# Microsoft® TCP/IP for Windows™ for Workgroups

**Version 1.0a**

**For the Microsoft Windows for Workgroups Operating System with Integrated Networking**

**Microsoft Corporation**

Information in this document is subject to change without notice and does not represent a commitment on the part of Microsoft Corporation. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or nondisclosure agreement.   No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or storage retrieval systems, for any purpose other than the purchaser's personal use, without the express written permission of Microsoft Corporation.

## Microsoft TCP/IP for Windows for Workgroups Software

## Introduction

## Overview of Microsoft TCP/IP for Windows for Workgroups

## Installing and Configuring Microsoft TCP/IP

## Understanding Name Resolution

## Networking Concepts for TCP/IP

## PROTOCOL.INI Parameters

## TCPUTILS.INI Parameters

## Error Messages

# License Agreement

**IMPORTANT--READ CAREFULLY BEFORE OPENING SOFTWARE PACKET(S): THE FOLLOWING LICENSE AGREEMENT SUPERSEDES ANY OTHER AGREEMENT PROVIDED WITH THE SOFTWARE. By opening the sealed packet(s) containing the software, you indicate your acceptance of the following Microsoft License Agreement ("Agreement"). If a separate multilingual license booklet is included in your product package, then the Warranty provisions of the applicable license in that booklet applies to you. No other part of the multilingual license booklet applies to you.**

MICROSOFT LICENSE AGREEMENT

Microsoft TCP/IP for Windows for Workgroups

This is a legal agreement between you (either an individual or an entity), and Microsoft Corporation. **By opening the enclosed sealed disk package(s) and/or by using the SOFTWARE you are agreeing to be bound by the terms of this Agreement. If you do not agree to the terms of this Agreement, promptly return the unopened disk package(s) and the accompanying items (including printed materials and binders or other containers) to the place you obtained them for a full refund.**

MICROSOFT SOFTWARE LICENSE

1. GRANT OF LICENSE. The Microsoft License Agreement ("Agreement") permits you to make and use an unlimited number of copies of the specified version of the Microsoft software product identified above (the "SOFTWARE") for your internal use provided that (a) the SOFTWARE is not modified in any way and (b) you maintain the copyright notice on all copies of the SOFTWARE.

2. COPYRIGHT. The SOFTWARE (including any images, "applets", photographs, animations, video, audio, music and text incorporated into the SOFTWARE) is owned by Microsoft or its suppliers and is protected by United States copyright laws and international treaty provisions. You may not copy the printed materials accompanying the SOFTWARE.

3. OTHER RESTRICTIONS. You may not rent or lease the SOFTWARE, but you may transfer the SOFTWARE and accompanying printed materials on a permanent basis provided you retain no copies and the recipient agrees to the terms of this Agreement. You may not reverse engineer, decompile, or disassemble the SOFTWARE except to the extent the foregoing restriction is expressly prohibited by applicable law. If the SOFTWARE is an update or has been updated, any transfer must include the most recent update and all prior versions.

LIMITED WARRANTY

LIMITED WARRANTY. Microsoft warrants that the original copy of the SOFTWARE will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of receipt. Any implied warranties on the SOFTWARE are limited to ninety (90) days. Some states/jurisdictions do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you.

CUSTOMER REMEDIES. Microsoft's entire liability and your exclusive remedy shall be, at Microsoft's option, either (a) return of the price paid or (b) repair or replacement of the SOFTWARE that does not meet Microsoft's Limited Warranty and that is returned to Microsoft with a copy of your receipt. This Limited Warranty is void if failure of the SOFTWARE has resulted from accident, abuse, or misapplication. Any replacement SOFTWARE will be warranted for the remainder of the original warranty period or thirty

(30) days, whichever is longer. Neither these remedies nor any product support services offered by Microsoft are available for this U.S.A. version product outside of the United States of America without proof of purchase from an authorized non-U.S. source.

NO OTHER WARRANTIES. To the maximum extent permitted by applicable law, Microsoft disclaims all other warranties, either express or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to the SOFTWARE and the accompanying written materials. This limited warranty gives you specific legal rights. You may have others, which vary from state/jurisdiction to state/jurisdiction.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES. To the maximum extent permitted by applicable law, in no event shall Microsoft or its suppliers be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use or inability to use this Microsoft product, even if Microsoft has been advised of the possibility of such damages. Because some states/jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

## U.S. GOVERNMENT RESTRICTED RIGHTS

The SOFTWARE and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software--Restricted Rights at 48 CFR 52.227-19, as applicable. Manufacturer is Microsoft Corporation/One Microsoft Way/Redmond, WA 98052-6399.

If you acquired this product in the United States, this Agreement is governed by the laws of the State of Washington.

If you acquired this product in Canada, this Agreement is governed by the laws of the Province of Ontario, Canada. Each of the parties hereto irrevocably attorns to the jurisdiction of the courts of the Province of Ontario and further agrees to commence any litigation which may arise hereunder in the courts located in the Judicial District of York, Province of Ontario.

If this product was acquired outside the United States, then local law may apply.

Should you have any questions concerning this Agreement, or if you desire to contact Microsoft for any reason, please contact the Microsoft subsidiary serving your country, or write: Microsoft Customer Sales and Service/One Microsoft Way/Redmond, WA 98052-6399.

Si vous avez acquis votre produit Microsoft au CANADA, la garantie limitée suivante vous concerne :

GARANTIE LIMITEE – Microsoft garantit que la performance de la copie originale du LOGICIEL sera substantiellement en conformité avec le(s) manuel(s) de produits qui accompagne(nt) le LOGICIEL pour une période de quatre-vingt-dix (90) jours à compter de la date de réception. Toute garantie implicite concernant le LOGICIEL et le matériel est limitée à quatre-vingt-dix (90) jours et un (1) an, respectivement.

RECOURS DU CLIENT – La seule obligation de Microsoft et votre recours exclusif seront, au choix de Microsoft, soit (a) le remboursement du prix payé ou (b) la réparation ou le remplacement du LOGICIEL ou du matériel qui n'est pas conforme à la Garantie Limitée de Microsoft et qui est retourné à Microsoft avec une copie de votre reçu. Cette Garantie Limitée est nulle si le défaut du LOGICIEL ou du matériel est causé par un accident, un traitement abusif ou une mauvaise application. Tout LOGICIEL de remplacement sera garanti pour le reste de la période de garantie initiale ou pour trente (30) jours, selon laquelle de ces deux périodes est la plus longue.

AUCUNE AUTRE GARANTIE – MICROSOFT DESAVOUE TOUTE AUTRE GARANTIE, EXPRESSE OU IMPLICITE, Y COMPRIS MAIS NE SE LIMITANT PAS AUX GARANTIES IMPLICITES DU CARACTERE ADEQUAT POUR LA COMMERCIALISATION OU UN USAGE PARTICULIER EN CE QUI CONCERNE LE LOGICIEL, LE(S) MANUEL(S) DE PRODUITS, LA DOCUMENTATION ECRITE ET TOUT MATERIEL QUI L'ACCOMPAGNENT. CETTE GARANTIE LIMITEE VOUS ACCORDE DES DROITS JURIDIQUES SPECIFIQUES.

