed by IP is 9.

-w timeout

Specifies a timeout interval in milliseconds.

destination-list

Specifies the remote hosts to ping.

More Information About Ping



[Ping--Notes](#)

Ping--Notes

The **ping** command verifies connections to remote host or hosts, by sending ICMP ECHO packets to the host and listening for ECHO REPLY packets. **Ping** waits for up to 1 seconds for each packet sent and prints the number of packets transmitted and received. Each received packet is validated against the transmitted message. By default, four ECHO packets containing 64 bytes of data (a periodic uppercase sequence of alphabetic characters) are transmitted.

You can use the ping utility to test both the host name and the IP address of the host. If the IP address is verified but the hostname is not, you may have a name resolution problem. In this case, be sure that the host name you are querying is in either the local HOSTS file or in the DNS database.

More Information About Ping



[Ping](#)

Route

Manipulates network routing tables. This command is available only if the TCP/IP protocol has been installed.

route [-f] [*command* [*destination*] [*gateway*]]

Parameters

-f

Clears the routing tables of all gateway entries. If this is used in conjunction with one of the commands, the tables are cleared prior to running the command.

command

Specifies one of four commands

<u>Command</u>	<u>Purpose</u>
print	Prints a route
add	Adds a route
delete	Deletes a route
change	Modifies an existing route

destination

Specifies the host to send *command*.

gateway

Specifies the gateway.

All symbolic names used for *destination* or *gateway* are looked up in the network and host name database files NETWORKS and HOSTS, respectively. If the command is **print** or **delete**, wildcards may be used for the destination and gateway, or the gateway argument may be omitted.

Telnet

This connectivity command starts terminal emulation with a remote system running a Telnet service.

The Telnet application is found in the Accessories program group if you install the TCP/IP connectivity utilities. Telnet is a Windows Sockets-based application that simplifies TCP/IP terminal emulation.

telnet [*host* [*port*]]

Parameters

host

Specifies the host name or IP address of the remote system you want to connect to, providing compatibility with applications such as Gopher and Mosaic.

port

Specifies the remote port you want to connect to, providing compatibility with applications such as Gopher and Mosaic. The default value is specified by the **telnet** entry in the SERVICES file. If no entry exists in the SERVICES file, the default connection port value is decimal 23.

To use Telnet

1. In File Manager, choose Run from the File menu and type **telnet** and press ENTER.
2. From the Connect menu in the Telnet window, choose Remote System.
3. In the Connect dialog box, type the host name you want to connect to, and then choose the Connect button.

A connection is made, and you can begin a work session.

4. To end a session, choose the Disconnect command from the Connect menu.

You can specify your preferences for items such as the screen font and color plus emulation options by choosing commands from the Options menu. You can also use commands from the Edit menu to select, copy, and paste text from the Clipboard.

Tracert

This diagnostic utility determines the route taken to a destination by sending Internet Control Message Protocol (ICMP) echo packets with varying Time-To-Live (TTL) values to the destination. Each router along the path is required to decrement the TTL on a packet by at least 1 before forwarding it, so the TTL is effectively a hop count. When the TTL on a packet reaches 0, the router is supposed to send back an ICMP Time Exceeded message to the source system. **Tracert** determines the route by sending the first echo packet with a TTL of 1 and incrementing the TTL by 1 on each subsequent transmission until the target responds or the maximum TTL is reached. The route is determined by examining the ICMP Time Exceeded messages sent back by intermediate routers. Notice that some routers silently drop packets with expired time-to-live (TTLs) and will be invisible to **tracert**.

tracert [-d] [-h *maximum_hops*] [-j *host-list*] [-w *timeout*] *target_name*

Parameters

-d

Specifies not to resolve addresses to hostnames.

-h *maximum_hops*

Specifies maximum number of hops to search for target.

-j *host-list*


Specifies loose source route along *host-list*.

-w *timeout*

Waits the number of milliseconds specified by *timeout* for each reply.

Troubleshooting with TCP/IP Diagnostic Utilities

This topic describes how to use the diagnostic utility to find solutions to TCP/IP networking problems. The following topics are included.

 [Troubleshooting IP Connections](#)

 [Using the Diagnostic Utilities](#)

<u>Utility</u>	<u>Usage</u>
<u>arp</u>	Allows a user to view and modify the ARP (address resolution protocol) table entries on the local computer.
<u>ipconfig</u>	Displays all current TCP/IP network configuration values and accepts DHCP commands to update or release TCP/IP network configuration values.
<u>nbtstat</u>	Provides network statistics for active and pending NetBIOS over TCP/IP connections.
<u>netstat</u>	Provides network statistics for all active and pending TCP/IP connections.
<u>ping</u>	Provides a simple mechanism to determine whether a remote TCP/IP system is reachable or functioning properly in the TCP/IP network.
<u>tracert</u>	Determines the route taken to a destination.

Troubleshooting IP Connections

If you have trouble installing Microsoft TCP/IP-32 on your computer, follow the suggestions in the error messages. You can also use the **ping** utility to isolate network hardware problems and incompatible configurations, allowing you to verify a physical connection to a remote computer.


Use the **ping** utility to test both the host name and the IP address of the host.


For syntax information, see the **ping** command.

To test TCP/IP using the ping utility

1. If the computer was configured using DHCP, use **ipconfig** to get the IP address.
2. Use **ping** to check the loopback address by typing **ping 127.0.0.1** and pressing ENTER at the DOS Prompt command line. The computer should respond immediately.
3. To determine whether you configured IP properly, use **ping** with the IP address of your computer, your default gateway, and a remote host.

If you cannot use **ping** successfully at any point, verify the following:

 The computer was restarted after TCP/IP was installed and configured.

 The local computer's IP address is valid and appears correctly in the TCP/IP Configuration dialog box.



The IP address of the default gateway and remote host are correct.



IP routing is enabled and the link between routers is operational.

If you can use **ping** to connect to other computers on a different subnet but cannot connect through File Manager or with **net use** `\\server\share`, verify the following:



The computer is WINS-enabled (if the network includes WINS servers),.



The WINS server addresses are correct, and the WINS servers are functioning.



The correct computer name was used.



The target host uses NetBIOS. If not, you must use FTP or Telnet to make a connection; in this case, the target host must be configured with the FTP server daemon or Telnet server daemon, and you must have correct permissions on the target host.



The scope ID on the target host is the same as the local computer.



A router exists between your system and the target system.



LMHOSTS contains correct entries, so that the computer name can be resolved.

If the IP address responds but the host name does not, you have a name resolution problem. In this case, use the following lists of common problems in name resolution to find solutions.

Errors Related to the HOSTS File



The HOSTS file or DNS do not contain the particular host name.



The host name in the HOSTS file or in the command is misspelled or uses different capitalization. (Host names are case sensitive.)



An invalid IP address is entered for the host name in the HOSTS file.



The HOSTS file contains multiple entries for the same host on separate lines.



A mapping for a computer name-to-IP address was mistakenly added to the HOSTS file (rather than LMHOSTS).

Errors Related to the LMHOSTS File



The LMHOSTS file does not contain an entry for the remote server.



The computer name in LMHOSTS is misspelled. (Notice that LMHOSTS names are converted to uppercase.)



The IP address for a computer name in LMHOSTS is not valid.

Using the Diagnostic Utilities

In addition to **ping**, the other diagnostic utilities such as **netstat** and **nbtstat** can be used to find and resolve connection problems. Although this is not a complete list, these examples show how you might use these utilities to track down problems on the network.

To determine the cause of long connect times after adding to LMHOSTS



Because this behavior can occur with large LMHOSTS file with an entry at the end of the file, mark the entry in LMHOSTS as a preloaded entry by following the mapping with the #PRE tag. Then use the **nbtstat -R** command to update the local name cache immediately.

Or, place the mapping higher in the LMHOSTS file.

The LMHOSTS file is parsed sequentially to locate entries without the #PRE keyword. Therefore, you should place frequently used entries near the top of the file and place the #PRE entries near the bottom.

To determine the cause of connection problems when specifying a server name



Use the **nbtstat -n** command to determine what name the server registered on the network.

The output of this command lists several names that the computer has registered. A name resembling the computers computer name should be present. If not, try one of the other unique names displayed by **nbtstat**.

The **nbtstat** utility can also be used to display the cached entries for remote computers from either #PRE entries in LMHOSTS or recently resolved names. If the name the remote computers are using for the server is the same, and the other computers are on a remote subnet, be sure that they have the computers mapping in their LMHOSTS files.

To determine why only IP addresses work for connections to foreign systems but not host names

1. Make sure that the appropriate HOSTS file and DNS setup have been configured for the computer by checking the host name resolution configuration using the Network option in Control Panel and then choosing the DNS button in the TCP/IP Configuration dialog box.
2. If you are using a HOSTS file, make sure that the name of the remote computer is spelled the same and capitalized the same in the file and by the application using it.
3. If you are using DNS, be sure that the IP addresses of the DNS servers are correct and in the proper order. Use **ping** with the remote computer by typing both the host name and IP address to determine whether the host name is being resolved properly.

To determine why a TCP/IP connection to a remote computer is not working properly



Use the **netstat -a** command to show the status of all activity on TCP and UDP ports on the local computer.

The state of a good TCP connection is usually established with 0 bytes in the send and receive queues. If data is blocked in either queue or if the state is irregular, there is probably a problem with the connection. If not, you are probably experiencing network or application delay.

To determine why the banner displayed with Telnet identifies a different computer, even when specifying the correct IP address

1. Make sure the DNS name and hosts table are up to date.
2. Make sure that two computers on the same network are not mistakenly configured with the same IP address.

The ethernet and IP address mapping is done by the ARP (address resolution protocol) module, which believes the first response it receives. So the impostor computers reply sometimes comes

back before the intended computers reply.

These problems are difficult to isolate and track down. Use the **arp -g** command to display the mappings in the ARP cache. If you know the ethernet address for the intended remote computer, you can easily determine whether the two match. If not, use **arp d** to delete the entry, then use **ping** with the same address (forcing an ARP), and check the ethernet address in the cache again by using **arp -g**.

Chances are that if both computers are on the same network, you will eventually get a different response. If not, you may have to filter the traffic from the impostor host to determine the owner or location of the system.

Troubleshooting TCP/IP Database Files

The following UNIX-style database files are stored in the `\systemroot\SYSTEM32\DRIVERS\ETC` when you install Microsoft TCP/IP:

HOSTS

Provides hostname-to-IP address resolution for Windows Sockets applications

LMHOSTS. SAM (sample file)

Provides NetBIOS name-to-IP address resolution for Windows networking

NETWORKS

Provides network name-to-network ID resolution for TCP/IP management

PROTOCOLS

Provides protocol name-to-protocol ID resolution for Windows Sockets applications

SERVICES

Provides service name-to-port ID resolution for Windows Sockets applications

To troubleshoot any of these files on a local computer:



Make sure the format of entries in each file match the format defined in the sample file originally installed with Microsoft TCP/IP.



Check for spelling or capitalization errors.








Check for invalid IP addresses and identifiers.

Introduction to Microsoft TCP/IP-32 Installation

These topics describe how to install Microsoft TCP/IP-32 for Windows for Workgroups and how to configure the protocols on your computer. Microsoft TCP/IP-32 is a 32-bit implementation of the industry-standard TCP/IP protocol.

In these topics, it is assumed that Windows for Workgroups version 3.11 has already been successfully installed on your computer.

The following topics are included:

-  [Before Installing Microsoft TCP/IP-32](#)
-  [Installing Microsoft TCP/IP-32](#)
-  [Configuring Microsoft TCP/IP-32](#)
-  [Configuring Microsoft TCP/IP-32 to Use DNS](#)
-  [Configuring Advanced Microsoft TCP/IP-32 Options](#)

Before Installing Microsoft TCP/IP-32

Important: The values used for manually configuring Microsoft TCP/IP-32 must be supplied by the network administrator.

You need to know whether you can use Dynamic Host Configuration Protocol (DHCP) configuration. You can choose this option if a DHCP server is installed on your internetwork.

If you cannot use DHCP for automatic configuration, you need to obtain these values from the network administrator so you can configure Microsoft TCP/IP-32 manually:



The IP address and subnet mask for each network adapter card installed on the computer.



The IP address for the default local gateway (IP router).



Whether your computer will use Domain Name System (DNS) and, if so, the IP addresses of the DNS servers on the internetwork.



WINS server addresses, if WINS servers are available on your network.

See Also

[Domain Name System Addressing](#)

[IP Addressing](#)

[Using Dynamic Host Configuration Protocol](#)

Installing Microsoft TCP/IP-32

You use the Network Setup application to install Microsoft TCP/IP-32.

To install Microsoft TCP/IP-32 on a Windows for Workgroups computer

1. In the Network program group in Program Manager, double-click the Network Setup icon.
2. In the Network Setup dialog box, choose the Drivers button to display the Network Drivers dialog box.
3. In the Network Drivers list, select the adapter over which you want to run TCP/IP, and choose the Add Protocol button. The list shows all the network adapters and protocols installed on your computer.

Important: If you have any TCP/IP stack from another vendor loaded, you must remove it before installing Microsoft TCP/IP-32. To do this, select it in the Network Drivers list and choose the Remove button. Then close the Network Setup dialog box, and start the process again to install this new software. Also, you can only have three protocols installed, so if you already have three protocols on your computer, you must remove one before installing this new protocol.

4. In the Add Network Protocol dialog box, select Unlisted Or Updated Protocol, and then choose OK.
5. In the Install Driver dialog box, specify the drive letter and path for your Microsoft TCP/IP-32 for Windows for Workgroups 3.11 disk, and then choose OK.
6. In the Unlisted Or Updated Protocol dialog box, the Protocols list shows the options available on the installation disk. Select Microsoft TCP/IP-32 3.11, and then choose OK.

The Microsoft TCP/IP-32 for Windows for Workgroups distribution files will be copied to your hard disk.

7. When the Network Drivers dialog box reappears, select Microsoft TCP/IP-32 3.11, and then choose the Setup button.

The Microsoft TCP/IP Configuration dialog box appears so you can configure your IP address, subnet mask, and other settings.

When you finish configuring Microsoft TCP/IP-32, you must restart your computer for the changes to take effect.

If you have trouble installing Microsoft TCP/IP-32 on your computer, follow the suggestions in the error messages. You can also use diagnostic utilities such as **ping** to isolate network hardware problems and incompatible configurations.

See Also

[Configuring Microsoft TCP/IP-32 Manually](#)

[Troubleshooting with TCP/IP Diagnostic Utilities](#)

Configuring Microsoft TCP/IP-32

For TCP/IP to work on your computer, it must be configured with the IP addresses, subnet masks, and default gateway for each network adapter on the computer. Microsoft TCP/IP-32 can be configured using two different methods:

Using Dynamic Host Configuration Protocol

If there is a DHCP server on your internetwork, it can automatically configure TCP/IP for your computer using DHCP.

Configuring TCP/IP Manually

If there is no DHCP server, you must manually configure all TCP/IP settings.

Using DHCP

The best method for ensuring easy and accurate installation of Microsoft TCP/IP-32 is to use automatic DHCP configuration, which uses the DHCP server to configure your local computer with the correct IP address, subnet mask, and default gateway.

You can take advantage of this method for configuring Microsoft TCP/IP-32 if there is a DHCP server installed on your network. The network administrator can tell you if this option is available.

To configure Microsoft TCP/IP-32 using DHCP

1. Make sure the Enable Automatic DHCP Configuration option is checked in the Microsoft TCP/IP Configuration dialog box.
2. When you restart the computer after completing installation, the DHCP server automatically provides the correct configuration information for your computer.

Any manual settings will override the automatic settings provided by DHCP. As a general rule, you should not change the automatic settings unless you specifically want to override a setting provided by DHCP.

Configuring Microsoft TCP/IP-32 Manually

After the Microsoft TCP/IP-32 protocol software is installed on your computer, you must manually provide valid addressing information if you cannot use automatic DHCP configuration.

Caution

Be sure to use the values for IP addresses and subnet masks that are supplied by your network administrator, because the network must not have duplicate IP addresses. If duplicate addresses occur, some computers on the network may function unpredictably.

To manually configure or reconfigure the TCP/IP protocol

1. In the Network Drivers dialog box, select Microsoft TCP/IP-32 3.11, and then choose the Setup button.
2. In the Adapter list, select the network adapter for which you want to set IP addresses.

The Adapter list contains all network adapters to which IP is bound on this computer. Initially, this list includes all adapters installed on this computer.

You must set specific IP addressing information for each bound adapter. The bindings for a network adapter determine how network protocols and other layers of network software work together.

3. For each bound network adapter, type values in the IP Address and Subnet Mask boxes. These parameters must be configured for each adapter with correct values provided by the network administrator.



The value in the IP Address box identifies the IP address for your local computer or, if more than one network card is installed in the computer, for the network adapter card selected in the Adapter box.



The value in the Subnet Mask box identifies the network membership for the selected network adapter and its host ID. The system uses this value to separate the IP address into host and network IDs.

Important: If you check the Enable Automatic DHCP Configuration option, any values you enter manually in this dialog box will override the automatic values that DHCP sets. If you turn on automatic DHCP configuration in this dialog box, you do not need to enter values and you will not be able to enter address or mask values.

4. For each network adapter, type the correct IP address value in the Default Gateway box, as provided by the network administrator.

This value specifies the IP address of the default gateway (or IP router) used to forward packets to other networks or subnets. This value should be the IP address of your local gateway.

The default gateway address is required only for computers on internetworks. If this parameter is not provided, IP functionality is limited to the local subnet unless a route is specified with the TCP/IP **route** command.

5. If there are WINS servers installed on your network and you want to use WINS in combination with broadcast name queries to resolve computer names, type IP addresses in the boxes for the primary and, optionally, the secondary WINS servers. The network administrator should provide the correct values for these parameters. These are global values for the computer, not just for individual adapters.

If IP addresses for WINS servers are not provided, the system uses **b-node** name query broadcasts plus the local LMHOSTS file to resolve computer names to IP addresses. Broadcast resolution is limited to the local network.

6. If you want to configure the advanced Microsoft TCP/IP-32 options, choose the Advanced button, and continue with the configuration procedure.
7. If you want to use the Domain Name System (DNS) for host name resolution, choose the DNS

button, and continue with the configuration procedure.

8. If you do not want to configure DNS or advanced options, or if you have completed the other configuration procedures, choose the OK button. When the Network Drivers dialog box reappears, choose the OK button.

Microsoft TCP/IP-32 has been configured. You must restart the computer for the configuration to take effect.

The \WINDOWS and \WINDOWS\SYSTEM directories will contain several files after Microsoft TCP/IP-32 is installed, including a default HOSTS file and a sample LMHOSTS file. The network administrator may require that replacement HOSTS or LMHOSTS files be used instead of these default files.

See Also

[Configuring Advanced Microsoft TCP/IP-32 Options](#)

[Configuring Microsoft TCP/IP-32 to Use DNS](#)

[Name Resolution for Windows Networking](#)

Configuring Microsoft TCP/IP-32 to Use DNS

Although TCP/IP uses IP addresses to identify and reach computers, users typically prefer to use host names. Domain Name System (DNS) is a naming service generally used in the UNIX networking community to provide standard naming conventions for IP workstations. TCP/IP utilities, such as **ftp** and **telnet**, can also use DNS in addition to the HOSTS file to find systems when connecting to foreign hosts or systems on your network.

Contact the network administrator to find out whether you should configure your computer to use DNS. Usually you will use DNS if you are using TCP/IP to communicate over the Internet or if your private internetwork uses DNS to distribute host information. For information, see [Domain Name System Addressing](#).

Microsoft TCP/IP provides a DNS client for resolving Internet or UNIX system names. Microsoft Windows networking provides dynamic name resolution for NetBIOS computer names via WINS servers and NetBIOS over TCP/IP.

If you choose to use DNS, you must configure how your computer will use DNS and the HOSTS file. TCP/IP must be installed before you set up the DNS connectivity options. DNS configuration is global for all network adapters installed on a computer.

Note: A TCP domain is not the same as a Windows NT or LAN Manager domain.

To configure or reconfigure TCP/IP DNS connectivity

1. In the Network Drivers dialog box, select Microsoft TCP/IP-32 3.11, and then choose the Setup button. The Microsoft TCP/IP Configuration dialog box appears.
2. Choose the DNS button. The DNS Configuration dialog box appears.
3. Optionally, type a name in the Host Name box.

The name can be any combination of A-Z letters, 0-9 numbers, and the hyphen (-), plus the period (.) character used as a separator. By default, this value is the Windows for Workgroups computer name, but the network administrator can assign another host name without affecting the computer name.

Note: Some characters that can be used in Windows for Workgroups computer names, particularly the underscore, cannot be used in host names.

The host name is used to identify the local computer by name for authentication by some utilities. Other TCP/IP-based utilities can use this value to learn the name of the local computer. Host names are stored on DNS servers in a table that maps names to IP addresses for use by DNS.

4. Optionally, type a name in the TCP Domain Name box. This is usually an organization name followed by a period and an extension that indicates the type of organization, such as microsoft.com.

The name can be any combination of A-Z letters, 0-9 numbers, and the hyphen (-), plus the period (.) character used as a separator.

This TCP (or DNS) Domain Name is used with the host name to create a fully qualified domain name (FQDN) for the computer. The FQDN is the host name followed by a period (.) followed by the domain name. For example, this could be **wolverine.microsoft.com**, where **wolverine** is the host name and **microsoft.com** is the domain name. During DNS queries, the local domain name is appended to short names.

5. In the Domain Name System (DNS) Server Search Order box, type the IP address of a DNS server that will provide name resolution. Then choose the Add button to move the IP address to the list on the right. The network administrator should provide the correct values for this parameter.

You can add up to three IP addresses for DNS servers. The servers running DNS will be queried in

the order listed. To change the order of the IP addresses, select an IP address to move, and then use the up and down buttons. To remove an IP address, select it and choose the Remove button.

6. In the Domain Suffix Search Order box, type the domain suffixes to add to your domain suffix search list, and then choose the Add button.

This list specifies the DNS domain suffixes to be appended to host names during name resolution. You can add up to six domain suffixes. To change the search order of the domain suffixes, select a domain name to move, and use the up- and down-arrow buttons. To remove a domain name, select it and choose the Remove button.

7. When you are done setting DNS options, choose the OK button.
8. When the Microsoft TCP/IP Configuration dialog box reappears, choose the OK button. When the Network Drivers dialog box reappears, choose the OK button.

The settings take effect after you restart the computer.

Configuring Advanced Microsoft TCP/IP-32 Options

If your computer has multiple network adapters connected to different networks using TCP/IP, you can choose the Advanced button in the Microsoft TCP/IP Configuration dialog box to configure options for the adapters or to configure alternate default gateways.

To configure or reconfigure advanced Microsoft TCP/IP-32 options

1. In the Network Drivers dialog box, select Microsoft TCP/IP-32 3.11, and then choose the Setup button. The Microsoft TCP/IP Configuration dialog box appears.
2. Choose the Advanced button. The Advanced Microsoft TCP/IP Configuration dialog box appears.
3. In the Adapter box, select the network adapter for which you want to specify advanced configuration values. The IP address and default gateway settings in this dialog box are defined only for the selected network adapter.
4. In the IP Addresses box, type an additional IP address and subnet mask for the selected adapter. Then choose the Add button to move the IP address to the list on the right. Repeat this process for each additional IP address. The network administrator should provide the correct values for this parameter.

This list specifies up to five additional IP addresses and subnet masks for identifying the selected network adapter. This can be useful for a system connected to one physical network that contains multiple logical IP networks.

5. In the Default Gateway box, type the IP address for an additional gateway that the selected adapter can use. Then choose the Add button to move the IP address to the list on the right. Repeat this process for each additional gateway. The network administrator should provide the correct values for this parameter.

This list specifies up to five additional default gateways for the selected network adapter.

To change the priority order for the gateways, select an address to move and use the up or down buttons. To remove a gateway, select it and choose the Remove button.

6. If you want to use DNS for name resolution on Windows networks, check the Enable DNS For Windows Name Resolution option.

If this option is checked, the system find the DNS server by using the IP address specified in the DNS Configuration dialog box, as described in [Configuring TCP/IP to use DNS](#).

7. If you want to use the LMHOSTS file for NetBIOS name resolution on Windows networks, check the Enable LMHOSTS Lookup option. If you already have a configured LMHOSTS File, choose the Import LMHOSTS button and specify the directory path for the LMHOSTS file you want to use.

By default, Enable LMHOSTS is not checked. If you check this option, the LMHOSTS file in the \WINDOWS directory is used if it exists.

8. Optionally, in the Scope ID box, type the computer's scope identifier, if required on an internetwork that uses NetBIOS over TCP/IP.

To communicate with each other, all computers on a TCP/IP internetwork must have the same scope ID. Usually this value is left blank. A scope ID may be assigned to a group of computers that only will communicate with each other and no other systems. Such computers can find each other if their scope IDs are identical. Scope IDs are used only for communication based on NetBIOS over TCP/IP.

The network administrator should provide the correct value, if required.

9. To turn on static IP routing, check the Enable IP Routing option.

This option allows this computer to forward datagrams between local IP interfaces.

This option is not available if your computer has only one network adapter and one IP address and subnet mask pair. If you have more than one IP address, this option is available.

10. If you want this computer to be used to resolve names based on the WINS database, check the Enable WINS Proxy Agent option.

This option allows other computers configured for b-node broadcast name resolution to benefit from the name services provided by a WINS server. This option is available only if IP addresses for WINS servers are provided in the Microsoft TCP/IP Configuration dialog box, as described in Configuring TCP/IP.

11. When you are done setting advanced options, choose the OK button. When the Microsoft TCP/IP Configuration dialog box reappears, choose the OK button. When the Network Drivers dialog box reappears, choose the OK button to complete advanced TCP/IP configuration.









Microsoft TCP/IP Configuration

After the Microsoft TCP/IP-32 software is installed on your computer, you must manually provide valid addressing information if you did not use the automatic DHCP configuration option when you installed TCP/IP.

Use the Microsoft TCP/IP Configuration dialog box to specify settings for TCP/IP. You choose the Network Setup icon in Control Panel, then in the Network Drivers dialog box, select Microsoft TCP/IP-32 3.11, and then choose the Setup button. The Microsoft TCP/IP Configuration dialog box appears.

To configure Microsoft TCP/IP-32, you must know the IP address and subnet mask for your network adapter.

Choose one of the following buttons for information about this dialog box:

-  [Adapter](#)
-  [IP Address](#)
-  [Subnet Mask](#)
-  [Default Gateway](#)
-  [Enable Automatic DHCP Configuration](#)
-  [Primary and Secondary WINS Servers](#)
-  [DNS Button](#)
-  [Advanced Button](#)

Default Gateway



For each network adapter, type the correct IP address value in the Default Gateway box, as provided by the network administrator.

This value specifies the IP address of the default gateway (or IP router) used to forward packets to other networks or subnets. This value should be the IP address of your local gateway.

This parameter is required only for computers on internetworks. If this parameter is not provided, IP functionality will be limited to the local subnet unless a route is specified with the TCP/IP **route** command.

Adapter



Select a network adapter for which you want to set IP addresses.

The Adapter list contains all network adapters to which IP is bound on this computer. Initially, this list includes all adapters installed on this computer.

You must set the IP addressing information for each bound adapter. The bindings for a network adapter determine how network protocols and other layers of network software work together.

IP Address



For each bound network adapter, type the correct values in the IP Address box.

This parameter must be configured for each adapter with correct values provided by the network administrator.

This value identifies the IP address for your local computer, or if more than one network card is installed in the computer, for the network adapter card selected in the Adapter box.

Subnet Mask



Enter the value in the Subnet Mask box that identifies the host ID and network membership for the selected network adapter.

If you check the Enable Automatic DHCP Configuration option, any values you enter manually in this dialog box will override the automatic values that DHCP sets. If you turn on automatic DHCP configuration in this dialog box, you do not need to enter values and you will not be able to enter IP address or mask values.

Caution

Duplicate addresses must not occur on the network. Duplicate addresses will cause some computers on the network to function unpredictably.

Enable Automatic DHCP Configuration



To turn on automatic configuration of TCP/IP parameters for this computer, check this option if there is a DHCP server on your internetwork to support dynamic host configuration. This is the preferred method for configuring Microsoft TCP/IP-32 on most Windows for Workgroups computers.

If you check this option, remember that any values you enter manually in this dialog box will override the automatic values

that DHCP sets.

Advanced Button



If you want to configure the advanced Microsoft TCP/IP-32 options, choose the Advanced button, which displays the Advanced Microsoft TCP/IP Configuration dialog box.

DNS Button



If you want to use the Domain Name System (DNS) for host name resolution, choose the DNS button, which displays the TCP/IP Connectivity Configuration dialog box.

Primary and Secondary WINS Servers



If you want to use WINS in combination with name query broadcasts to resolve computer names to IP addresses, type the IP address for the primary and secondary WINS servers in these boxes.

Your network administrator must supply these addresses. If addresses are not provided, Microsoft TCP/IP uses name query broadcasts plus the local LMHOSTS file to resolve computer names to IP addresses.









LMHOSTS and HOSTS Files

These files are stored on the local computer and contain lists of known IP addresses mapped with corresponding host names. This method of name resolution was the predecessor to DNS and is used in Windows for Workgroups for small-scale networks or remote subnets where WINS is not available. The computer checks its LMHOSTS file to find an IP address for a particular remote host.

Advanced Microsoft TCP/IP Configuration

Use the Advanced Microsoft TCP/IP Configuration dialog box if your computer has multiple network adapters connected to different networks using TCP/IP. To display this dialog box, choose the Advanced button in the Microsoft TCP/IP Configuration dialog box. Use this dialog box to configure options for the adapters or to configure alternate default gateways.

Choose one of the following buttons for information about this dialog box:

-  Adapter
-  IP Address and Subnet Mask
-  Default Gateway(s)
-  Enable DNS for Windows Name Resolution
-  Enable IP Forwarding
-  Enable LMHOSTS Lookup
-  Enable WINS Proxy Agent
-  Scope ID

Adapter



Select the network adapter for which you want to specify advanced configuration values.

The IP address and default gateway settings in this dialog box are defined only for the selected network adapter.

IP Address and Subnet Mask



Type an additional IP address and subnet mask for the selected adapter. Then choose the Add button to move the IP address to the list on the right. Repeat this process for each additional IP address. The network administrator should provide the correct values for this parameter.