PAS D'OBLIGATION POUR LES DOMMAGES INDIRECTS – MICROSOFT OU SES FOURNISSEURS N'AURONT D'OBLIGATION EN AUCUNE CIRCONSTANCE POUR TOUT AUTRE DOMMAGE QUEL QU'IL SOIT (Y COMPRIS, SANS LIMITATION, LES DOMMAGES ENTRAINES PAR LA PERTE DE BENEFICES, L'INTERRUPTION DES AFFAIRES, LA PERTE D'INFORMATION COMMERCIALE OU TOUTE AUTRE PERTE PECUNIAIRE) DECOULANT DE L'UTILISATION OU DE L'IMPOSSIBILITE D'UTILISATION DE CE PRODUIT MICROSOFT, ET CE, MEME SI MICROSOFT A ETE AVISE DE LA POSSIBILITE DE TELS DOMMAGES. EN TOUT CAS, LA SEULE OBLIGATION DE MICROSOFT EN VERTU DE TOUTE DISPOSITION DE CETTE CONVENTION SE LIMITERA AU MONTANT EN FAIT PAYE PAR VOUS POUR LE LOGICIEL.

La présente Convention est régie par les lois de la province d'Ontario, Canada. Chacune des parties à la présente reconnaît irrévocablement la compétence des tribunaux de la province d'Ontario et consent à instituer tout litige qui pourrait découler de la présente auprès des tribunaux situés dans le district judiciaire de York, province d'Ontario.

Au cas où vous auriez des questions concernant cette licence ou que vous désiriez vous mettre en rapport avec Microsoft pour quelque raison que ce soit, veuillez contacter la succursale Microsoft desservant votre pays, dont l'adresse est fournie dans ce produit, ou écrire à : Microsoft Customer Sales and Service, One Microsoft Way, Redmond, Washington 980526399.

# Installing the Software

The instructions for installing Microsoft TCP/IP for Windows for Workgroups contained in this help file supersede the instructions provided in the Windows for Workgroups 3.11 Resource Kit Addendum.

See the READTCP!.TXT file provided on the Microsoft TCP/IP for Windows for Workgroups disk (also copied in the WINDOWS directory when TCP/IP is installed) for additional information on installing Microsoft TCP/IP for Windows for Workgroups.

If you have installed a previous version of Microsoft TCP/IP for Windows for Workgroups (such as the TCP/IP support offered with LAN Manager Version 2.2), be sure to remove it using the Network section of Windows Setup before running the installation procedure described in the section entitled "Installing and Configuring Microsoft TCP/IP."

## Related Topics:

Installing Microsoft TCP/IP

# For Support

Support for Microsoft TCP/IP for Windows for Workgroups is not available from the standard Windows for Workgroups Product Support Services phone line. If you have questions, please contact your Microsoft Solution Channel member. Support for this product is also available through Microsoft's fee-based support plans. For information locating a Solutions Channel member near you or about Microsoft's support options, call Microsoft Inside Sales in the United States at (800) 227-4679 or your local Microsoft subsidiary.

# Before You Begin

The *Microsoft® TCP/IP for Windows™ for Workgroups Installation and Configuration Guide* describes how to install, configure, and troubleshoot Microsoft TCP/IP on a workstation running the Microsoft Windows for Workgroups operating system with integrated networking.

This manual assumes that you are familiar with the Microsoft Windows for Workgroups operating system. If you are not familiar with this product, refer to your Microsoft Windows for Workgroups documentation set.

# How to Use This Manual

Turn to the part that contains the information you need.

**"Overview of Microsoft TCP/IP for Windows for Workgroups"**
Provides an overview of the Microsoft TCP/IP network protocol and IP addresses.

**"Installing and Configuring Microsoft TCP/IP"**
Describes the process for installing and configuring Microsoft TCP/IP on a workstation running the Microsoft Windows for Workgroups operating system.

**"Understanding Name Resolution"**
Describes how the HOSTS and LMHOSTS host table files enable your workstation to access resources on different TCP/IP networks.

**"Networking Concepts for TCP/IP"**
Introduces key TCP/IP networking concepts for network administrators interested in a technical discussion of TCP/IP.

**"PROTOCOL.INI Parameters"**
Contains information about parameters that can be changed in the PROTOCOL.INI file.

**"TCPUTILS.INI Parameters"**
Describes the adjustable parameters in the TCPUTILS.INI file.

**"Error Messages"**
Lists the possible error messages for TCP/IP and for the **ping** and **addname** utilities, and provides recovery procedures.

# Documentation Conventions

This manual uses several type styles and special characters.

**Documentation Type Styles and Special Characters**

| Convention | Use |
|---|---|
| **bold** | Represents commands, command options, and file entries. Type bold words exactly as they appear (for example, **net use**). |
| *italic* | Introduces new terms and represents variables. For example, the variable *computername* indicates that you type the name of a workstation or a server. |
| `monospace` | Represents examples, screen displays, and error messages. |
| ALL UPPERCASE | Represent filenames and paths. You can, however, type entries in uppercase or lowercase letters, or a combination of the two. |
| SMALL CAPITALS | Represent key names (for example, CTRL, ENTER, and F2). |
| [brackets] | Enclose optional items in syntax statements. For example, [*password*] indicates that you can choose to type a password with the command. Type only the information within the brackets, not the brackets themselves. |

# Finding Further Information

If you are integrating with a LAN Manager network, you might also want to refer to the following manuals in the Microsoft LAN Manager documentation set:

*Microsoft LAN Manager Installation and Configuration Guide*
Provides information about installing LAN Manager software, about the network device drivers used with LAN Manager, and about configuring workstations and servers.

*Microsoft LAN Manager Administrator's Guide*
Provides step - by - step instructions for administering a LAN Manager network.

*Microsoft LAN Manager Administrator's Reference*
Provides reference information about LAN Manager commands and utilities for MS® OS/2® computers, and about the LAN Manager program directory and initialization file.

For a more technical discussion of the topics mentioned in this manual, refer to the following texts and articles:

Calbaum, M., Porcaro, F., Ruegsegger, M., and Backman, B. 1993. "Untangling the Windows Sockets API." *Dr. Dobb's Journal,* February:66-71.

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

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

Comer, D. and Stevens, D. 1993. *Internetworking with TCP/IP Volume III: Client* - *Server Programming and Applications*, Englewood Cliffs, N.J.: Prentice Hall.

Hall, M., Tofiq, M., Arnold, G., Treadwell D., and Sanders, H. 1993. "Windows Sockets: An Open Interface for Network Programming Under Microsoft Windows," Version 1.1, Revision A.

Leffler, S. et. al. *An Advanced 4.3BSD Interprocess Communication Tutorial*.

Stevens, W.R. 1990. *UNIX Network Programming*, Englewood Cliffs, N.J.: Prentice Hall.

Volkman, Victor R. 1992. "Plug into TCP/IP with Windows Sockets." *Windows/DOS Developer's Journal,* December:6-18.

# Overview

This section introduces the Microsoft TCP/IP for Windows for Workgroups operating system with integrated networking. It provides background material on basic TCP/IP concepts and gives you the information you need before installing Microsoft TCP/IP for Windows for Workgroups. For more detailed information on TCP/IP and its integration with Microsoft Windows for Workgroups, see "Networking Concepts for TCP/IP."

*Transmission Control Protocol/Internet Protocol* (TCP/IP) is a networking protocol that provides communication across interconnected networks, between computers with diverse hardware architectures and various operating systems. TCP/IP can be used with Windows for Workgroups or to connect to Microsoft LAN Manager or non - Microsoft (for example, UNIX®) hosts.

# What Is TCP/IP 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 systems on a network. Microsoft TCP/IP for Windows for Workgroups enables enterprise networking and connectivity on your Windows for Workgroups - based desktop system. Adding TCP/IP to a Windows for Workgroups configuration offers the following advantages:

- ***A standard, routable, enterprise networking protocol for Windows for Workgroup services***

  TCP/IP is viewed as the most complete and accepted networking protocol available. Virtually all modern operating systems offer TCP/IP support, and most large networks rely on TCP/IP for all their network traffic.