This list specifies up to 5 additional IP addresses and subnet masks for identifying the selected network adapter. This can be useful for a system connected to one physical network that contains multiple logical IP networks.

Default Gateway(s)



Type the IP address for an additional gateway that the selected adapter can use. Then choose the Add button to move the IP address to the list on the right. Repeat this process for each additional gateway. The network administrator should provide the correct values for this parameter.

This list specifies up to 5 additional default gateways for the selected network adapter.

To change the priority order for the gateways



Select an address to move and use the up or down arrow buttons.

To remove a gateway



Select it and choose the Remove button.

Enable DNS for Windows Name Resolution



Check this option if you want to use DNS name resolution on Windows networks.

If this option is checked, the system finds the DNS server by using the IP address specified in the Microsoft TCP/IP Connectivity Configuration dialog box, as described earlier in this chapter. Checking this option enables DNS name resolution for use by Windows networking applications.

Enable LMHOSTS Lookup



Check the Enable LMHOSTS Lookup option if you want to use the LMHOSTS file for name resolution.

If you check this option, the LMHOSTS file in the \WINDOWS directory is used.

Enable IP Forwarding



Check the Enable IP Forwarding option to turn on static IP routing.

This option allows this computer to forward datagrams between local IP interfaces based on entries in the route table made using the **route** utility.

This option is not available if your computer has only one network adapter and one IP address and subnet mask pair. If you have more than one IP address, this option is available.

Enable WINS Proxy Agent

This option allows other computers configured for broadcast name resolution to use this computer as a gateway to a WINS server.



Check the Enable WINS Proxy Agent option if you want this computer to be used as a gateway to a WINS server.

Important

You can only enable a WINS proxy agent if WINS servers are available on your internetwork. This option is only enabled if IP addresses for WINS servers are provided in the Microsoft TCP/IP Configuration dialog box.

Scope ID



Type the computer's scope identifier, if required.

To be able to communicate, all computers on a TCP/IP internetwork must have the same scope ID. This identifies information to use if a DNS server is not found for name resolution. Usually this value is left blank. The network administrator should provide the correct value for this parameter, if required.

TCP/IP Connectivity Configuration

Use the TCP/IP Connectivity Configuration dialog box to set a number of options for Microsoft TCP/IP-32 and other TCP/IP-based applications. This dialog box appears when you choose the DNS button in the Microsoft TCP/IP Configuration dialog box.

Contact the network administrator to find out whether you should configure your computer to use DNS. Usually you will use DNS if you are using TCP/IP to communicate over the Internet or if your private internetwork uses DNS to distribute host information. Windows networking provides a dynamic name resolution system of its own for the local network.

If you choose to use DNS, you must configure how your computer will use DNS and the HOSTS file. DNS configuration is global for all network adapters installed on a computer. The settings take effect after you restart the computer.

Choose one of the following buttons for information about this dialog box:



Host Name



TCP Domain Name



Domain Name System (DNS) Search Order



Domain Suffix Search Order

Host Name



Optionally, type a name in the Host Name box. The name can be any combination of A-Z letters and 0-9 numbers, and the hyphen (-), plus the period (.) character used as a separator.

To determine the correct host name to use, ask your network administrator.

The host name is used to identify the local computer by name for authentication by some utilities. Other TCP/IP-based utilities can use this value to learn the name of the local computer. By default, this value is the Windows for Workgroups computer name, but your network administrator can assign another host name without affecting the computer name. Names are stored on DNS servers in a table that maps to Internet Protocol (IP) addresses for use by the Domain Name System (DNS).

TCP Domain Name



Optionally, type a name in the TCP Domain Name box. This is usually an organization name followed by a period and an extension that indicates the type of organization, such as microsoft.com.

The name can be any combination of A-Z letters and 0-9 numbers, and the hyphen (-), plus the period (.) character used as a separator.

To determine the correct TCP Domain Name for your group, ask your network administrator.

This TCP Domain Name is used with the host name to create a fully qualified domain name (FQDN) for the computer. The FQDN is the host name followed by a period (.) followed by the domain name. For example, this could be **corp01.trey.com**, where **corp01** is the host name and **trey.com** is the domain name. During DNS queries, the local domain name is appended to short names.

Note: A DNS or TCP domain is not the same as a Windows NT or LAN Manager domain.

Domain Name System (DNS) Search Order



Type the IP address of a DNS server that will provide name resolution. Then choose the Add button to move the IP address to the list on the right. The network administrator should provide the correct values for this parameter.

You can add up to three IP addresses for DNS servers. The servers running DNS will be queried in the order listed.

To change the list order, select an entry. Then from the Order buttons, choose the Up button to move the entry higher in the

list, or choose the Down button to move it lower in the list. To remove an IP address, select it and choose the Remove button.

If the Name Resolution method is Hosts File Only, this option is not available.

Domain Suffix Search Order



Type the domain suffixes to add to your domain suffix search list, and then choose the Add button.

This list specifies the DNS domain suffixes to be appended to host names during host name resolution. You can add up to 6 domain suffixes. To change the search order of the domain suffixes, select a domain name to move, and use the up and down arrow buttons. To remove a domain name, select it and choose the Remove button.

WINS Server

A Windows NT Server 3.5 computer running Microsoft TCP/IP and the Windows Internet Name Service server software.

When WINS servers are installed and configured on an internetwork, almost all users (even those working with non-Windows Networking systems) can enjoy the benefits of using computer names (rather than IP addresses) to communicate with systems that are not on the local network.

IP Address

The Internet Protocol (IP) address is used to identify a system on a network and to specify routing information on an internetwork. Each system on the internetwork must be assigned a unique IP address, which is made up of the official *network ID* (assigned by InterNIC) plus a unique *host ID* assigned by the network administrator.

An IP address is usually represented in dotted decimal notation, with each of the 4 bytes of the address separated with periods. For example: 102.53.94.97

To determine the correct IP address, ask your network administrator.

DHCP

Dynamic Host Configuration Protocol. A protocol that

provides safe, reliable, and simple TCP/IP configuration. There must be one or more DHCP servers installed on the network for users to take advantage of automatic DHCP configuration.

When DHCP configuration is used, the computer is automatically configured for TCP/IP when the computer is started after the Microsoft TCP/IP software is installed.

Domain Name System (DNS)

A hierarchical naming service usually provided on networked UNIX systems for mapping host names with Internet Protocol (IP) addresses.

Although TCP/IP uses IP addresses to identify and reach computers, users typically prefer using names. Domain Name System (DNS) is a naming service generally used in the UNIX networking community to provide standard naming conventions for IP workstations. TCP/IP utilities, such as **ftp** and **telnet**, can also use DNS in addition to the HOSTS file to find systems when connecting to foreign hosts or systems on your network.

DHCP Server

A DHCP server is a Windows NT Server 3.5 computer that is running Microsoft TCP/IP and Dynamic Host Configuration Protocol server software.

When one or more DHCP servers are installed and configured on an internetwork, Windows for Workgroups client computers can use it to automatically configure their IP addresses, subnet masks, default gateways, and other information.

WINS

Windows Internet Name Service. A protocol for a distributed database for registering and querying computer names-to-IP address mappings in a routed network environment.

WINS Proxy Agent

Windows for Workgroups client computers can query WINS servers directly to resolve computer names as IP addresses. Foreign systems (non-Windows networking computers) can use WINS proxy agents, which intercept name query broadcasts from other computers and resolve computer names via WINS servers.

Any Windows for Workgroups computer with Microsoft TCP/IP-32 can be designated a WINS proxy agent on an internetwork where WINS servers are available with TCP/IP installed.

Subnet Mask

A 32-bit value that is used by the IP software to extract the network ID and host ID from the IP address.

The subnet mask is usually represented in dotted decimal notation, with each of the 4 bytes of the address separated with periods. For example: 255.255.0.0

To determine your correct subnet mask, ask your network administrator.

TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) is a networking protocol that provides communication across interconnected networks made up of computers with diverse hardware architectures and various operating systems. TCP/IP can be used with Windows for Workgroups, to communicate with devices using other Microsoft networking products, or to communicate with non-Microsoft hosts, such as UNIX.

The TCP/IP protocol family is a standard set of networking protocols, or rules, that govern how data is passed between computers on a network. TCP/IP is used to connect the Internet, the worldwide internetwork connecting universities, research labs, U.S. defense installations, and corporations. (By convention, Internet is capitalized when referring to the worldwide internetwork.) The same protocols are used in private internetworks that connect several local area networks.

Default Gateway






TCP/IP networks are connected by gateways, or routers, that know which networks are connected on the

internetwork. The default gateway is the local IP router that is used to find destinations beyond the local network. The default gateway forwards packets to other gateways until the packet is eventually delivered to a gateway connected to the specified destination.

Networking Concepts for TCP/IP

The following topics explain the various components of the Internet Protocol suite and IP addressing. For additional information about these topics, see the related Requests for Comments or *Internetworking with TCP/IP, Volume I*, by Douglas E. Comer (Prentice Hall, 1991). These topics provide conceptual information about two key features for Microsoft TCP/IP: Dynamic Host Configuration Protocol (DHCP) and Windows Internet Name Service (WINS), and also describe how Domain Name System (DNS) addressing works and how TCP/IP fits in the Windows network architecture.

The following topics are included:

-  [Internet protocol suite](#)
-  [IP addressing](#)
-  [Dynamic Host Configuration Protocol](#)
-  [WINS and Name Resolution for Windows Networking](#)
-  [TCP/IP and Windows networking](#)

Internet Protocol Suite

TCP/IP refers to the Internet suite of protocols. It includes a set of standards that specify how computers communicate and gives conventions for connecting networks and routing traffic through the connections.

The Internet protocols are a result of a Defense Advanced Research Projects Agency (DARPA) research project on network interconnection in the late 1970s. It was mandated on all United States defense long-haul networks in 1983 but was not widely accepted until it was integrated with 4.2 Berkeley Software Distribution (BSD) UNIX. The popularity of TCP/IP is based on:



Robust client-server framework. TCP/IP is an excellent client-server application platform, especially in wide-area network (WAN) environments.



Information sharing. Thousands of organizations share data, electronic mail, and services on the Internet using TCP/IP.



General availability. Implementations of TCP/IP are available on nearly every popular computer operating system. Source code is widely available for many implementations. Vendors for bridges, routers, and network analyzers all offer support for the TCP/IP protocol suite within their products.

The topics below introduce the components of the IP protocol suite. Some knowledge of the architecture and interaction between TCP/IP components is useful for both administrators and users, but most of the details discussed here are transparent when you are actually using TCP/IP.



[Transmission Control Protocol and Internet Protocol](#)



[User Datagram Protocol](#)



[Address Resolution Protocol and Internet Control Message Protocol](#)



[Routing and IP Gateways](#)

Transmission Control Protocol and Internet Protocol

Transmission Control Protocol (TCP) and Internet Protocol (IP) are only two members of the IP protocol suite. IP is a protocol that provides packet delivery for all other protocols within the TCP/IP family. IP provides a best-effort, connectionless delivery system for computer data. That is, IP packets are not guaranteed to arrive at their destination, nor are they guaranteed to be received in the sequence in which they were sent. The protocol's checksum feature confirms only the IP header's integrity. Thus, responsibility for the data contained within the IP packet (and the sequencing) is assured only by using higher-level protocols.

Perhaps the most common higher-level IP protocol is TCP. TCP supplies a reliable, connection-based protocol over (or encapsulated within) IP. TCP guarantees the delivery of packets, ensures proper sequencing of the data, and provides a checksum feature that validates both the packet header and its data for accuracy. In the event that the network either corrupts or loses a TCP/IP packet during transmission, TCP is responsible for retransmitting the faulty packet. This reliability makes TCP/IP the protocol of choice for session-based data transmission, client-server applications, and critical services such as electronic mail.

This reliability has a price. TCP headers require the use of additional bits to provide proper sequencing of information, as well as a mandatory checksum to ensure reliability of both the TCP header and the packet data. To guarantee successful data delivery, the protocol also requires the recipient to acknowledge successful receipt of data.

Such acknowledgments (or ACKs) generate additional network traffic, diminishing the level of data throughput in favor of reliability. To reduce the impact on performance, most hosts send an acknowledgment for every other segment or when an ACK timeout expires.

User Datagram Protocol

If reliability is not essential, User Datagram Protocol (UDP), a TCP complement, offers a connectionless datagram service that guarantees neither delivery nor correct sequencing of delivered packets (much like IP). Higher-level protocols or applications may provide reliability mechanisms in addition to UDP/IP. UDP data checksums are optional, providing a way to exchange data over highly reliable networks without unnecessarily consuming network resources or processing time. When UDP checksums are used, they validate both header and data. ACKs are also not enforced by the UDP protocol; this is left to higher-level protocols.

Address Resolution Protocol and Internet Control Message Protocol

Two other protocols in the IP suite perform important functions, although these are not directly related to the transport of data: Address Resolution Protocol (ARP) and Internet Control Message Protocol (ICMP). ARP and ICMP are maintenance protocols that support the IP framework and are usually invisible to users and applications.

IP packets contain both source and destination IP addresses, but the hardware address of the destination node must also be known. IP acquires a node's hardware address by broadcasting a special inquiry packet (an ARP request packet) containing the IP address of the node with which it is attempting to communicate. All of the ARP-enabled nodes on the IP network detect these broadcasts, and the node that owns the IP address in question replies by sending its hardware address to the requesting node in an ARP reply packet. The hardware/IP address mapping is then stored in the requesting node's ARP cache for subsequent use. Because the ARP reply can also be broadcast to the network, it is likely that other nodes on the network can use this information to update their own ARP caches. (You can use the **arp** utility to view the ARP tables.)

ICMP allows two nodes on an IP network to share IP status and error information. This information can be used by higher-level protocols to recover from transmission problems or by network administrators to detect network trouble. Although ICMP packets are encapsulated within IP packets, they are not considered to be a higher-level protocol (ICMP is required in every TCP/IP implementation). The **ping** utility makes use of the ICMP echo request and echo reply packets to determine whether a particular IP node on a network is functional. This is useful for diagnosing IP network or gateway failures.

IP Addressing

A host is any device attached to the network that uses TCP/IP. To receive and deliver packets successfully between hosts, TCP/IP relies on three pieces of information that the user provides: IP address, subnet mask, and default gateway.

The network administrator provides each of these pieces of information for configuring TCP/IP on a computer. Windows for Workgroups users on networks with DHCP servers can take advantage of automatic system configuration and do not need to manually configure TCP/IP parameters. The following topics provide details about IP addresses:



[IP Addresses](#)



[Network ID and Host ID](#)



[Subnet Masks](#)



[Routing and IP Gateways](#)

See Also

[Dynamic Host Configuration Protocol](#)

IP Addresses

Every host interface, or node, on a TCP/IP network is identified by a unique IP address. This address is used to identify a node on a network; it also specifies routing information in an internetwork. The IP address identifies a computer as a 32-bit address that is unique across a TCP/IP network. An address is usually represented in dotted decimal notation, which depicts each octet (eight bits, or one byte) of an IP address as its decimal value and separates each octet with a period. An IP address looks like this:

102.54.94.97

Important: Because IP addresses identify nodes on an interconnected network, each node on the internetwork must be assigned a unique IP address, valid for its particular network.