- ***An architecture that facilitates connection to foreign systems***

  Because most operating systems offer TCP/IP, many standard utilities have been designed to access and transfer data between heterogeneous environments. Examples of such standards include File Transfer Protocol (FTP) and Telnet (Terminal Emulation Protocol). The Windows Sockets interface offers compatibility with many foreign host connectivity products. Several applications vendors support this Application Programming Interface (API) standard.

- ***A robust, cross - platform client - server framework***

  TCP/IP for Windows for Workgroups offers the Windows Sockets interface, which is ideal for developing client - server applications. A Windows Sockets application developed to be used with Microsoft TCP/IP will be able to run other vendors' Windows Sockets - compliant stacks as well.

# How Does TCP/IP Work?

The name "TCP/IP" is somewhat confusing since TCP (transmission control protocol) and IP (internet protocol) are really only two protocols in the family of Internet protocols. Over time, however, "TCP/IP" has been used in industry to denote the family of common Internet protocols. How do these protocols work and what do they do? The following sections briefly explain how the TCP and IP protocols work. For a more detailed explanation, see "Networking Concepts for TCP/IP." For a full explanation, see *Internetworking with TCP/IP, Volume I* by Douglas E. Comer (Prentice Hall, 1991).

## Related Topics:

How TCP Works
How IP Works
Example

## How TCP Works

TCP is a reliable, *connection - oriented* protocol. Connection - oriented implies that TCP first establishes a connection between the two systems that intend to exchange data. Since most networks are built on *shared media* (for example, several systems sharing the same cabling), it is necessary to break chunks of data into manageable pieces so that no two communicating systems monopolize the network. These pieces are called *packets.* When an application sends a message to TCP for transmission, TCP breaks the message into packets, sized appropriately for the network, and sends them over the network.

### Related Topics:

Sequence Numbers, Checksum, and Port ID
TCP Headers

### *Sequence Numbers, Checksum, and Port ID*

Because a single message is often broken into many packets, TCP marks these packets with *sequence numbers* before sending them. The sequence numbers allow the receiving system to properly reassemble the packets into the original message. Being able to reassemble the original message is not enough – the accuracy of the data must also be verified. TCP does this by computing a *checksum*. A checksum is a simple mathematical computation applied, by the sender, to the data contained in the TCP packet. The recipient then does the same calculation on the received data and compares the result with the checksum that the sender computed. If the results match, the recipient sends an acknowledgment (ACK). If the results do *not* match, the recipient asks the sender to resend the packet. Finally, TCP uses *port IDs* to specify which application running on the system is sending or receiving the data.

### *TCP Headers*

The port ID, checksum, and sequence number are inserted into the TCP packet in a special section called the *header*. The header is at the beginning of the packet containing this and other "control" information for TCP.

## How IP Works

IP is the *messenger protocol* of TCP/IP. The IP protocol, much simpler than TCP, basically addresses and sends packets. IP relies on three pieces of information, which you provide, to receive and deliver packets successfully: *IP address*, *subnet mask*, and *default gateway*.

### Related Topics:

IP Addresses
Subnet Mask
Network and Host IDs
Default Gateway

## *IP Addresses*

The *IP address* identifies your system on the TCP/IP network. IP addresses are 32 - bit addresses that are globally unique on a network. They are generally represented in *dotted decimal notation.* Dotted decimal notation (explained in "Networking Concepts for TCP/IP") separates the four bytes of the address with periods. An IP address looks like this:

```
102.54.94.97
```

Although an IP address is a single value, it really contains two pieces of information:

- Your system's network ID
- Your system's host (or system) ID

## *Subnet Mask*

The *subnet mask*, also represented in dotted decimal notation, is used to extract these two values from your IP address. The value of the subnet mask is determined by setting the network ID bits of the IP address to 1's and the host ID bits to 0's. The result allows TCP/IP to determine the host and network IDs of the local workstation. For example:

**Table 1. Understanding an IP Address**

| | | |
|---|---|---|
| When the *IP address* is | 102.54.94.97 | (specified by the user) |
| And the *subnet mask* is | 255.255.0.0 | (specified by the user) |
| The *network ID* is | 102.54 | (IP address and subnet mask) |
| And the *host ID* is | 94.97 | (IP address and subnet mask) |

## *Network and Host IDs*

The *network ID* identifies a group of systems that are all located on the same physical network. In *internetworks* (networks formed by a collection of networks), there are as many unique network IDs as there are networks. TCP/IP networks are connected by *routers* (or *gateways*), which have knowledge of the networks that are connected in the internet. The *host ID* identifies your system within a particular network ID.

### *Default Gateway*

The *default gateway* is needed only for systems that are part of an internet. When IP gets ready to send a packet on the wire, it inserts the local (source) IP address and destination address of the packet in the IP header, and verifies that the network ID of the destination matches the source. If they match, the packet is sent directly to the destination system on the local network. If the network IDs do *not* match, the packet is forwarded to the default gateway for delivery. Since the default gateway has knowledge of the network IDs of the other networks in the internet, it forwards 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*.

## Example

Here is an example to show how TCP/IP might deliver a message. To keep things simple, the example uses an analogy with the U.S. postal system, to describe how these protocols work.

Steve (source host ID) in Seattle (source network ID) wants to send a two page letter (message) to Dana (destination host ID) in Dartmouth (destination network ID). There is a limit to the length of the message that can be sent in a single letter (maximum transmission unit, or MTU).

First, Steve establishes the connection by writing a short note to Dana: *"Dana, I am going to send you a two - page letter now via mail. Is this okay?   -Steve"* Steve then puts the letter in an envelope (IP packet) and addresses it to Dana in Dartmouth (destination IP address).

Steve's mail carrier (Steve's default gateway) picks up the letter, doesn't know where Dartmouth is, and forwards it to the Seattle post office (gateway). From Seattle, the messages goes (routes) to the Dartmouth post office (Dana's default gateway). The Dartmouth mail carrier delivers it to Dana.

Dana writes back (ACKs) to Steve saying: *"Okay, I'm ready and waiting for your letter."* She addresses the envelope using the return address from the envelope (IP packet) to Steve in Seattle. Dana's mail carrier then carries her reply to the Dartmouth post office, and it travels back the same way Steve's letter came.

Steve realizes that his letter (message) is too long to fit in a single envelope (IP packet), tears off the first page (TCP packet) and writes *"1 of 2"* in the margin (TCP header), puts it into the envelope addressed to Dana (IP packet), and mails it.

The first page is successfully routed to Dana, who opens the envelope and reads the contents. The story is so compelling that she gets impatient after waiting for the second message and mails a quick note (ACK) to Steve: *"Steve, I received the first page of the message, but I haven't seen any since. Are you okay?"*

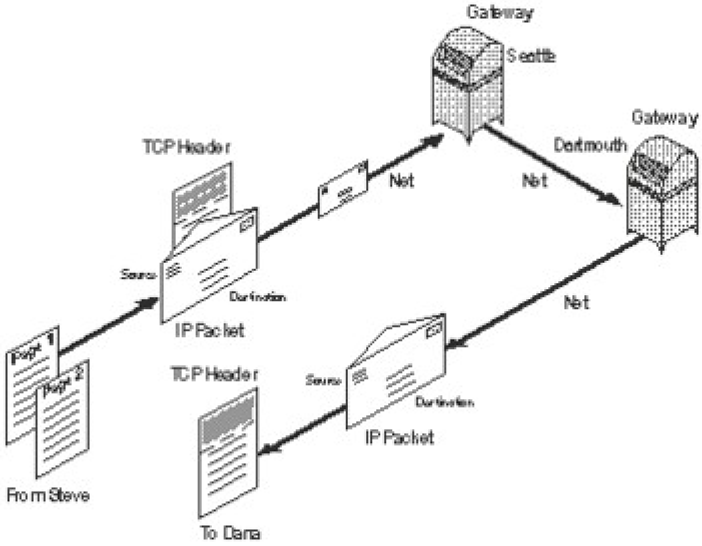


**Figure 1.      How TCP, IP addressing, and routing work**

Steve reads Dana's letter and immediately takes the second page (TCP packet), writes *"2 of 2"* in the margin (TCP header), puts it in an envelope addressed to Dana (IP packet), and mails it.

The second note is also successfully routed to Dana, so she replies (ACKs) with one final note: *"Good story, Steve. Thanks for the laughs."*

# Overview of Windows Sockets

Microsoft TCP/IP for Windows for Workgroups includes support for Windows Sockets. A socket provides an end point to a connection; two sockets form a complete path. A socket works as a bi - directional pipe for incoming and outgoing data. The Windows Sockets API is a networking API tailored for use by programmers using the Microsoft Windows operating system. Windows Sockets is a public specification based on Berkeley UNIX sockets and aims to:

- Provide a familiar networking API to programmers using Windows or UNIX.
- Offer binary compatibility between heterogeneous Windows - based TCP/IP stack and utilities vendors.
- Support both connection - oriented and connectionless protocols.

If you are running an application that uses Windows Sockets, be sure to enable Windows Sockets when you configure Microsoft TCP/IP. If you are unsure whether any of your applications use Windows Sockets, check with your TCP/IP utilities application vendor.

## Related Topics:

To get a copy of the Windows Sockets specification via anonymous FTP:
To get a copy of the Windows Sockets specification from CompuServe®:

## To get a copy of the Windows Sockets specification via anonymous FTP:

1. Type

   **ftp microdyne.com**

2. Log in as *anonymous*.
3. Type your electronic   mail address for the *password*.
4. Type

   **cd /pub/winsock/winsock-1.1**

5. Choose the file with the format you want, ASCII (.TXT), PostScript® (.PS), or Microsoft Word for Windows (.DOC), and then type

   **get winsock.*ext***

***Or***

1. Type

   **ftp vax.ftp.com**

2. Log in as *anonymous.*
3. Type your e lectronic mail address for the *password*.
4. Type

   **cd /pub/winsockapi/winsock-1.1**

5. Choose the file with the format you want, ASCII (.TXT), PostScript (.PS), or Microsoft Word for Windows (.DOC), and then type

   **get winsock.*ext***

## To get a copy of the Windows Sockets specification from CompuServe®:

1. Type

   **go msl**

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. Send e lectronic   mail to *winsockapirequest@microdyne.com* to subscribe to this mailing list.

# Overview

This section describes how to install TCP/IP for Windows for Workgroups and configure the protocol on your workstation. The TCP/IP protocol family is not installed as part of the Windows for Workgroups product. This section assumes that Windows for Workgroups has already been successfully installed on your system.

# Before Installing Microsoft TCP/IP

You need to know the following information before you install Microsoft TCP/IP on your workstation:

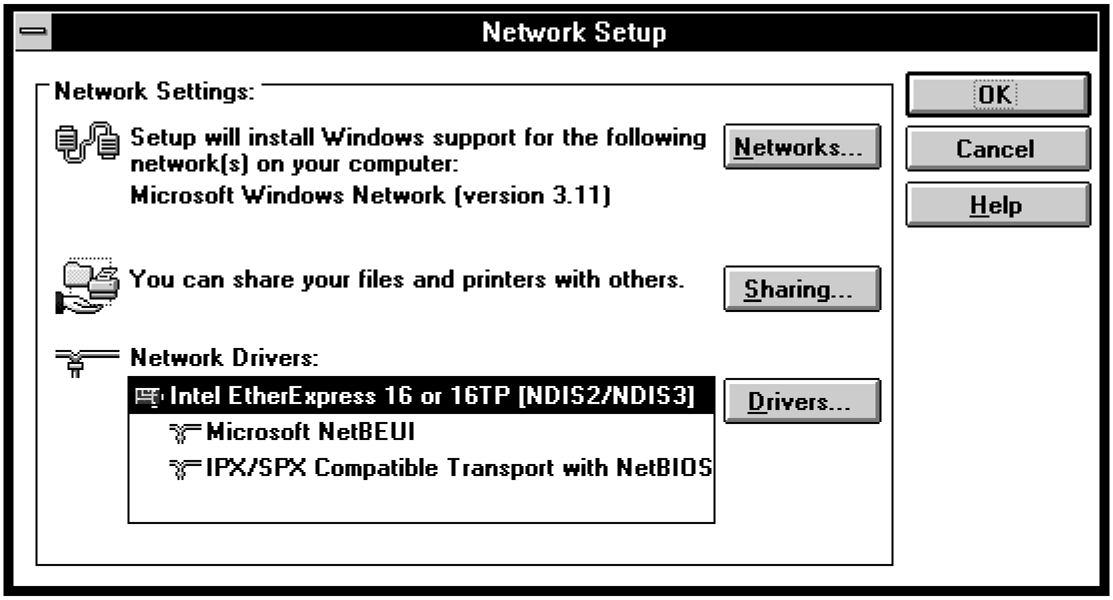
- Default gateway
- IP address
- Subnet mask
- Whether to enable Windows Sockets
- Whether to enable Domain Name Service (DNS) lookups (see "Networking Concepts for TCP/IP")

# Installing Microsoft TCP/IP

Before you install Microsoft TCP/IP on your workstation, be sure the Windows Setup is closed.

**To install Microsoft TCP/IP on a workstation with Windows for Workgroups**

1. Run Windows Setup from the Main program group and select Change Network Settings from the Options menu, or run Network Setup from the Network program group. The Network Setup dialog box appears:



Network Setup

Network Settings:

Setup will install Windows support for the following network(s) on your computer:
Microsoft Windows Network (version 3.11)

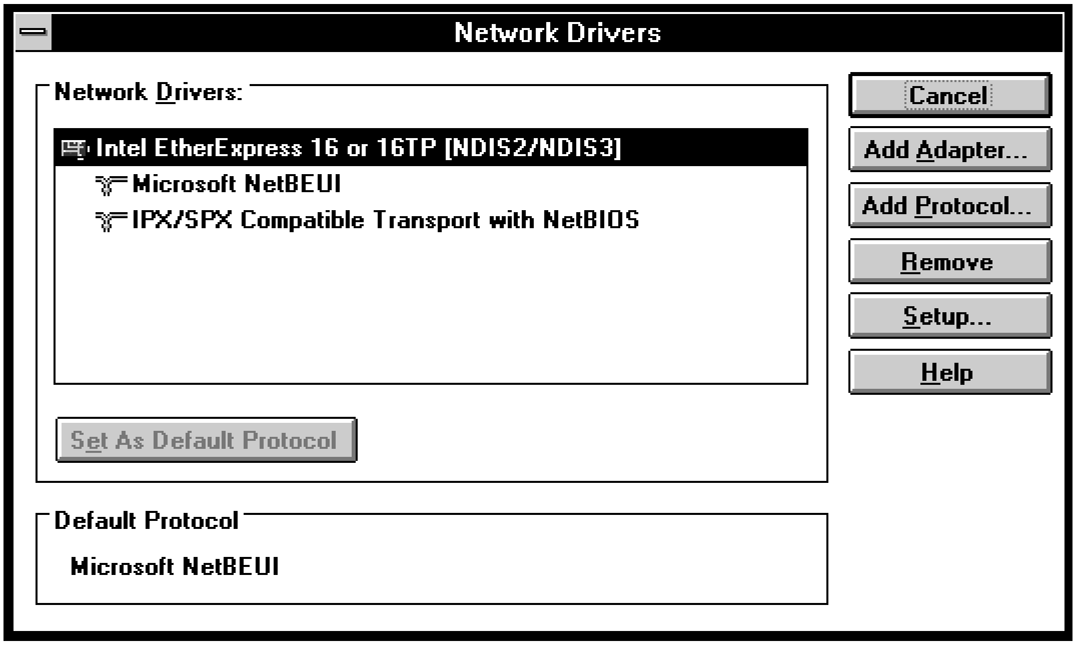
You can share your files and printers with others.