See Also

[Network ID and Host ID](#)

[Subnet Masks](#)

Network ID and Host ID

Although an IP address is a single value, it contains two pieces of information: the network ID and the host (or system) ID for your computer.



The network ID identifies a group of computers and other devices that are all located on the same logical network. In internetworks (networks formed by a collection of local area networks), there is a unique network ID for each network.



The host ID identifies your computer within a particular network ID. (A host is any device that is attached to the network and uses TCP/IP.)

Networks that connect to the public Internet must obtain an official network ID from the InterNIC to guarantee IP network ID uniqueness. The InterNIC can be contacted via electronic mail at info@internic.net (1-800-444-4345 or, for Canada and overseas, 619-455-4600). Internet registration requests can be sent to hostmaster@internic.net. You can also use FTP to connect to is.internic.net, then log in as **anonymous**, and change to the /INFOSOURCE/FAQ directory.

Although private networks not connected to the Internet can choose to use their own network identifier, obtaining a valid network ID from InterNIC allows a private network to connect to the Internet in the future without reassigning addresses.

The Internet community has defined address classes to accommodate networks of varying sizes. Each network class can be discerned from the first octet of its IP address. The following table summarizes the relationship between the first octet of a given address and its network ID and host ID fields. It also identifies the total number of network IDs and host IDs for each address class that participates in the Internet addressing scheme. This sample uses w.x.y.z to designate the bytes of the IP address.

Class	w values*	Network ID	Host ID	Number of networks	Number of hosts per net
A	1-126	w	x.y.z	126	16,777,214
B	128-191	w.x	y.z	16,384	65,534
C	192-223	w.x.y	z	2,097,151	254

* Inclusive range for the first octet in the IP address. The address 127 is reserved for loopback testing and interprocess communication on the local computer; it is not a valid network address. Addresses 224 and above are reserved for special protocols (IGMP multicast and others) and cannot be used as host addresses.

A network node uses the network ID and host ID to determine which packets it should receive or ignore and to determine the scope of its transmissions (only nodes with the same network ID accept each other's IP-level broadcasts).

Because the sender's IP address is included in every outgoing IP packet, it is useful for the receiving node to derive the originating network ID and host ID from the IP address field. This is done by using subnet masks.

Subnet Masks

Subnet masks are 32-bit values that allow the recipient of IP packets to distinguish the network ID portion of the IP address from the host ID. Like an IP address, the value of a subnet mask is frequently represented in dotted decimal notation. Subnet masks are determined by assigning 1's to bits that belong to the network ID and 0's to the bits that belong to the host ID. Once the bits are in place, the 32-bit value is converted to dotted decimal notation, as shown in the following table.

Address	Bits for subnet mask	Default subnet mask
Class A	11111111 00000000 00000000 00000000	255.0.0.0
Class B	11111111 11111111 00000000 00000000	255.255.0.0
Class C	11111111 11111111 11111111 00000000	255.255.255.0

The result allows TCP/IP to determine the host and network IDs of the local workstation. For example: when the IP address is 102.54.94.97 and the subnet mask is 255.255.0.0, the network ID is 102.54 and the host ID is 94.97.

Although configuring a host with a subnet mask might seem redundant after examining Table 4.2 (since the class of a host is easily determined), subnet masks are also used to further segment an assigned network ID among several local networks.

For example, suppose a network is assigned the Class-B network address 144.100. This is one of over 16,000 Class-B addresses capable of serving more than 65,000 nodes. However, the worldwide corporate network to which this ID is assigned is composed of 12 international LANs with 75 to 100 nodes each. Instead of applying for 11 more network IDs, it is better to use subnetting to make more effective use of the assigned ID 144.100.

The third octet of the IP address can be used as a subnet ID, to define the subnet mask 255.255.255.0. This splits the Class-B address into 256 subnets: 144.100.0 through 144.100.255, each of which can have 254 nodes. (Host IDs 0 and 255 should not be assigned to a workstation; they are used as broadcast addresses, which are typically recognized by all workstations.) Any 12 of these network addresses could be assigned to the international LANs in this example. Within each LAN, each computer is assigned a unique host ID, and they all have the subnet mask 255.255.255.0.

The preceding example demonstrates a simple (and common) subnet scheme for Class-B addresses. Sometimes it is necessary to segment only portions of an octet, using only a few bits to specify subnet IDs (such as when subnets exceed 256 nodes). Each user should check with the local network administrator to determine the network's subnet policy and the correct subnet mask. For all systems on the local network, the subnet mask must be the same for that network ID.

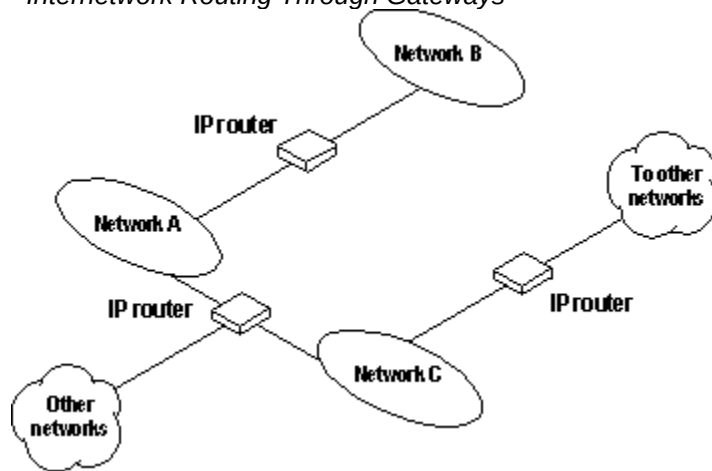
Important: All computers on a physical network should use the same subnet mask and network ID; otherwise, addressing and routing problems can occur.

Routing and IP Gateways

TCP/IP networks are connected by gateways (or routers), which have knowledge of the networks connected in the internetwork. Although each IP host can maintain static routes for specific destinations, the default gateway is usually used to find remote destinations. (The default gateway is needed only for computers that are part of an internetwork.)

When IP prepares to send a packet, it inserts the local (source) IP address and the destination address of the packet in the IP header and verifies that the network ID of the destination matches the network ID of the source. If they match, the packet is sent directly to the destination computer on the local network. If the network IDs do not match, the routing table is examined for static routes. If none are found, the packet is forwarded to the default gateway for delivery. Because the default gateway knows the network IDs of the other networks in the internetwork, it can forward the packet to other gateways until the packet is eventually delivered to a gateway connected to the specified destination. This process is known as routing.

Internetwork Routing Through Gateways



When individual IP subnets are connected to an internetwork, IP gateways are used to provide routing (packet delivery) between the networks. When a TCP/IP node attempts to communicate with a network that has a different network ID, a gateway (or a series of gateways) must forward the packet to the destination network. A gateway maintains routing tables that specify the direction (address of the next gateway) a packet should take to reach its destination.

Typically, gateways are IP routers, which are computers that run IP routing software and that have two or more network adapters; each adapter is connected to a different physical network. On networks that are not part of an internetwork, IP gateways are not required. If a network is part of an internetwork and a system does not specify a default gateway (or if the gateway computer is not operating properly), only communication beyond the local subnet is impaired. Users can add static routes by using the **route** command to specify a route for a particular system. Static routes always override the use of default gateways.

If the default gateway becomes unavailable, the computer cannot communicate outside its own subnet. Multiple default gateways can be assigned to prevent such a problem. When a computer is configured with multiple default gateways, retransmission problems result in the system trying the other routers in the configuration to ensure internetworking communications capabilities.

To configure multiple default gateways, you must provide an IP address for each gateway in the Advanced Microsoft TCP/IP Configuration dialog box.

Dynamic Host Configuration Protocol

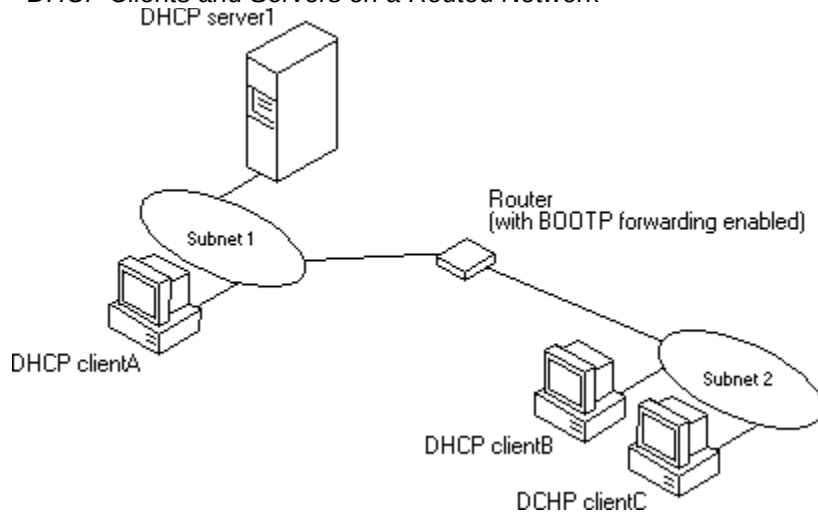
Assigning and maintaining IP address information can be an administrative burden for network administrators responsible for internetwork connections. Contributing to this burden is the problem that many users do not have the knowledge necessary to configure their own computers for internetworking and must therefore rely on their administrators.

The Dynamic Host Configuration Protocol (DHCP) was established to relieve this administrative burden. DHCP provides safe, reliable, and simple TCP/IP network configuration, ensures that address conflicts do not occur, and helps conserve the use of IP addresses through centralized management of address allocation. DHCP offers dynamic configuration of IP addresses for computers, plus the ability to associate duration leases with assigned addresses. DHCP also provides for internetwork configuration of certain diskless workstations.

As an example of how maintenance tasks are made easy with DHCP, the IP address is released automatically for a DHCP client computer that is removed from a subnet, and a new address for the new subnet is automatically assigned when that computer reconnects on another subnet. Neither the user nor the network administrator needs to intervene to supply new configuration information.

The following illustration shows an example of a DHCP server providing configuration information on two subnets. If, for example, ClientC is moved to Subnet 1, the DHCP server will automatically supply new TCP/IP configuration information the next time that ClientC is started.

DHCP Clients and Servers on a Routed Network



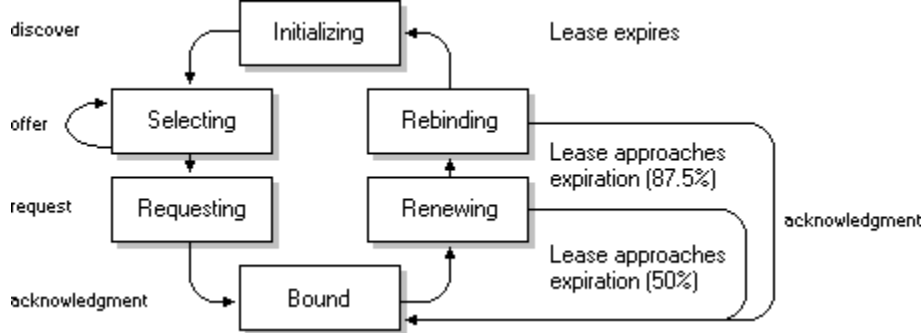
DHCP uses a client-server model. During system startup (the initializing state), a DHCP client computer sends a discover message that is broadcast to the local network and may be relayed to all DHCP servers on the private internetwork. Each DHCP server that receives the discover message responds with an offer message containing an IP address and valid configuration information for the client that sent the request.

The DHCP client collects the configuration offerings from the servers and enters a selecting state. At this time, one or more DHCP servers on the internetwork can have outstanding offers for this client. Each DHCP server reserves the address it offers, so that the same address cannot be offered to another client.

When the client enters the requesting state, it chooses one of the configurations and sends a request message that identifies the DHCP server for the selected configuration. Each of the DHCP servers that received the original discover message also receives the request message. Any servers that had outstanding offers for this client then return the offered addresses to their free address pools.

The selected DHCP server sends a DHCP acknowledgment message that contains the address first sent during the discovery stage, plus a valid lease for the address and the TCP/IP network configuration parameters for the client. After the client receives the acknowledgment, it enters a bound state and can now participate on the TCP/IP network and complete its system startup. Client computers that have local storage save the received address for use during subsequent system startup.

DHCP Client State Transition During System Startup



For more information about DHCP relaying, see the documentation for your router.

Name Resolution for Windows Networking

Configuring Windows for Workgroups with TCP/IP requires the computer name and IP address, which are unique identifiers for the computer on the network. On the local network, the computer name is the permanent name for the computer. For TCP/IP and the Internet, the computer name is the name plus a DNS domain name. The IP address is the unique address by which all other TCP/IP devices on the internetwork recognize that computer.

Computers use IP addresses to identify each other, but users usually find it easier to work with computer names. A mechanism must be available on a TCP/IP network to resolve names to IP addresses. To ensure that both name and address are unique, the Windows for Workgroups computer using TCP/IP registers its name and IP address on the network during startup. A Windows for Workgroups computer can use one or more of the following methods to ensure accurate name resolution in TCP/IP internetworks:

Windows Internet Name Service and Broadcast Name Resolution

Windows for Workgroups computers can use the Windows Internet Name Service (WINS) if one or more WINS servers are available containing a dynamic database that maps IP addresses to computer names. WINS can be used in conjunction with broadcast name resolution for an internetwork where other name resolution methods are inadequate. WINS is a NetBIOS over TCP/IP mode of operation defined in Requests for Comments (RFCs) 1001 and 1002 as p-node.

Windows for Workgroups computers can also use broadcast name resolution, as defined by the b-node protocol in RFCs 1001 and 1002. This method relies on a computer making IP-level broadcasts to register its name by announcing it on the network. If another system in the broadcast area challenges that name, the broadcasting computer claims the name. The computer is responsible for challenging attempts to register a duplicate name and for responding to name queries for its registered name.

Domain Name System Addressing

The Domain Name System (DNS) provides a way to look up name mappings when connecting a computer to foreign hosts using NetBIOS over TCP/IP or Windows Sockets applications such as FTP. DNS is a distributed database designed to relieve the traffic problems that arose with the explosive growth of the Internet in the early 1980s.

Name Resolution with Host Files

The HOSTS and LMHOSTS files on a local computer contain lists of known IP addresses mapped with corresponding computer names. This method of name resolution was the predecessor to DNS and is still used in Windows for Workgroups for small-scale networks or remote subnets where WINS is not available. The computer checks its LMHOSTS file to find an IP address for a particular remote computer using Microsoft networking or HOSTS when using Windows Sockets applications.

See Also

[Understanding NetBIOS over TCP/IP](#)

Understanding NetBIOS over TCP/IP

NetBIOS over TCP/IP is the session-layer network service that performs name-to-IP address mapping for name resolution. This section describes the various implementations of NetBIOS over TCP/IP, as defined in RFCs 1001 and 1002 to specify how NetBIOS should be implemented over TCP/IP.

These implementations determine how network resources are identified and accessed. The two most important aspects of the related naming activities are registration and resolution. Registration is the process used to acquire a unique name for each node (computer system) on the network. A computer typically registers itself when it starts. Resolution is the process used to determine the specific address for a computer name.

The NetBIOS over TCP/IP implementations include the following:



b-node

This implementation uses broadcasts to resolve names.



p-node

This implementation uses point-to-point communications with a name server to resolve names.



m-node

This implementation uses b-node first (broadcasts), then p-node (name queries) if the broadcast fails to resolve a name.



h-node

This implementation uses p-node first for name queries, then b-node if the name service is unavailable.