Network Drivers:

- Intel EtherExpress 16 or 16TP [NDIS2/NDIS3]
- Microsoft NetBEUI
- IPX/SPX Compatible Transport with NetBIOS

**Figure 2. Network Setup dialog box used to install network drivers**

2. Choose the Drivers... button. The Network Drivers dialog box appears:



Network Drivers

Network Drivers:

- Intel EtherExpress 16 or 16TP [NDIS2/NDIS3]
- Microsoft NetBEUI
- IPX/SPX Compatible Transport with NetBIOS

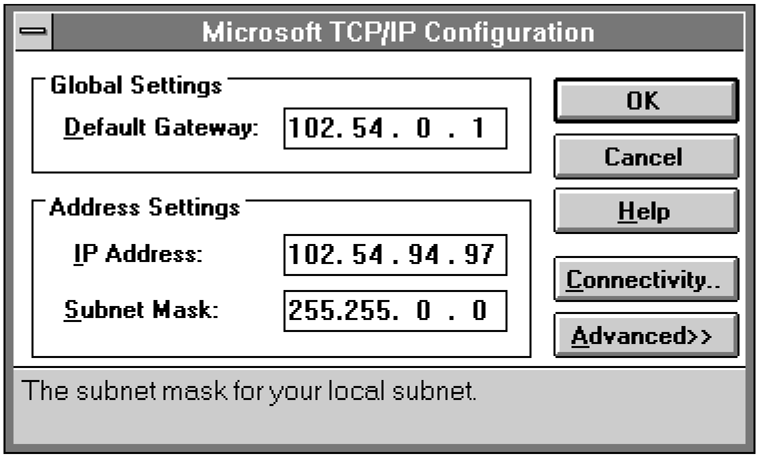
Default Protocol
Microsoft NetBEUI

**Figure 3. Network Drivers dialog box used to add additional protocols**

3. In the Network Drivers list box, select the adapter on which you want to run Microsoft TCP/IP.
4. Choose the Add Protocol... button. The Add Network Protocol dialog box appears. Note: You must have your network card set up as NDIS 2/NDIS 3 or you will receive an error message telling you to do so.
5. Select the Unlisted or Updated Protocol option from the list of available protocols and then choose OK. The Install Driver dialog box appears.
6. Insert the Microsoft TCP/IP for Windows for Workgroups disk in drive A (or drive B, as appropriate).
7. In the Install Driver dialog box, enter the location of the Microsoft TCP/IP for Windows for Workgroups disk (usually A:), and choose OK.

   The Unlisted or Updated Protocol dialog box appears.
8. Select Microsoft TCP/IP.
9. Choose OK.

The Microsoft TCP/IP for Windows for Workgroups software is now on the workstation's hard drive, and the Microsoft TCP/IP Configuration dialog box appears.



Microsoft TCP/IP Configuration

Global Settings

Default Gateway: 102. 54 . 0 . 1

Address Settings

IP Address: 102. 54 . 94 . 97

Subnet Mask: 255.255. 0 . 0

The subnet mask for your local subnet.

**Figure 4. Microsoft TCP/IP Configuration dialog box**

The Microsoft TCP/IP for Windows for Workgroups software is copied to your workstation. Continue with the configuration procedure, as described in the next section, "Configuring TCP/IP."

# Configuring TCP/IP

After the Microsoft TCP/IP protocol software is installed on your workstation, you *must* configure it with valid addressing information.

## To configure Microsoft TCP/IP on a workstation

1. In the Microsoft TCP/IP Configuration dialog box, enter values for the following parameters: Default Gateway, IP Address, Subnet Mask.

**Tip:** When entering IP addresses, you can use the space bar, the period key, the right arrow key, or the mouse to advance to the next field in the address. As in other dialog boxes, use the TAB key to advance to the next field in the dialog box. When your cursor is located in a field box, a "hint" for that field appears at the bottom of the dialog box.

**Default Gateway**
Specifies the IP address of the default gateway used to forward packets to other networks or subnets. This parameter is required only for nodes on internetworks. If this parameter is not provided, IP functionality will be limited to the local subnet. Your network administrator should provide you with the correct value for this parameter.

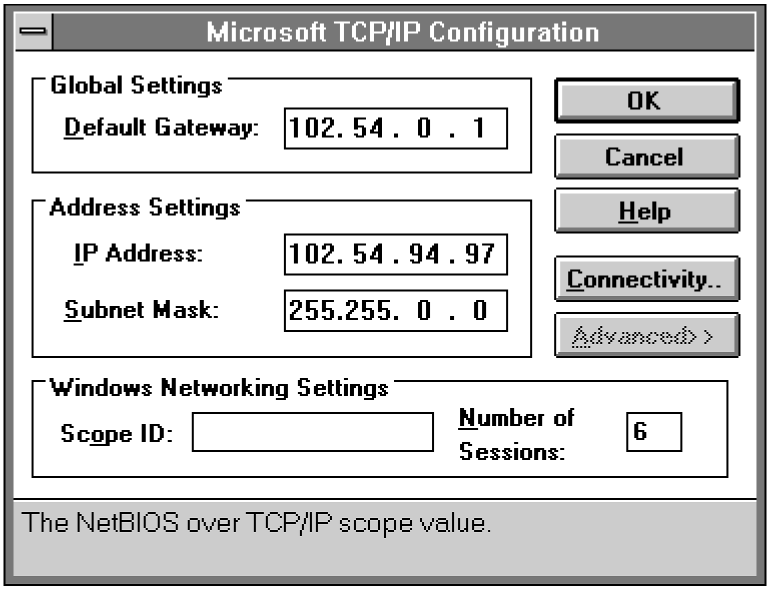
**IP Address**
Specifies the IP address associated with your local workstation. (For more information about IP addresses, see "Overview of Microsoft TCP/IP for Windows for Workgroups" and "Networking Concepts for TCP/IP".) Your network administrator should provide you with the correct value for this parameter.

**Caution:** Duplicate IP addresses on a network might cause some systems on the network to "hang" or function unpredictably.

**Subnet Mask**
Specifies the subnet mask associated with the adapter to which TCP/IP is bound. Each interface used by TCP/IP must have a subnet mask configured. This allows the workstation to separate the IP address into host and network IDs. The subnet mask defaults to the appropriate value shown in Table 3 (in the section "Networking Concepts for TCP/IP"). Your network administrator should provide you with the correct value for this parameter.

**Note:** The TCP/IP for Windows for Workgroups configuration program will check the validity of the IP address, default gateway, and subnet mask automatically. If you receive a message indicating that you have not configured these parameters properly, check with your network administrator or see "Networking Concepts for TCP/IP," for more information.