BNode with LMHOSTS and combinations

For DHCP users on a Windows network, the node type is assigned by the DHCP Manager. When WINS servers are in place on the network, NetBIOS over TCP/IP resolves names on a client computer by communicating with the WINS server. When WINS servers are not in place, NetBIOS over TCP/IP uses b-node broadcasts to resolve names. In Microsoft TCP/IP-32, the VNBT.386 module provides the NetBIOS over TCP/IP functionality that supports name registration and resolution implementations.

See Also

[TCP/IP and Windows Networking](#)

B-Node

The b-node implementation uses broadcasts for name registration and resolution. That is, if NT_PC1 wants to communicate with NT_PC2, it will broadcast to all machines that it is looking for NT_PC2 and then wait a specified time for NT_PC2 to respond. This implementation has two major problems:



In a large environment, it loads the network with broadcasts.



Routers do not forward broadcasts, so computers that are on the opposite side of a router never receive the requests.

P-Node

The p-node implementation addresses the issues that b-node does not solve. In a p-node environment, computers neither create nor respond to broadcasts. When they are first started, all computers register themselves with a WINS server, which is a NetBIOS Name Server (NBNS) as described in RFCs 1001/1002. The WINS server is responsible for knowing computer names and addresses and for ensuring that no duplicate names exist on the network. All computers must be configured to know the address of the WINS server.

In this environment, when NT_PC1 wants to communicate with NT_PC2, it queries the WINS server for the address of NT_PC2. When NT_PC1 gets the appropriate address from the WINS server, it goes directly to NT_PC2 without broadcasting. Because these are all direct connections, this implementation avoids loading the network with broadcasts. Because broadcasts are not used and because the address is received directly, computers can span routers.

The most significant problems with p-node are the following:



All computers must be configured to know the address of the WINS server



If for any reason the WINS server is down, computers that rely on the WINS server to resolve addresses cannot get to any other systems on the network.

M-Node

The m-node implementation was created primarily to solve the problems associated with b-node and p-node by combining the two. In an m-node environment, a computer first attempts registration and resolution using b-node. If that is unsuccessful, it then switches to p-node and tries again. Because this uses b-node first, it does not solve the problem of generating broadcast traffic on the network. However, this implementation can cross routers. Also, because b-node is always tried first, computers on the same side of a router continue to operate as usual if the WINS server is down.

M-node is rarely used because it still has one problem: it does not reduce broadcasts.

H-Node

Like m-node, the h-node implementation, which is currently in RFC draft form, is a combination of b-node and p-node that uses broadcasts as a last effort. Because p-node is used first, no broadcasts are generated if the WINS server is running and computers can span routers. If the WINS server is down, b-node is used, so computers on the same side of a router continue to operate as usual. For name registration, this implementation uses m-node (broadcast, then name query).

The h-node implementation does more than change the order for using b-node and p-node. If the WINS server is down so that local broadcasts (b-node) must be used, the computer will continue to poll the WINS server. As soon as the WINS server can be reached again, the system switches back to p-node. Also, h-node can be optionally configured to use LMHOSTS before resorting to b-node.

The h-node implementation solves the most significant problems associated with broadcasts and operating in a routed environment. For Microsoft TCP/IP-32 users who configure TCP/IP manually, this is the implementation used by default, unless the user does not check the Query WINS option when configuring TCP/IP.

B-Node with LMHOSTS and Combinations

Another b-node variation is also used in Microsoft networks to span routers without a WINS server and p-node implementation. Modified b-node uses a list of computers and addresses in a list stored in an LMHOSTS file. If a b-node attempt fails, the system looks in LMHOSTS to find a name and then uses the associated address to cross the router. However, each computer must have this list, which creates an administrative burden in maintaining and distributing the list. In Windows for Workgroups, some extensions have been added to this file to make it easier to manage, but modified b-node is not an ideal solution.

Some sites may need to use both b-node and p-node implementations at the same site. Although this configuration can work, administrators must exercise extreme caution in doing so, using it only for transition situations. Because p-node hosts disregard broadcasts and b-node hosts rely on broadcasts for name resolution, the two hosts can potentially be configured with the same NetBIOS name, leading to unpredictable results.






Windows Internet Name Service and Broadcast Name Resolution

In a TCP/IP network, computer names can be resolved using any combination of WINS, name query broadcasts, the LMHOSTS file, and DNS. WINS provides a distributed database for registering and querying dynamic computer name-to-IP address mappings in a routed network environment. If you are administering a routed network, WINS is your best choice for name resolution, because it is designed to solve the problems that occur with name resolution in more complex internetworks.

WINS reduces the use of local broadcasts for name resolution and allows users to easily locate systems on remote networks. Furthermore, when dynamic addressing through DHCP results in new IP addresses for computers that move between subnets, the changes are automatically updated in the WINS database. Neither the user nor the network administrator needs to make manual accommodations for name resolution in such a case.

The WINS protocol is based on and is compatible with the p-node protocol defined in RFCs 1001 and 1002, so it is interoperable with any other implementations of these RFCs.

The following topics provide an overview of how WINS and name query broadcasts provide name resolution on Windows networks.

-  [WINS in a Routed Environment](#)
-  [Name Resolution with WINS and Broadcasts](#)
-  [WINS Name Registration](#)
-  [WINS Name Release](#)
-  [WINS Name Renewal](#)

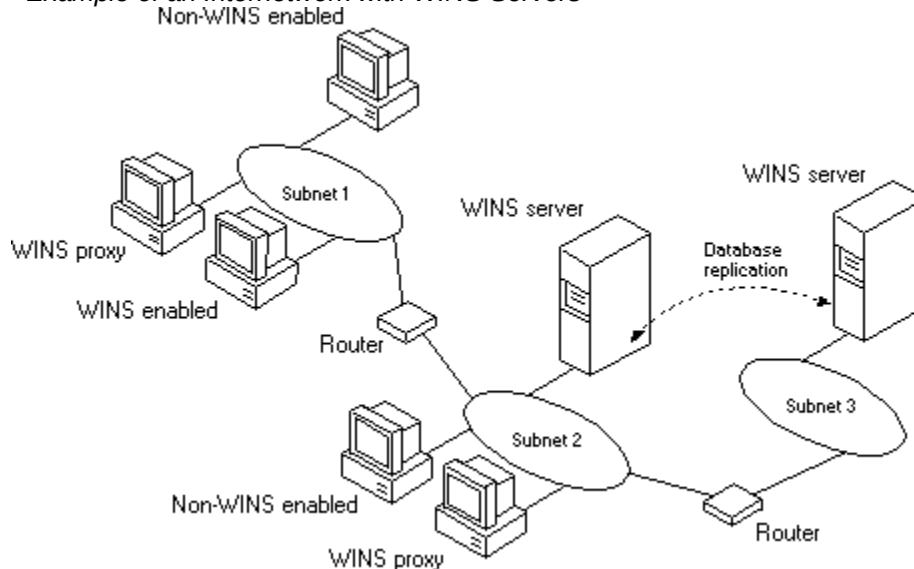
WINS in a Routed Environment

WINS consists of two components: the WINS server, which handles name queries and registrations, and the client software, which queries the server for computer name resolution.

Windows networking clients (WINS-enabled Windows NT or Windows for Workgroups 3.11 computers) can use WINS directly. Non-WINS computers on the internetwork that are b-node compatible, as described in RFCs 1001 and 1002 can access WINS through proxies, which are network WINS-enabled computers that listen to name query broadcasts and then respond for names that are not on the local subnet.

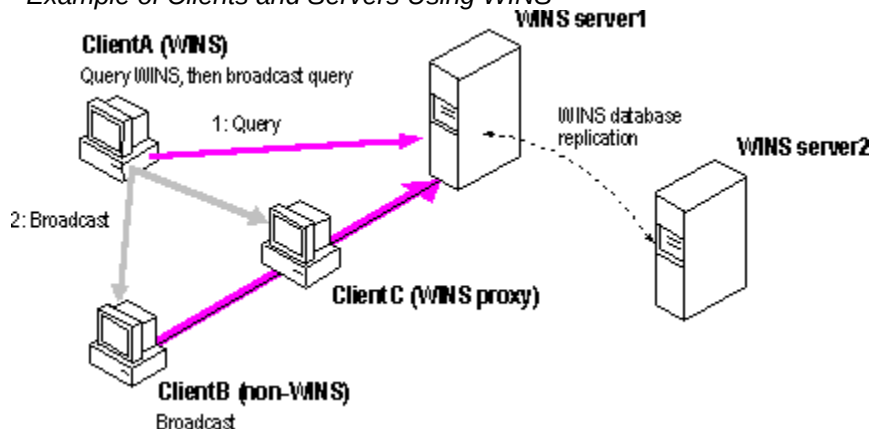
The following illustration shows a small internetwork, with two local area networks connected by a router. One of the subnets includes a WINS name server, which can be used by clients on both subnets. WINS-enabled computers, including proxies, access the WINS server directly, and the computers using broadcasts access the WINS server through proxies. (However, proxies only pass name query packets, not name registrations.)

Example of an Internetwork with WINS Servers



The proxy communicates with the WINS server to resolve names (rather than maintaining its own database) and then caches the names for a certain time. The proxy can serve as an intermediary, by either communicating with the WINS server or supplying a name-to-IP address mapping from its cache. The following illustration shows the relationships among WINS servers and clients, including proxies for non-WINS computers.

Example of Clients and Servers Using WINS



In the above illustration, ClientA can resolve names by first querying the WINS server and then, if that fails, using broadcast name queries. ClientB, which is not WINS-enabled, can only resolve names using broadcast name queries, but when ClientC receives the broadcast, it forwards the request to the WINS server and returns the name to ClientB.

If the Windows for Workgroups computer is also DHCP-enabled, the computer will usually be configured with WINS server information. To enable WINS name resolution for a computer that does not use DHCP, check the Query WINS option in the Microsoft TCP/IP Configuration dialog box. To designate a proxy, check the Enable WINS Proxy Agent option in the Advanced Microsoft TCP/IP Configuration dialog box.

Name Resolution with WINS and Broadcasts

Name query requests are generated when a user (or process) requests a connection with a remote computer on a network. For example, typing `net use x: \\treycorp\resources` at the command prompt causes a name query request packet to be broadcast for the computer named treycorp. Because TCP/IP recognizes IP addresses and Windows networking services use only names, the computer on the internetwork must resolve name-to-IP addressing for the computer it wants to communicate with.

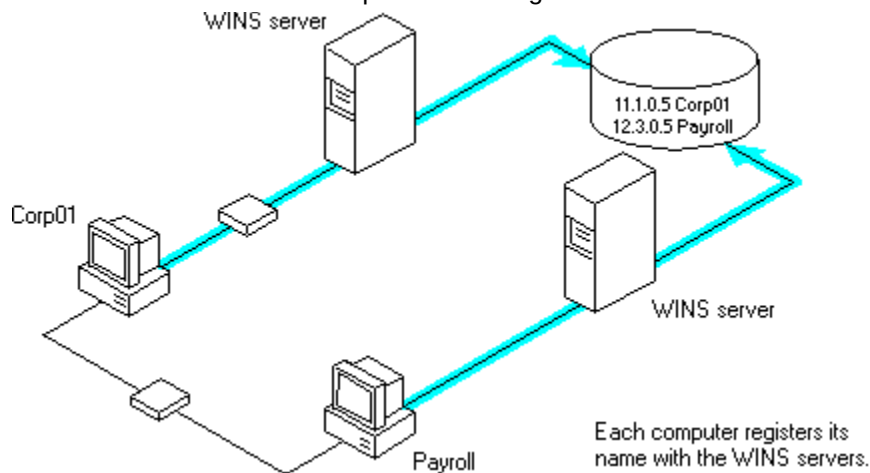
With WINS servers in place on the internetwork, names are resolved using two basic methods, depending on whether WINS resolution is available and enabled on the particular computer.

Note: Whatever name resolution method is used, the process is transparent to the user after the system is configured.

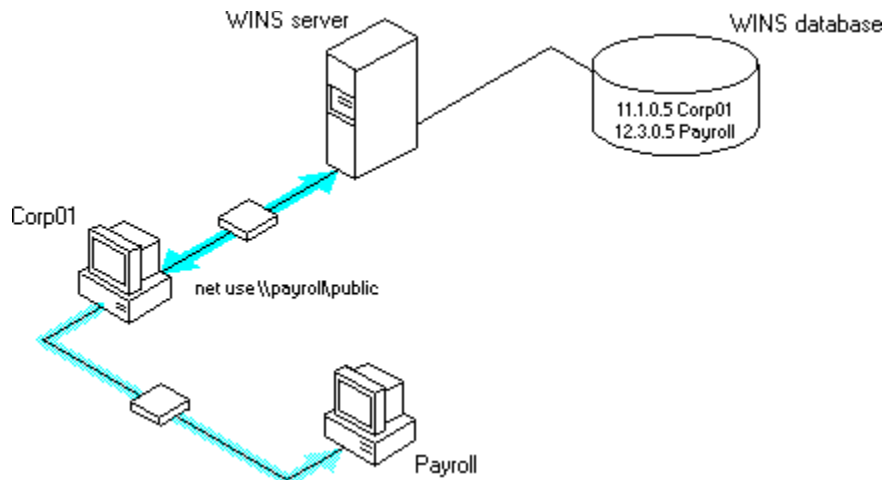
If WINS is not enabled The computer registers its name by broadcasting name registration request packets to the local subnet via UDP datagrams. To find a particular computer, the non-WINS computer broadcasts name query request packets on the local subnet, although this broadcast cannot be passed on through IP routers. If local name resolution fails, the local LMHOSTS file is consulted. These processes are followed whether the computer is a network server, a workstation, or another device.

If WINS is enabled The computer first queries the WINS server, and if that does not succeed, it broadcasts its name registration and query requests via UDP datagrams. Name resolution involves the following series of steps:

1. During TCP/IP configuration, the computer's name is registered with the WINS server, and the IP address of the WINS server is stored locally so the WINS server can be found on the internetwork. The WINS database is replicated among all WINS servers on the internetwork.



2. A name query request is sent first to the WINS server, including requests from remote clients that are routed through an IP router. This request is a UDP datagram. If the name is found in the WINS database, the client can establish a session based on the address mapping received from WINS.



3. If querying the WINS server does not succeed, the system broadcasts name query request packets (if broadcast resolution is enabled on that system), in the same manner as a non-WINS-enabled computer.

WINS servers accept and respond to UDP and TCP name queries. Any name-to-IP address mapping registered with a WINS server can be provided reliably as a response to a name query. However, a mapping in the database does not ensure that the related device is currently running, only that a computer claimed the particular IP address and that it is a currently valid mapping. Other Windows networking services determine whether the computer is actually running or whether such a computer is present on a particular network node.

WINS Name Registration

Name registration ensures that the computer's name and IP address are unique for each device.

If WINS is enabled The name registration request is broadcast locally to check for conflicts and then sent directly to the WINS server to be added to the database if there is no conflict. A WINS server accepts or rejects a computer name registration depending on the current contents of its database. If the database contains a different address for that name, WINS challenges the current entry to determine whether the originally registering device still claims the name. If another device is using that name, WINS rejects the new name registration request. Otherwise, WINS accepts the entry and adds it to its local database together with a timestamp, an incremental unique version number, and other information.

If WINS is not enabled For a non-WINS computer to register its name, a NetBIOS name registration request packet is broadcast to the local network, stating its computer name and IP address. Any device on the network that previously claimed that NetBIOS name challenges the name registration with a negative name registration response, resulting in an error. If the registration request is not challenged within a specific time period, the computer adopts that name and address.

Once a non-WINS computer has claimed a name, it must challenge duplicate name registration attempts and respond positively to name queries issued on its registered name by sending a positive name query response. This response contains the IP address of the computer so that the two systems can establish a session.

WINS Name Release

When a computer finishes with a particular name (such as when the Server service is stopped), it no longer challenges other registration requests for the name. This is referred to as releasing a name.

If WINS is enabled Whenever a computer is shut down properly, it releases its name to the WINS server, which marks the related database entry as released. If the entry remains released for a certain period of time, the WINS server marks it as extinct, and the version number is updated so that the database changes will be propagated among the WINS servers. Extinct entries remain in the database for a designated period of time to ensure that the change is propagated to all WINS servers.

If a name is marked released at a WINS server and a new registration arrives using that name but a different address, the WINS server can immediately give that name to the requesting client because it knows that the old client is no longer using that name. (This might happen, for example, when a DHCP-enabled laptop changes subnets.) If that computer released its name during an orderly shutdown, the WINS server will not challenge the name. If the computer restarts because of a system reset, the name registration with a new address will cause the WINS server to challenge the registration, but the challenge will fail, and the registration will succeed, because the computer no longer has the old address.

If WINS is not enabled When a non-WINS computer releases a name, a broadcast is made to allow any systems on the network that might have cached the name to remove it. Upon receiving name query packets specifying the deleted name, the computer simply ignores the request, allowing other computers on the network to acquire the name that it has released.

For non-WINS computers to be accessible from other subnets, their names must be statically added to the WINS database or in the LMHOSTS file(s) in the remote system(s), because they will respond only to name queries that originate on their local subnet.

WINS Name Renewal

A renewal is a timed reregistration of a computer's name with the WINS server. When the WINS server registers a name, it returns a renewal interval for the name, and the client must reregister within that time or the WINS server will mark the name as released and eventually as extinct and available for use. A request for name renewal is treated the same as a new name registration.

Renewal provides registration reliability through periodic reregistering of names with the WINS servers.

Domain Name System Addressing

The Domain Name System (DNS) is a distributed database, providing a hierarchical naming system for identifying hosts on the Internet. DNS was developed to solve the problems that arose when the number of hosts on the Internet grew dramatically in the early 1980s. The specifications for DNS are defined in RFCs 1034 and 1035.

The DNS database is a tree structure called the domain name space, where each node or domain is named and can contain subdomains. The domain name identifies the domain's position in the database in relation to its parent domain, with a period (.) separating each part of the names for the network nodes of the DNS domain.

The root of the DNS database is managed by the Internet Network Information Center. The top-level domains were assigned organizationally and by country. These domain names follow the international standard ISO 3166. Two-letter abbreviations are used for countries, and various abbreviations are reserved for use by organizations, as shown in the following example.

DNS domain name abbreviation	Type of organization
com	Commercial (for example, microsoft.com)
edu	Educational (for example, mit.edu for Massachusetts Institute of Technology)
gov	Government (for example, nsf.gov for the National Science Foundation)
org	Noncommercial organizations (for example, fidonet.org for FidoNet)
net	Networking organizations (for example, nsf.net for NSFNET)

Each DNS domain is administered by different organizations, which usually break their domains into subdomains and assign administration of the subdomains to other organizations. Each domain has a unique name, and each of the subdomains have unique names within their domains. The label for each network node is a name of up to 63 characters. The fully qualified domain name (FQDN), which includes the names of all network nodes leading back to the root, is unique for each host on the Internet. A particular DNS name could be similar to the following, for a commercial host:

`accounting.trey.com`

DNS uses a client-server model, where the DNS servers contain information about a portion of the DNS database and make this information available to clients, called resolvers, that query the name server across the network. DNS name servers are programs that store information about parts of the domain name space called zones. The administrator for a domain sets up name servers that contain the database files with all the resource records describing all hosts in their zones. DNS resolvers are clients that are trying to use name servers to gain information about the domain name space.

Windows for Workgroups includes the DNS resolver functionality used by NetBIOS over TCP/IP and by Windows Sockets connectivity applications such as **ftp** and **telnet** that query the name server and interpret the responses.

The key task for DNS is to present friendly names for users and then resolve those name to IP addresses, as required by the internetwork. Name resolution is provided through DNS by the name servers, which interpret the information in an FQDN to find its specific address. If a local name server doesn't contain the data requested in a query, it sends back names and addresses of other name servers that could contain the information. The resolver then queries the other name servers until it finds the specific name and address it needs. This process is made faster because name servers

continuously cache the information learned about the domain name space as the result of queries.

All the resolver software necessary for using DNS on the Internet is installed with Microsoft TCP/IP-32. To use DNS for name resolution, and to enable WINS name resolution for FQDNs, you specify options in the Microsoft TCP/IP Connectivity Configuration dialog box.

See Also

[Configuring TCP/IP to Use DNS](#)

Name Resolution with Host Files

For computers located on remote subnets where WINS is not used, the HOSTS and LMHOSTS files provide mappings for names to IP addresses. This is the name resolution method used on internetworks before DNS and WINS were developed. The HOSTS file can be used as a local DNS equivalent. The LMHOSTS file can be used as a local WINS equivalent. Each of these files is also known as a host table.

Sample versions of LMHOSTS and HOSTS files are added to the \WINDOWS directory when you install Microsoft TCP/IP. These files can be edited using any ASCII editor, such as Notepad, which is part of Windows for Workgroups.

The LMHOSTS file is read when WINS or broadcast name resolution fails, and resolved entries are stored in a system cache for later access.

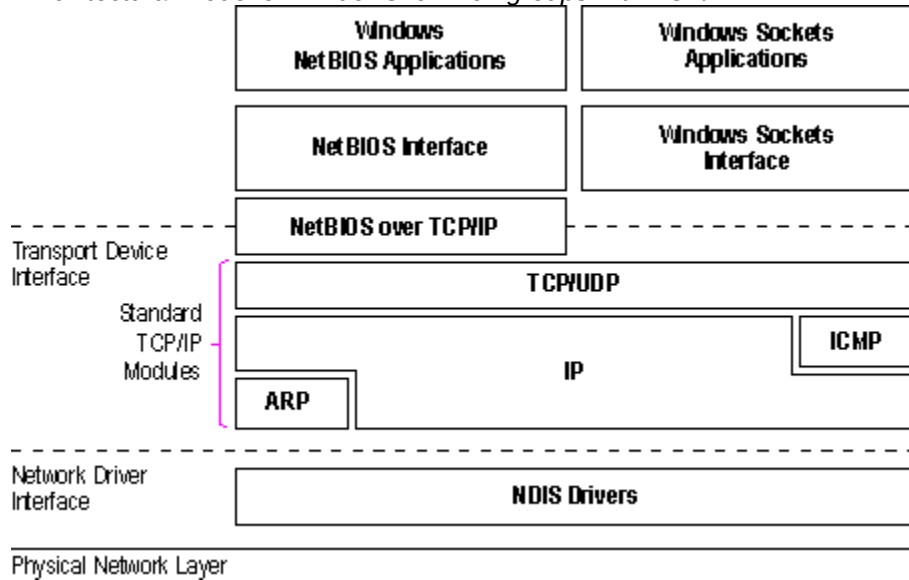
See Also

[Setting Up LMHOSTS.](#)

TCP/IP and Windows Networking

The architecture of the Microsoft Windows for Workgroups operating system with integrated networking is protocol-independent. This architecture, illustrated in the following figure, provides Windows for Workgroups file, print, and other services over any network protocol that uses the network basic input/output system (NetBIOS) standard. NetBIOS-compliant protocols package network requests for applications in their respective formats and send the requests to the appropriate network adapter via the network device interface specification (NDIS) interface. The NDIS specification allows multiple network protocols to reside over a wide variety of network adapters and media types.

Architectural Model of Windows for Workgroups with TCP/IP



Under the Windows for Workgroups transport-independent architecture, TCP/IP is a protocol family that can be used to offer Windows networking capabilities. The TCP/IP protocol gives Windows NT, Windows for Workgroups, and LAN Manager computers transparent access to each other and allows communication with non-Microsoft systems in the enterprise network.

Windows Sockets

Microsoft TCP/IP for Windows for Workgroups also includes support for Windows Sockets. The Windows Sockets API is a networking API used by programmers creating applications for both the Microsoft Windows NT and Windows for Workgroups operating systems.

Windows Sockets is an open standard that is part of the Microsoft Windows Open System Architecture (WOSA) initiative. It is a public specification based on Berkeley UNIX sockets, which means that UNIX applications can be quickly ported to Microsoft Windows and Windows NT. Windows Sockets provides a single standard programming interface supported by all the major vendors implementing Microsoft TCP/IP for Windows networking. The Microsoft TCP/IP utilities use Windows Sockets, as do 16-bit Windows-based applications created under the Windows Sockets standard. Most TCP/IP users will use programs that comply with the Windows Sockets standard, such as **ftp** or **telnet** or third-party applications.

Typical Windows Sockets applications include graphic connectivity utilities, terminal emulation software, Simple Mail Transfer Protocol (SMTP) and electronic mail clients, network printing utilities, SQL client applications, and corporate client-server applications.

The Windows Sockets standard allows a developer to create an application with a single common interface and a single executable that can run over many of the TCP/IP implementations provided by vendors. The goals for Windows Sockets are the following:



Provide a familiar networking API to programmers using Windows NT, Windows for Workgroups, or UNIX



Offer binary compatibility between vendors for heterogeneous Windows-based TCP/IP stacks and utilities



Support both connection-oriented and connectionless protocols

If you are interested in developing a Windows Sockets application, specifications for Windows Sockets are available on the Internet from <ftp.microsoft.com>, on CompuServe® in the MSL library, and in the Microsoft Win32® Software Developers Kit.

To get the Windows Sockets specification via anonymous FTP

1. Start **ftp** and connect to **microsoft.com** (or **ftp 198.105.232.1**).
2. Log on as **anonymous**.
3. Type your electronic mail address for the *password*.
4. Type **cd ladvsys\wwinsock\spec11** and press ENTER.
5. Use the **dir** command to see the list of available file types. If you want binary data such as in the Microsoft Word version, type **bin** and press ENTER.
6. Determine the file with the format you want [for example, ASCII (.TXT), PostScript® (.PS), or Microsoft Word (.DOC)], and then type **get winsock.ext** where *ext* is the format that you want, such as **winsock.doc** for the Microsoft Word version.

To get the Windows Sockets specification from CompuServe

1. Type **go msl** and press ENTER.
2. Browse using the keywords **windows sockets**.
3. Choose the file with the format you want [ASCII (.TXT), PostScript (.PS), or Microsoft Word for Windows (.DOC)], and then type **get winsock.ext**

There is also an electronic mailing list designed for discussion of Windows Sockets programming.

To subscribe to the Windows Sockets mailing list



Send electronic mail to listserv@sunsite.unc.edu with a message body that contains **subscribe winsock** *user's-email-address*.

You can use the same procedure to subscribe to two mailing lists called **winsock-hackers** and **winsock-users**.

Setting Up LMHOSTS

The LMHOSTS file is commonly used to locate remote computers for Windows networking file, print, and remote procedure services and for domain services such as logons, browsing, replication, and so on.



[Editing the LMHOSTS file](#)



[Using LMHOSTS with Dynamic Name Resolution](#)

See Also

[Name Resolution with Host Files](#)

Editing the LMHOSTS File

The LMHOSTS file used in Windows networking contains mappings of IP addresses to Windows networking computer names (which are NetBIOS names). Microsoft LAN Manager 2.x TCP/IP LMHOSTS files are compatible with Microsoft TCP/IP-32.

You can use Notepad or any other text editor to edit the sample LMHOSTS.SAM file that is automatically installed in the \WINDOWS directory.

The following topics provide some basic rules and guidelines for LMHOSTS.







[Rules for LMHOSTS](#)



[Guidelines for LMHOSTS](#)

Rules for LMHOSTS

The following rules apply for entries in LMHOSTS:

-  Each entry should be placed on a separate line.
-  The IP address should begin in the first column, followed by the corresponding computer name.
-  The address and the computer name should be separated by at least one space or tab.
-  The # character is usually used to mark the start of a comment. However, it can also designate special keywords.

The keywords listed in the following table can be used in LMHOSTS.

#PRE

Added after an entry to cause that entry to be preloaded into the name cache. By default, entries are not preloaded into the name cache but are parsed only after WINS and name query broadcasts fail to resolve a name. #PRE must be appended for entries that also appear in #INCLUDE statements; otherwise, the entry in #INCLUDE is ignored.

#DOM:<domain>

Added after an entry to associate that entry with the domain specified by <domain>. This keyword affects how the Browser and Logon services behave in TCP/IP environments. To preload a #DOM entry, you must also add the #PRE keyword to the line.

#INCLUDE <filename>

Forces the system to seek the specified <filename> and parse it as if it were local. Specifying a Uniform Naming Convention (UNC) <filename> allows you to use a centralized LMHOSTS file on a server. If the server is located outside of the local broadcast area, you must add a mapping for the server before its entry in the #INCLUDE section, and also append #PRE to ensure the server name is preloaded.

#BEGIN_ALTERNATE

Marks the beginning of a group of multiple #INCLUDE statements. Any single successful INCLUDE causes the group to succeed.

#END_ALTERNATE

Used to mark the end of an #INCLUDE grouping.

\0xnn

Support for nonprinting characters in NetBIOS names. Enclose the NetBIOS name in double quotation marks and use \0xnn hexadecimal notation to specify a hexadecimal value for the nonprinting character. This allows custom applications that use special names to function properly in routed topologies. However, LAN Manager TCP/IP does not recognize the hexadecimal format, so you surrender backward compatibility if you use this feature.

Note that the hexadecimal notation applies only to one character in the name. The name should be padded with blanks so the special character is last in the string (character 16).

The following example shows how all of these keywords are used:

```
102.54.94.98      localsvr      #PRE
102.54.94.97      trey          #PRE #DOM:networking #net group's domain controller
102.54.94.102     "appname     \0x14"          #special app server
102.54.94.123     popular      #PRE          #source server
```

```
#BEGIN_ALTERNATE
```

```
#INCLUDE \\localsrv\public\lmhosts
```

```
#INCLUDE \\trey\public\lmhosts
#END_ALTERNATE
```

In the above example:



The servers named **localsvr** and **trey** are preloaded so they can be used later in an #INCLUDE statement in a centrally maintained LMHOSTS file.



The server named "**appname** **\0x14**" contains a special character in its name (including the blanks), so its name is enclosed in double quotation marks.



The server named **popular** is preloaded.

Guidelines for LMHOSTS

When you use a host table file, be sure to keep it up to date and organized. Follow these guidelines:



Update the host table file whenever a computer is changed or removed from the network.



Because host table files are searched one line at a time from the beginning, list remote computers in priority order, with the ones used most often at the top of the file. This increases the speed of searches for the entries used most often.