2. To set advanced NetBIOS over TCP/IP (NBT) parameters, choose the Advanced>> button. The dialog expands, as shown in Figure 5.



Microsoft TCP/IP Configuration

Global Settings

Default Gateway: 102. 54 . 0 . 1

Address Settings

IP Address: 102. 54 . 94 . 97

Subnet Mask: 255.255. 0 . 0

Windows Networking Settings

Scope ID:

Number of Sessions: 6

The NetBIOS over TCP/IP scope value.

Figure 5. Microsoft TCP/IP Advanced Configuration Dialog Box

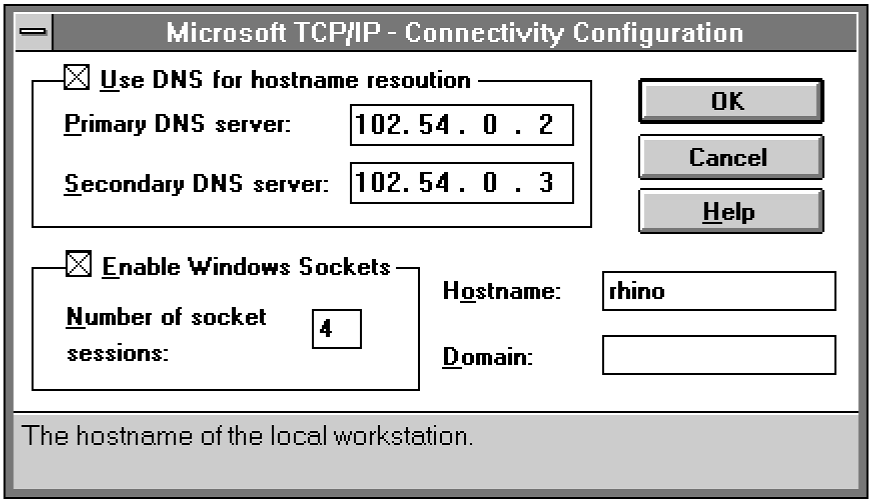
3. Type values for the Scope ID and Number of Sessions, as described below.

**Scope ID**
Specifies the NetBIOS scope parameter for the NBT module. To be able to communicate, all computers on a NetBIOS network must have the same scope ID. Your network administrator can provide you with the correct value for this parameter, but you can generally leave this value blank.

**Number of Sessions**
Specifies the number of simultaneous NBT sessions that your workstation can have. If a lot of people will connect to your workstation at one time, increase this value.

If Windows Sockets is enabled, the sum of the Number of Socket Sessions and the Number of Sessions values must be less than or equal to 22. The default Number of Sessions is 6.

4. Choose the Connectivity button to set options for the Microsoft TCP/IP utilities and other TCP/IP - based applications, such as Windows Sockets - based applications. A dialog box similar to that in Figure 6 appears.



Microsoft TCP/IP - Connectivity Configuration

☒ Use DNS for hostname resoution

Primary DNS server: 102. 54 . 0 . 2

Secondary DNS server: 102. 54 . 0 . 3

☒ Enable Windows Sockets

Number of socket sessions: 4

Hostname: rhino

Domain:

The hostname of the local workstation.

**Figure 6. Microsoft TCP/IP - Connectivity Configuration Dialog Box**

Setting these parameters allows you to specify remote TCP/IP nodes by their hostname rather than by their IP address. The local HOSTS file also facilitates this. (For details about hostname resolution, see "Understanding Name Resolution.")

**Use DNS for Hostname Resolution**

Determines whether or not to enable DNS (domain name service) hostname resolution. When selected, the DNR (domain name resolver) software is loaded at startup and is used to resolve hostnames, in conjunction with the local HOSTS file.

**Primary DNS Server**

Specifies the IP address for the primary DNS server that will be used to resolve hostnames. If DNS is not to be used when resolving domain names (when the Use DNS for Hostname Resolution box is cleared), this list has no effect and is disabled.

**Secondary DNS Server**

Specifies the IP address of the secondary DNS server used for hostname resolution. If DNS is not to be used when resolving domain names (when the Use DNS for Hostname Resolution box is cleared), this list has no effect and is disabled.

**Enable Windows Sockets**

Specifies whether or not the Windows Sockets interface will load at startup. Select this option only if you are running applications that use the Microsoft MS - DOS® sockets interface or Windows Sockets. (For a list of vendors and applications that use Windows Sockets, see the enclosed *Windows Sockets Vendor Information* card.)

**Number of Socket Sessions**

Specifies the number of sockets sessions that will be allocated at startup. If Window Sockets is not selected, this entry is disabled. If Window Sockets is selected, this parameter is required. The sum of the Number of Socket Sessions and the Number of Sessions parameters must be less than or equal to 22. The value of the Number of Sockets Sessions parameter can be 1 through 21. The default value is 4.

**Hostname**

Specifies the hostname for this computer. The hostname is used to identify the local workstation by name for authentication by utilities. Other TCP/IP - based utilities and applications can use this value to learn the name of the local workstation. This value defaults to the Windows for Workgroups computer name and it can be altered without affecting the computer name's value. The **Hostname** parameter is optional.

**Domain**

Identifies your group in the DNS hierarchical naming convention, with descending levels of detail. The fully qualified domain name (FQDN) for the workstation is the hostname followed by a period (.) followed by the domain name, for example, **rhino.microsoft.com**, where **rhino** is the hostname and **microsoft.com** is the domain name. During DNS queries, the local domain name is appended to short names. Specifying a **Domain** parameter is optional.

**Note:** The DNS domain is not the same as a LAN Manager domain.

5. When you are done setting connectivity values, choose OK.

   The Microsoft TCP/IP - Connectivity Configuration dialog box closes.

6. Choose the OK button to accept the configuration values you set and to close the Microsoft TCP/IP Configuration dialog box.

   Microsoft TCP/IP is now listed as a protocol under your network adapter card in the Network Drivers dialog box.

7. Choose the Close button.

The Network Drivers dialog box closes.

8. In the Network Setup dialog box, choose the OK button.

   A message appears, notifying you that your startup files have been updated.

9. Choose the OK button.

   A message box appears, notifying you that you must reboot for Microsoft TCP/IP to take effect.

10. To make changes to your system files, choose the Continue button before rebooting your workstation, or choose the Restart Computer button to reboot your computer and put Microsoft TCP/IP into effect on your workstation.

**Note:** If you change any of the TCP/IP parameters, exposed in the configuration dialogs or in the PROTOCOL.INI file, you *must* reboot your workstation for the change(s) to take effect.

## Related Topics:

If Microsoft RPC Is Installed

## If Microsoft RPC Is Installed

If Microsoft RPC is installed on your system, you must copy RPC16C3.DLL from the distribution diskette to your WINDOWS\SYSTEM directory in order for Windows Sockets to work properly with Microsoft TCP/IP for Windows for Workgroups.

# Troubleshooting IP Connections

If you have trouble installing Microsoft TCP/IP on your workstation, follow the suggestions in the error messages, or check your Windows for Workgroups documentation.

## Related Topics:

The Ping Utility

## The Ping Utility

The **ping** utility can isolate network hardware problems and incompatible configurations by allowing you to verify a physical connection to a remote computer. The syntax of the **ping** utility is:
**ping** *remote_computer* [**-t** [*timeout_value*]] [**-n** [*num_times*]]

where
*remote_computer*
Is the hostname or IP address of a remote computer.