Use `#PRE` statements to preload popular entries into the local computer's name cache and to preload servers that are included with `#INCLUDE` statements.



The `#PRE` entries should be left for the end of the file, because these are preloaded into the cache at system startup time and are not accessed later.



Any comment lines add to the parsing time, because each line is processed individually.

Using LMHOSTS with Dynamic Name Resolution

The broadcast name resolution method used by Windows networking computers provides a simple, dynamic mechanism for locating resources by name on a TCP/IP network.

If WINS servers are in place on an internetwork, users do not have to rely on broadcast queries for name resolution, since WINS is the preferred method for name resolution. LMHOSTS can be used with or without WINS. You will want to use LMHOSTS for smaller networks or to find hosts on remote networks that are not part of the WINS database (since name query requests are not broadcast beyond the local subnet).

Because broadcast name resolution relies on IP-level broadcasts to locate resources, unwanted effects can occur in routed IP topologies. In particular, resources located on remote subnets do not receive name query requests, because routers do not pass IP-level broadcasts. For this reason, Windows networking allows you to manually provide computer name and IP address mappings for remote resources.

These topics describe how the LMHOSTS file can be used to enhance Windows networking in routed environments. The following topics are included:



Specifying remote servers in LMHOSTS



Designating domain controllers using #DOM



Using centralized LMHOSTS files

Specifying Remote Servers in LMHOSTS

Computer names can be resolved outside the local broadcast area if computer name and IP address mappings are specified in the LMHOSTS file. For example, suppose computer ClientA wants to connect to computer ServerB, which is outside of its IP broadcast area. Both computers are configured with Microsoft TCP/IP.

Under a strict b-node broadcast protocol, ClientA's name query request for ServerB would fail (by timing out), because ServerB is located on a remote subnet and does not respond to ClientA's broadcast requests. So an alternate method is provided for name resolution. The system maintains a limited cache of computer name and IP address mappings, which is initialized at system startup. When a workstation needs to resolve a name, the cache is examined first and, if there is no match in the cache, the system uses broadcast name resolution. If this method fails, the LMHOSTS file is used. If this last method fails, the name is unresolved, and an error message appears.

This strategy allows the LMHOSTS file to contain a large number of mappings without requiring a large chunk of static memory to maintain an infrequently used cache. At system startup, the name cache is preloaded only with entries from LMHOSTS that are tagged with the #PRE keyword. For example, the LMHOSTS file could contain the following:

```
102.54.94.91      accounting      #accounting server
102.54.94.94      payroll        #payroll server
102.54.94.97      stockquote    #PRE #stock quote server
102.54.94.102    printqueue    #print server in Bldg 10
```

In this example, the server named **stockquote** is preloaded into the name cache because it is tagged with the #PRE keyword. Entries in the LMHOSTS file can represent Windows NT Workstation computers, Windows NT Server computers, LAN Manager servers, or Windows for Workgroups computers running Microsoft TCP/IP-32. There is no need to distinguish among different platforms in LMHOSTS.

Note: The tag #PRE allows backward compatibility with LAN Manager 2.x LMHOSTS files and offers added flexibility in Windows networking. Under LAN Manager, the # character identifies a comment, so all characters thereafter are ignored. But #PRE is a valid tag for Windows networking.

In the above example, the servers named **accounting**, **payroll**, and **printqueue** would be resolved only after the cache entries failed to match and after broadcast protocol failed to locate them. After nonpreloaded entries are resolved, their mappings are cached for a period of time for reuse.

The preload name cache is limited to 100 entries by default. This limit affects only entries marked with #PRE that cannot fit in the cache. If you specify more than 100 entries, only the first 100 #PRE entries will be preloaded. Any additional #PRE entries will be ignored at startup but will be resolved when the system parses the LMHOSTS file after dynamic resolution fails.

Finally, you can reprime the name cache by using the **nbtstat -R** command to purge and reload the name cache, reread the LMHOSTS file, and insert entries tagged with the #PRE keyword. The **nbtstat** command allows for removing or correcting preloaded entries that may have been mistyped or any names cached by successful broadcast resolution.

Designating Domain Controllers Using #DOM

The most common use of LMHOSTS is for locating remote servers for file and print services. LMHOSTS can also be used to find domain controllers running TCP/IP in routed environments. Windows NT domain controllers maintain the user account security database and manage other network-related services. Because large Windows NT domains can span multiple IP subnets, it is possible that routers could separate the domain controllers from one another or separate computers from domain controllers.

The #DOM keyword can be used in LMHOSTS files to distinguish a Windows NT domain controller from a Windows NT workstation, a LAN Manager server, or a Windows for Workgroups system. To use the #DOM tag, follow the name and IP address mapping in LMHOSTS with the #DOM keyword, a colon, and the domain in which the domain controller participates. For example:

```
102.54.94.97    treydc    #DOM:treycorp    #The treycorp domain controller
```

Note: For domain controller entries that will be accessed frequently, use the #PRE keyword after the #DOM tag (order is important if LAN Manager servers also use this LMHOSTS file).

Using the #DOM keyword to designate domain controllers adds entries to a special *internet group name cache* that is used to limit internetwork distribution of requests intended for the local domain controller. When domain controller activity such as a logon request occurs, the request is sent on the special internet group name. In the local IP-broadcast area, the request is sent only once and picked up by any local domain controllers. However, if you use #DOM to specify domain controllers in the LMHOSTS file, Microsoft TCP/IP-32 uses datagrams to also forward the request to domain controllers located on remote subnets.

Examples of such domain controller activities include domain controller pulses (used for account database synchronization), logon authentication, password changes, master browser list synchronization, and other domain management activities.

For domains that span subnets, LMHOSTS files can be used to map important members of the domain using #DOM. The following lists some guidelines for doing this.



For each local LMHOSTS file on a Windows networking computer that is a member in a domain, there should be #DOM entries for all domain controllers in the domain that are located on remote subnets. This ensures that logon authentication, password changes, browsing, and so on all work properly for the local domain. These are the minimum entries necessary to allow a computer to participate in a Windows networking internetwork.



For local LMHOSTS files on all servers that can be domain controllers, there should be mappings for the primary domain controller's name and IP address, plus mappings for all other backup servers. This ensures that promoting a server to domain controller status does not affect the ability to offer all services to members of the domain.



If trust relationships exist between domains, all domain controllers for all trusted domains should also be listed in the local LMHOSTS file.



For domains that you want to browse from your local domain, the local LMHOSTS files should contain at least the name and IP address mapping for the domain controller in the remote domain. Again, backup servers should also be included so that promotion to domain controller does not impair the ability to browse remote domains.

For small- to medium-sized networks with fewer than 20 domains, a single common LMHOSTS file usually satisfies all workstations and servers on the internetwork. To achieve this, systems should use the replicator service to maintain synchronized local copies of the global LMHOSTS or use centralized LMHOSTS files, as described in [Using Centralized LMHOSTS Files](#).

Using Centralized LMHOSTS Files

With Microsoft TCP/IP, you can include other LMHOSTS files from local and remote computers. The primary LMHOSTS file is always located in the \WINNT\SYSTEM32\DRIVERS\ETC directory on the local computers. Most networks will also have an LMHOSTS file maintained by the network administrator, so administrators should maintain one or more global LMHOSTS files that users can rely on. This is done using #INCLUDE statements rather than copying the global file locally. Then use the replicator service to distribute multiple copies of the global file(s) to multiple servers for reliable access.

To provide a redundant list of servers maintaining copies of the same LMHOSTS file, use the #BEGIN_ALTERNATE and #END_ALTERNATE keywords. This is known as a block inclusion, which allows multiple servers to be searched for a valid copy of a specific file. The following example shows the use of the #INCLUDE and #_ALTERNATE keywords to include a local LMHOSTS file (in the C:\PRIVATE directory):

```
102.54.94.97      treydc    #DOM:treycorp    #treycorp domain controller
102.54.94.99      treybdc   #DOM:treycorp    #backup DC in treycorp domain

#INCLUDE          c:\private\lmhosts          #include a local lmhosts

#BEGIN_ALTERNATE
#INCLUDE          \\treydc\public\lmhosts      #source for global file
#INCLUDE          \\treybdc\public\lmhosts     #backup source
#INCLUDE          \\localsvr\public\lmhosts    #backup source
#END_ALTERNATE
```

Important: This feature should never be used to include a remote file from a redirected drive, because the LMHOSTS file is shared between local users who have different profiles and different logon scripts. Even on single-user systems, redirected drive mappings can change between logon sessions.

In the above example, the servers **treydc** and **treybdc** are located on remote subnets from the computer that owns the file. The local user has decided to include a list of preferred servers in a local LMHOSTS file located in the C:\PRIVATE directory. During name resolution, the system first includes this private file, then gets the global LMHOSTS file from one of three locations: **treydc**, **treybdc**, or **localsvr**. All names of servers in the #INCLUDE statements must have their addresses preloaded using the #PRE keyword; otherwise, the #INCLUDE statement will be ignored.

If the server named **localsvr** is either defined in the user's private LMHOSTS file or is located in the IP-broadcast area and can be resolved by using the broadcast protocol, then the block inclusion is satisfied if one of the three sources for the global LMHOSTS is available and none of the other servers are used. If no server is available, or for some reason the LMHOSTS file or path is incorrect, an event is added to the event log to indicate that the block inclusion failed.

Tip: Because both the LMHOSTS file and the name cache are searched sequentially, place frequently used servers near the top of the file, then less frequently used servers, followed by remote #INCLUDEs. Finally, the #PRE entries should be left for the end of the file, because these are preloaded into the cache at system startup time and are not accessed later.

Configuration Settings in SYSTEM.INI and PROTOCOL.INI

These topics describe how the protocol settings and system configuration files can be modified for Microsoft TCP/IP-32, which is an NDIS-compliant protocol. This information supersedes information provided in the NETWORK.WRI file for Windows for Workgroups version 3.11 or in the *Microsoft Windows for Workgroups 3.11 Resource Kit*.



Editing initialization files



SYSTEM.INI: system initialization file



PROTOCOL.INI: network initialization file

Editing Initialization Files

The following tips should be observed when editing initialization and configuration files:



Make a backup copy of all files that you might modify, including CONFIG.SYS, AUTOEXEC.BAT, PROTOCOL.INI, and SYSTEM.INI.



Use a text editor such as Notepad or Edit to modify any configuration file. All configuration files must be saved in text-only (ASCII) format.



Exclude the memory address range for your network adapter card (and any other adapters installed in your computer) by using the **x =** option for EMM386.EXE in CONFIG.SYS, or by using the **emmexclude =** entry in the **[386Enh]** section of the SYSTEM.INI file. This will ensure that you don't have a conflict in the upper memory area.

You must restart Windows for Workgroups after editing initialization files for the changes to take effect.

Caution Always create a backup copy of the .INI file before you edit it. Use extreme care when making changes, because errors can lead to unexpected results when you run Windows for Workgroups.

SYSTEM.INI: System Initialization File

The SYSTEM.INI file contains global system information that Windows for Workgroups uses when it starts.

The following sections in SYSTEM.INI can contain entries related to Microsoft TCP/IP-32.



[386Enh], which contains new entries for TCP/IP files



[DNS], which contains settings Domain Name System (DNS)



[MSTCP], which contains settings for Microsoft TCP/IP-32



[NBT], which contains settings for NetBIOS over TCP/IP (the name resolution module for TCP/IP-32)



[network_card], which contains TCP/IP settings for a specific network card

Many of the entries described here are rarely needed and do not appear in the SYSTEM.INI file unless you add them. Most entries have built-in values that are used if the entry does not appear in SYSTEM.INI. You might need to change a value to improve performance of the system or of a specific application.

[386Enh] Entries in SYSTEM.INI

When Microsoft TCP/IP-32 is installed, two entries in **[386Enh]** are changed to reflect the supporting files that are installed.

netmisc = *string*

Entries are added for WSOCK.386 and WSTCP.386.

transport = *string*

Entries are added for VIP.386, VNBT.386, VDHCP.386, VTCP.386, and VTDI.386.

[DNS] Entries in SYSTEM.INI

When Microsoft TCP/IP-32 is installed, this section is added with the following entries that record settings for the Domain Name System (DNS). All of these settings can be specified in the dialog boxes for configuring Microsoft TCP/IP-32.

DNSDomains = *comma-separated strings*

Specifies, in the order to be consulted, the DNS domain names to be appended to a host name.

DNSLookupOrder = *integer*

Specifies the order in which DNS servers and the HOSTS file are consulted for DNS name resolution.

DNSServers = *IP address*

Specifies the IP address for the DNS name server to be consulted for name resolution.

DomainName = *string*

Specifies the DNS domain name for this computer.

HostName = *string*

Specifies the host name that identifies this computer in DNS.

[MSTCP] Entries in SYSTEM.INI

When Microsoft TCP/IP-32 is installed, the **[MSTCP]** section is added to SYSTEM.INI to specify settings for TCP/IP configuration. The following parameters can be added to the **[MSTCP]** section by editing SYSTEM.INI. None of these parameters can be changed using the Microsoft TCP/IP configuration dialog boxes.

BSDUrgent = 0 or 1

If true (1), specifies that Microsoft TCP/IP is to treat urgent data the way some UNIX systems do (with a maximum of 1 byte of urgent data, for example). If false (0), specifies that the stack is to handle urgent data as specified by RFC 1122.

DeadGWDetect = 0 or 1

Specifies whether Microsoft TCP/IP-32 will use another gateway if the current default gateway seems to be down.

Default = 1 (true)

DefaultRcvWindow = 16-bit number

Specifies the default receive window advertised by TCP.

Default = 8192

DefaultTOS = 8-bit number

Specifies the default TOS for IP packets initiated by Microsoft TCP/IP-32.

Default = 0

DefaultTTL = 8-bit number

Specifies the default time to live (TTL) for IP packets from Microsoft TCP/IP-32.

Default = 32

EnableRouting = 0 or 1

Specifies whether to enable static routing. Microsoft TCP/IP-32 does not supply a routing protocol, so all route table entries must be entered using the **route** command.

Default = 0 (false)

IGMPLevel = 0, 1, or 2

Specifies the level of support allowed for IP multicast. The levels correspond to the levels in RFC 1112.

Default = 2

KeepAliveTime= 32-bit number

Specifies the connection idle time in milliseconds before TCP will begin sending keepalives, if keepalives are enabled on a connection.

Default = 2 hours (7200000)

KeepAliveInterval = 32-bit number

Specifies the time in milliseconds between retransmissions of keepalives, once the **KeepAliveTime** has expired. Once **KeepAliveTime** has expired, keepalives are sent every **KeepAliveInterval** milliseconds until a response is received, up to a maximum of **MaxDataRetries** before the connection is aborted.

Default = 1 second (1000)

MaxConnections = 32-bit number

Specifies the maximum number of concurrent connections.

Default = 100

MaxConnectRetries = 32-bit number

Specifies the number of times a connection attempt (SYN) will be retransmitted before giving up. The initial retransmission timeout is 3 seconds, and it is doubled each time up to a maximum of 2 minutes.

Default = 3

MaxDataRetries = 32-bit number

Specifies the maximum number of times a segment carrying data or an FIN will be retransmitted before the connection is aborted. The retransmission timeout itself is adaptive and will vary according to link conditions.