**-t** [*timeout_value*]
Is the number of seconds that this node waits for an ICMP *echo reply* from a remote computer. The range is from 1 through 300 seconds; the default is 20 seconds.

**-n** [*num_times*]
Is the number of times **ping** sends an *echo request* to the remote computer. The default is 1 echo request.

**Note:** There is a one second delay between echo requests.

Use the **ping** utility to test both the hostname and the IP address of the host. If the IP address is verified but the hostname is not, you have a name resolution problem. In this case, be sure that the hostname you are querying is in either the local HOSTS file or the in DNS database. (For information about the HOSTS file, see "Understanding Name Resolution.")

### Related Topics:

If You Will Not Run PING.EXE

### *If You Will Not Run PING.EXE*

In order to run the ping utility, you must first run NMTSR.EXE. The nmtsr executable is a terminate-and-stay-resident (TSR) program that is loaded before the Windows operating system is started. It is started with an appropriately placed line in the AUTOEXEC.BAT file.

If, however, you will not run the ping utility and you want to remove NMTSR.EXE:

1. In your AUTOEXEC.BAT file, remove or comment out (add REM to the beginning of the line) the line that loads NMTSR.EXE.
2. Reboot your computer.

# Overview

This section describes how the HOSTS and LMHOSTS files and the **addname** utility work, and how to use them to access resources on a different TCP/IP network.

# Dynamic Resolution

TCP/IP connections are established based on IP addresses, but because users generally prefer to use names, a mechanism for mapping names to IP addresses is useful.

The domain name service (DNS) provides a way to look up name mappings when connecting your workstation to foreign hosts via applications such as FTP. To use the DNS, you must enable the domain name resolver (DNR) module on the Microsoft TCP/IP - Connectivity Configuration dialog box. (See "Installing and Configuring Microsoft TCP/IP" for details.)

NBT (NetBIOS over TCP/IP) provides a dynamic way for locating Windows for Workgroups - based hosts on the local network. (This mechanism is described in more detail in "Networking Concepts for TCP/IP.") For Windows for Workgroups - based systems located on remote subnets, the LMHOSTS file is needed to provide computername - to - IP address mappings.

## Related Topics:

Host Files
Addname

# Host Files

The HOSTS and LMHOSTS files contain lists of known IP addresses. Each of these files is also known as a *host table*. Both the LMHOSTS and HOSTS files are located in the WINDOWS directory and can be edited using any ASCII editor (such as EDLIN and EDIT, which are part of the MS - DOS operating system, version 5.00 and later).

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 workstation is changed, added to, or removed from the network.
- Because host table files are searched one line at a time from the beginning, list remote workstations in priority order, with the ones used most often at the top of the file. This arrangement increases the speed of searches for the most often used entries.

## Related Topics:

HOSTS
LMHOSTS

### *HOSTS*

If the DNR is not used or fails to match a hostname to an IP address, Microsoft TCP/IP searches the local host table file, HOSTS, for mappings of IP addresses to hostnames of remote computers. The HOSTS file is loaded into the workstation's memory when the workstation is started.

The HOSTS file format is the same as the format for host tables in the 4.3 BSD (Berkeley Software Distribution) UNIX */etc/hosts* file. For example, the entry for a node with an address of 192.102.73.6 and a hostname of MIS.HOST.COM looks like this:

```
192.102.73.6     mis.host.com
```

A sample HOSTS file is created when you install Microsoft TCP/IP for the Windows for Workgroups operating system. Edit this file to include remote hostnames and their IP addresses for each computer with which you will communicate.

### *LMHOSTS*

The LMHOSTS file is a local text file that maps IP addresses to NetBIOS names of remote servers with which you want to communicate.

For example, the host table file entry for a node with an address of 192.45.36.5 and a hostname of CPQ386 looks like this:

```
192.45.36.5      CPQ386
```

The LMHOSTS file format is the same as the format for host tables in the 4.2 BSD (Berkeley Software Distribution) UNIX */etc/hosts* file.

The LMHOSTS file is read when the workstation is started and stored in a system cache for later access. When a name must be resolved, NBT checks this cache before doing a b - node name discovery. (For details, see "Networking Concepts for TCP/IP.") The **addname** utility can be used to manipulate this cache.

# Addname

The **addname** utility temporarily adds entries to the NetBIOS name cache for use in your current work session (any new entries added will be deleted the next time you reboot your workstation). The syntax of the **addname** utility looks like this:

**addname** *computername ipaddress*

**addname** *computername* **/delete**

**addname** [**/load** | **/save**] [*filename*]

where

*computername*
Is the name of the remote server whose entry you want to add or delete. Computer names can have as many as 15 characters. These computer names do not affect the assignment of any local computer name or user name.

*ipaddress*
Is the IP address that corresponds to *computername*. If the *computername* is stored in the LMHOSTS file, the new IP address temporarily replaces the existing IP address.

**/delete**
Deletes the specified *computername* from the workstation's list of current entries but not from the LMHOSTS file.

**/load**
Loads a set of entries from the specified file. All current entries are deleted, and all entries in the specified file are added. If there is not enough room for all new entries in the file, an error is reported and no changes are made to the list of current entries. If you do not specify a filename, the default LMHOSTS file is used.

**/save**
Stores the list of current entries in the specified file. All entries in the file are overwritten by the current entries. If you do not specify a filename, the default LMHOSTS file is used. To delete an entry from the default **addname** configuration file, use **addname /save** after deleting the entry.

*filename*
Specifies the name of the file to load or save.

When you type **addname** with no options, the list of current **addname** entries appears.

# Overview

This section explains in some detail the various components of the Internet protocol suite, IP addressing, subnet masks, routing, and NetBIOS over TCP/IP. For additional information about any of the topics discussed here, see *Internetworking with TCP/IP, Volume I* by Douglas E. Comer (Prentice Hall, 1991).

# 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. It is used to connect the Internet − a worldwide internetwork connecting universities, research labs, Department of Defense installations, and corporations. (According to convention, "Internet" is capitalized when referring to the worldwide, connected internet.)

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 the integration with 4.2 BSD (Berkeley Software Distribution) 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 academic, defense, scientific, and commercial organizations share data, electronic mail, and services on the connected 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. Additionally, bridge, router, and network analyzer vendors all offer support for the TCP/IP protocol family within their products.

The following discussion introduces the components of the IP protocol suite. Although many of the details discussed are transparent to the user, knowledge of the architecture and interaction between the components is useful.

# Transmission Control Protocol (TCP) and Internet Protocol (IP)

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 of the other protocols within the TCP/IP family. It 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.

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 IP corrupts or loses a TCP/IP packet, TCP is responsible for retransmitting the faulty packet(s). This reliability defines TCP/IP as the protocol of choice for session - based data transmission, client - server applications, and critical services such as electronic mail.

This reliability does not come without a price. TCP headers require the use of additional bits to provide proper sequencing 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, TCP implements a throttle mechanism that allows the required frequency of ACKs to vary with the reliability of the data link. This permits highly reliable connections to use fewer ACKs and less computing power.

# User Datagram Protocol (UDP)

If reliability is not essential, the TCP complement, *user datagram protocol* (UDP), offers a connectionless datagram service that guarantees neither delivery nor correct sequencing of delivered packets (much like IP). Higher - level protocols or applications provide any reliability mechanisms in addition to UDP/IP. UDP data checksums are optional, providing a manner to exchange data over highly reliable networks without unnecessarily consuming processing time or network resources. 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 (ARP) and Internet Control Message Protocol (ICMP)

Although not directly related to the transport of user or application data, two other protocols in the IP suite perform important functions: *address resolution protocol* (ARP) and *internet control message protocol* (ICMP). These two protocols are maintenance protocols that support the IP framework and are generally invisible to users and to applications.