Default = 5

PMTUBlackHoleDetect = 0 or 1

Specifies whether the stack will attempt to detect Maximum Transmission Unit (MTU) routers that do not send back ICMP fragmentation-needed messages. Setting this parameter when it is not needed can cause performance degradation.

Default = 0 (false)

PMTUDiscovery = 0 or 1

Specifies whether the stack will attempt to do path MTU discovery as specified in RFC 1191.

Default = 1 (true)

RoutingBufSize = 32-bit number

Specifies the total amount of buffer space to allocate for routing packets. This parameter is ignored if **EnableRouting=0**.

Default = 73216

RoutingPackets = 32-bit number

Specifies the maximum of packets that may be routed simultaneously. This parameter is ignored if **EnableRouting=0**.

Default = 50

The entries in **[MSTCP]** also include the following, which can be configured using the TCP/IP Configuration dialog boxes:

Interfaces = *network_card*

Specifies the network card(s) configured and bound to the TCP/IP protocol.

[NBT] Entries in SYSTEM.INI

NetBIOS over TCP/IP (VNBT.386) is the module in Microsoft TCP/IP32 that provides NetBIOS session management, including NetBIOS name resolution, either by broadcast or using WINS servers. The following values can be configured for NetBIOS over TCP/IP, although not all appear in SYSTEM.INI by default. The value used depends on the following:



If the parameter is defined in SYSTEM.INI, that value is used.



If the parameter is not specified in SYSTEM.INI but is a DHCP parameter specified by the DHCP server, the value for the DHCP option is used.



If the parameter is not defined in SYSTEM.INI and is not specified as a DHCP parameter, the default value is used.

BroadcastAddress = *broadcast address in hexadecimal*

Specifies the address to use for NetBIOS name query broadcasts.

Default = Calculated based on the IP address and the subnet mask

BcastNameQueryCount = *integer*

Specifies the number of times the system will retry NetBIOS name query broadcasts.

Default = 3

BcastQueryTimeout = *milliseconds*

Specifies the period of time the system will wait before timing broadcast name queries. The minimum value is 100.

Default = 750

CacheTimeout = *milliseconds*

Specifies how long NetBIOS names are cached. The minimum is 60000 milliseconds (1 minute).

Default = 360000 milliseconds (6 minutes)

DnsServerPort = *port*

Specifies which DNS port on the DNS server to send queries when resolving a name using DNS.

Default = 53

EnableDNS = *0 or 1*

Specifies whether Windows networking applications that use WINS for name resolution can also use DNS for name resolution, where 1 = true.

Default = 0 (false)

EnableProxy = *0 or 1*

Specifies whether this computer is a WINS proxy agent, where 1 = true.

Default = 0

InitialRefreshT.O. = *milliseconds*

Specifies the interval over which to contact WINS to refresh the name. The minimum is 16 minutes, and the maximum is approximately 50 days (0xFFFFFFFF).

Default = 16 minutes (16*60*1000 in milliseconds)

LANABASE = *NetBIOS LAN adapter number*

Determines the NetBIOS LAN adapter number that NetBIOS over TCP/IP will use. The number assigned is based on other installed transports. You should only need to change the LAN adapter

number if you have a NetBIOS application that specifies it must use a certain LAN adapter number.
Default = Determined by Setup, depending on the other network protocols that have been installed and they order in which they were installed

LmHostFile = *pathname*

Specifies the fully qualified path to the LMHOSTS file. If no value is specified, LMHOSTS functionality is disabled. During installation of Microsoft TCP/IP-32, this entry is added to SYSTEM.INI if the Enable LMHOSTS File option is checked in the Advanced Microsoft TCP/IP Configuration dialog box.
Default = %windows%\lmhosts

LmhostsTimeout = *milliseconds*

Specifies the period of time the system will wait before timing when seeking LMHOSTS for name resolution. The minimum value is 1000 (1 second).
Default = 10000 (10 seconds)

NameServer1 = *###.###.*

NameServer2 = *###.###.*

Specifies the IP address for the WINS server to use for NetBIOS name resolution. These values can be configured via DHCP.
Default = no entry

NameServerPort = *port*

Specifies the UDP port on the name server to which to send name queries or registrations.
Default = 137

NameSrvQueryCount = *integer*

Specifies the number of times the system will try to contact the WINS server for NetBIOS name resolution.
Default = 3

NameSrvQueryTimeout = *milliseconds*

Specifies how long the system will wait before timing out a name server query. The minimum is 100.
Default = 750

NameTableSize = *integer*

Specifies the maximum number of names in the NetBIOS name table. The minimum allowable value is 1 and the maximum is 255.
Default = 17

NodeType = *1, 2, 4, or 8*

Specifies the mode of NetBIOS name resolution used by NetBIOS over TCP/IP, where 1 = b-node, 2 = p-node, 4 = m-node, and 8 = h-node. This value can be configured via DHCP.

Default = 1 (b-node), if no value is specified; if WINS servers are specified and **NodeType** is not, then the default is 8 (h-node)

RandomAdapter = *0 or 1*

For a multihomed computer, specifies whether to respond with an IP address randomly from the set of addresses on the computer or whether to return the IP address of the adapter that the request came in upon.

Default = 0 (not random; that is, return the address of the adapter that the request came in upon)

Scopeld = (blank by default)

Specifies the NetBIOS scope ID to be added to the NetBIOS computer name for Windows networking. This value can be configured via DHCP.

SessionKeepAlive = *milliseconds*

Specifies how often to send session keepalive packets on active sessions. The minimum is 60*1000 milliseconds (60 seconds).

Default = 3600 seconds (1 hour)

SessionTableSize = *integer*

Specifies the maximum number of sessions in the NetBIOS session table. The minimum allowable value is 1 and the maximum is 255.

Default = 255

SingleResponse = *0 or 1*

For a multihomed computer, specifies whether to send all IP addresses on a name query request from WINS. If this value is 1 (true), the system will send one address; otherwise, it will return all the addresses of its adapters.

Default = 0

Size/Small/Medium/Large = *1, 2, or 3*

Specifies how many buffers of various types to preallocate and the maximum that can be allocated, where 1 = small, 2 = medium, and 3 = large.

Default = 1; the default is 3 if the WINS proxy is enabled

[*network_card*] Entries in SYSTEM.INI

When Microsoft TCP/IP-32 is installed, the [***network_card***] section is added to SYSTEM.INI to specify TCP/IP settings for the network card, where the name of the card is the name of the section, such as [***ms\$niupctp0***]. The entries in this section can include the following:

Binding = *network_card*

Specifies the name of the network card to which the TCP/IP protocol is bound.

Description = *card_type*

Specifies the type of network card. For example, DEC (DE201) EtherWorks Turbo/IP.

IPAddress = *###.###*

Specifies the IP address for the network card in the form *w.x.y.z*, such as 11.103.41.12.

IPMask = *###.###*

Specifies the subnet mask for the network card, such as 255.255.255.0.

MaxMTU = *16-bit integer*

Specifies the maximum size datagram IP can pass to a media driver. SNAP and/or source routing headers (if used on the media) are not included in this value. For example, on an Ethernet MaxMTU will default to 1500. The actual value used will be the minimum of the value specified with this parameter and the size reported by the media driver.

Default = The size reported by the media driver.

PROTOCOL.INI: Network Initialization File

[MSTCP32] Entries in PROTOCOL.INI

When Microsoft TCP/IP-32 is installed, an **[MSTCP32]** section is added to PROTOCOL.INI to record the name of the network card to which TCP/IP is bound, and the LAN adapter number is identified. For example:

```
[MSTCP32]
BINDINGS = MS$NIUPCTP
LANABASE = 2
```

[network.setup] Entries in PROTOCOL.INI

When Microsoft TCP/IP-32 is installed, an additional entry is added to the **[network.setup]** section of PROTOCOL.INI for the LAN adapter number. The format for this entry is the following:

lanax = card_ID,0 | 1,tcpip-32

For example:

```
lana3 = ms$niupctp,1,tcpip-32
```

Internet and Vendor Sources for Windows Sockets Applications

A list of Internet sources for Windows Sockets applications is available via info@lcs.com.

The following vendors provide Windows Sockets applications.

AGE Logic, Inc.
9985 Pacific Heights Blvd.
San Diego, CA 92121
Phone: (619) 455-8600
Fax: (619) 597-6030
X Window software

American Computer & Electronics Corp.
209 Perry Parkway
Gaithersburg, MD 20877
Phone: (301) 258-9850
Fax: (301) 921-0434
Network management

Attachmate Corporation
3617 131st Avenue SE
Bellevue, WA 98006-9930
Phone: (800) 426-6283
Fax: (206) 747-9924
Terminal emulation

Beame and Whiteside
P.O. Box 8130
Dundas, Ontario L9H 5E7
CANADA
Phone: (416) 765-0822
Fax: (416) 765-0815
Terminal emulation, file transfer, remote process execution, e-mail, NFS, network printing

Digital Equipment Corporation
Attn: Lori Heron
2 Results Way
MRO2-2/D10
Marlboro, MA 01752-3011
Phone: (508) 467-7855
Fax: (508) 467-1926
eXcursion, X Window server and client libraries

Distinct Corporation
14395 Saratoga Ave. Suite 120
Saratoga, CA 95070
Phone: (408) 741-0781
Fax: (408) 741-0795
Terminal emulation, file transfer, X Window, remote process execution, email, NFS, ONC/RPC

Esker, Inc.
1181 Chess Drive, Suite C
Foster City, CA 94404
Phone: (415) 341-9065

Fax: (415) 341-6412

Terminal emulation, file transfer, X Window, remote process execution, NFS

Executive Systems/XTree Company

4115 Broad Street Bldg. #1

San Luis Obispo, CA 93401-7993

Phone: (805) 541-0604

Fax: (805) 541-4762

Network management

Frontier Technologies Corporation

10201 North Port Washington Road

Mequon, Wisconsin 53092

Phone: (414) 241-4555

Fax: (414) 241-7084

Terminal emulation, file transfer, remote process execution, e-mail, NFS, NNTP, TelnetD, network printing

Gallagher & Robertson A/S

Postboks 1824, Vika

0123 OSLO

NORWAY

Phone: (+47) 2 41 85 51

Fax: (+47) 2 42 89 22

Terminal emulation, file transfer

Genisys Comm, Inc.

314 S. Jay Street

Rome, NY 13440

Phone: (315) 339-5502

Fax: (315) 339-5528

Terminal emulation, file transfer

Gradient Technologies, Inc.

577 Main Street, Suite 4

Hudson, MA 01749

Phone: (508) 562-2882

Fax: (508) 562-3549

DCE (OSF distributed computing environment)

Hummingbird Communications Ltd.

2900 John Street, Unit 4

Markham, Ontario L3R 5G3

CANADA

Phone: (416) 470-1203

Fax: (416) 470-1207

File transfer, remote process execution, terminal emulation, X Window

Hypercube, Inc.

Unit 7-419 Phillip Street

Waterloo, Ontario N2L 3X2

CANADA

Phone: (519) 725-4040

Fax: (519) 725-5193

Modeling software, remote process execution

I-Kinetics, Inc.
19 Bishop Allen Drive
Cambridge, MA 02139
Phone: (617) 661-8181
Fax: (617) 661-8625
Middleware, remote process execution

John Fluke Mfg. Co.
P.O. Box 9090
Everett, WA 98206
Phone: (206) 356-5847
Fax: (206) 356-5790
Instrument control software

JSB Computer Systems Ltd.
Cheshire House, Castle Street
Macclesfield, Cheshire
ENGLAND SK11 6AF
Phone: (+44) 625-433618
Fax: (+44) 625-433948

JSB Corporation [USA]
Suite 115, 108 Whispering Pines Drive
Scotts Valley, CA 95066
Phone: (408) 438-8300
Fax: (408) 438-8360
Terminal emulation, file transfer, X Window, remote process execution, virtual sockets library

Lanera Corporation
516 Valley Way
Milpitas, CA 95035
Phone: (408) 956-8344
Fax: (408) 956-8343
Terminal emulation, file transfer, X Window, remote process execution, NFS, SNMP

Microdyne Corp.
239 Littleton Road
Westford, MA 01886
Phone: (508) 392-9953
Fax: (508) 392-9962
File transfer

NetManage, Inc.
20823 Stevens Creek Blvd.
Cupertino, CA 95014
Phone: (408) 973-7171
Fax: (408) 257-6405
Terminal emulation, file transfer, X Window, e-mail, NFS, TN3270, BIND, SNMP

Network Computing Devices
9590 SW Gemini
Beaverton, OR 97005
Phone: (503) 641-2200
Fax: (503) 643-8642
X Window

Spry, Inc.
1319 Dexter Ave. N
Seattle, WA 98109
Phone: (206) 286-1412
Fax: (206) 286-1722
Terminal emulation, file transfer, email, network printing

SunSelect
2 Elizabeth Drive
Chelmsford, MA 01824-4195
Phone: (508) 442-2300
Fax: (508) 250-2300
E-mail

TurboSoft Pty Ltd.
248 Johnston Street
Annandale, NSW 2038
AUSTRALIA
Phone: (+612) 552-1266
Fax: (+612) 552-3256
Terminal emulation, file transfer, network printing

Unipalm Ltd.
216, Science Park, Milton Road
Cambridge, Cambridgeshire
CB4 4WA ENGLAND
Phone: (+44) 223-420002
Fax: (+44) 223-426868
E-mail

VisionWare UK
57 Cardigan Lane
Leeds, ENGLAND LS4 2LE
Phone: (+44) 532-788858
Fax: (+44) 532-304676

VisionWare USA
1020 Marsh Road
Suite 220
Menlo Park, CA 94025
Phone: (415) 325-2113
Fax: (415) 325-8710
Terminal emulation, file transfer, X Window, remote process execution

VisiSoft
430 10th Street NW, Suite S008
Atlanta, GA 30318
Phone: (404) 874-0428
Fax: (404) 874-6412
Network management

Walker Richer & Quinn, Inc.
1500 Dexter Ave. N.
Seattle, WA 98109
Phone: (206) 217-7500
Fax: (206) 217-0293

Terminal emulation, file transfer, X Window

Using MSTCP32.DEF for Administrative Installation

An administrator can modify the file MSTCP32.DEF that is provided with Microsoft TCP/IP to provide custom configuration defaults to be added to SYSTEM.INI during installation.

If you use MSTCP32.DEF when installing Microsoft TCP/IP, each workstation will automatically have appropriate values set for TCP/IP. If your site uses DHCP servers, individual users will not be asked to provide any values during the installation--everything will be configured automatically.

The template for MSTCP32.DEF contains the following entries. Any valid parameter for TCP/IP can also be added to customize this file.

```
[mstcp]
deadgwdetect=1
pmtudiscover=1
```

```
[netbt]
nameserver1=
nameserver2=
scopeid=
```

```
[dns]
domain=
dnsservers=
dnsdomains=
```

```
[misc]
dhcpenabled=0
ipmask=
defaultgateway=
```

Note: If **RouterEnabled=0** in MSTCP32.DEF, then the Enable IP Routing check box is always dimmed in the Advanced Microsoft TCP/IP dialog box. If **RouterEnabled=1**, the check box is available if there are multiple IP addresses or multiple network cards, and the user can enable the router by checking this option.

For more information about MSTCP32.DEF, see the template included with the Microsoft TCP/IP files.