Although IP packets contain both source and destination IP addresses, the hardware address of the destination node must also be known. (This section assumes that the transmission type is Ethernet. Ethernet adapters contain a 48 - bit, globally unique address in permanent memory.) IP can acquire 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. Since 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.

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 or not a particular IP node on a network is functional. This is useful for diagnosing IP network or gateway failures.

# IP Addressing

Every host interface on a TCP/IP network is identified by an IP address. This address is used to identify a node on a network; it also specifies routing information in an internetwork. IP addresses are 32 - bit values typically represented in dotted decimal notation. Dotted decimal notation depicts each octet (or byte) of an IP address by its decimal value, separating each by a period, as in 102.54.94.97. IP addresses are used to provide nodes on a network with a unique address without relying on the underlying hardware to ensure unique addressing.

**Note:** Since IP addresses identify nodes on an interconnected network, each node on the internet *must* be assigned a unique IP address.

Although represented as a single value, IP addresses provide two pieces of information: the *network ID* and the *host ID* for a node. The network ID (which must be unique among all networks within a connected internet) specifies the network to which a node is attached. The host ID (which is unique among nodes within a network) identifies the node within its network. Networks that connect to the public Internet must obtain an official network ID from Defense Data Network - Network Information Center (DDN - NIC) to protect IP network ID uniqueness. Once assigned a network ID, the local network administrator must assign unique host IDs for computers within the network. Although private networks that are not connected to the Internet can choose to use their own network identifier, obtaining a valid network ID from DDN - NIC 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 is easily derived by the first octet (byte) of its IP address. Table 2 summarizes the relationship between the first octet of a given address and its host and network ID fields. It also identifies the total number of network and host IDs for each address class that participates in the Internet addressing scheme. This sample uses IP address w.x.y.z.

**Table 2. IP Address Classes**

| Class | w values (inclusive) | Net ID | Host ID | Available nets | Available hosts/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 |

**Note:** The network address 127 is reserved for loopback testing and interprocess communication on the local computer; it is not a network address.

A node uses the network and host IDs to determine which packets it should receive or ignore and to determine the scope of the transmissions it produces (only nodes with the same network ID accept one another's IP - level broadcasts). Since the sender's IP address is included in every outgoing IP packet, it is useful for the receiving node to derive the originating network and host ID from the IP address field. This is accomplished using *subnet masks*.

## Related Topics:

Subnet Masks
Routing

## 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 Table 3).

**Table 3. Examples of Subnet Masks for Standard IP Address Classes**

| Class | Default subnet mask |
|---|---|
| A | 255.0.0.0 |
| B | 255.255.0.0 |
| C | 255.255.255.0 |

Although configuring a host with a subnet mask might seem redundant after examining Table 3, 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. Table 2 shows that 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 effectively splits the Class - B address into 256 Class - C addresses: 144.100.0, 144.100.1, . . ., 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). Be sure to check with your local network administrator to determine your network's subnet policy and your correct subnet mask.

**Note:** It is important that all computers on a physical network use the same subnet mask and network ID; otherwise, addressing and routing problems can occur.

## Routing

When individual IP subnets are connected to an internet, IP gateways or IP routers are used to provide *routing* (packet delivery) between the networks. When a TCP/IP node attempts to communicate with a different network (when source and destination network IDs differ), a gateway (or a series of gateways) must forward the packet to the appropriate destination network. Gateways maintain routing tables that specify the *direction* (address of the next gateway) a packet should take to reach its destination, as well as a table of local hosts on the networks it interconnects.

Typically, gateways are IP *routers*, or computers with two or more network adapters that are running some type of IP routing software; each adapter is connected to a different physical network. On networks that are not part of an internet, IP gateways are not required. If a network is part of an internet and a node does not specify a default gateway (or the gateway computer is not operating properly), only communication beyond the local subnet is impaired. Currently, Microsoft TCP/IP recognizes only a single default gateway per node. That is, each TCP/IP node must rely on a single gateway to deliver packets to other networks. The network administrator can provide the address of the local gateway. The Microsoft TCP/IP installation software checks to ensure that the network ID for the default gateway matches the network ID of the local IP address. (IP routing and intergateway protocols are beyond the scope of this discussion.)

# TCP/IP with Windows for Workgroups

The Microsoft Windows for Workgroups operating system with integrated networking is based on a protocol - independent architecture. This architecture, illustrated in Figure 7, provides Microsoft Windows for Workgroup services over any network protocol that adheres to the *network basic input/output system* (NetBIOS) standard. The NetBIOS - compliant protocol(s) package application network requests in their respective format(s) and send the requests to the appropriate network hardware 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.

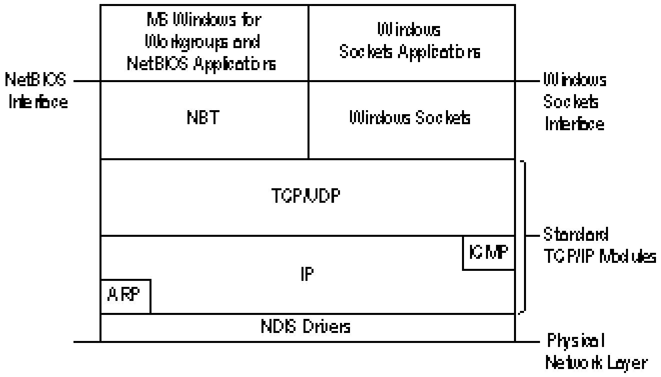


**Figure 7. Architectural Model of Windows for Workgroups and TCP/IP**

## Related Topics:

NetBIOS over TCP/IP
NetBIOS Name Resolution

## NetBIOS over TCP/IP

*NetBIOS over TCP/IP* (NBT) is a protocol module that provides NetBIOS naming services over TCP/IP. This layer allows NetBIOS applications, which establish connections and base data delivery on names, to function over TCP/IP, which relies on IP addresses. Using NBT, NetBIOS applications can locate other NetBIOS computers by name and simultaneously acquire the computers' IP addresses.

*Requests for comments* (RFCs), which are official documents that detail the IP protocol suite, provide a specification to do exactly this in RFC numbers 1001 and 1002. Microsoft TCP/IP complies with RFC 1001/1002. The current TCP/IP implementations are based on a *modified b - node* model, as described in the next section.

## NetBIOS Name Resolution

Two tasks are required of Windows for Workgroups - nodes running TCP/IP:

- Registration and release of NetBIOS names
- Discovery of NetBIOS names

Configuring the Windows for Workgroups operating system with TCP/IP requires *computername* and *IP address* information. (For more information about configuration, see "Installing and Configuring Microsoft TCP/IP.") The *computername* is the permanent NetBIOS name to which the workstation responds. The *IP address* is the unique address by which all other TCP/IP nodes on the network recognize the workstation or server.

Since both name and address must be unique, the following process is used. When NBT is initialized, a NetBIOS *name registration request* packet is broadcast to the network stating the NetBIOS name and IP address for the node. Any node on the network that has previously claimed the NetBIOS name is required to challenge the name registration with a *negative name registration response.* Such challenges result in an error during initialization of the NBT module. If the registration request is not challenged within a specific time period, the node adopts that name and address and proceeds. Successive NetBIOS name registrations proceed in the same manner.

When a node has finished with a particular NetBIOS name (such as when the Server service is stopped), it no longer challenges other registration requests for the name. This mechanism is referred to as *releasing* a NetBIOS name.

Since TCP/IP recognizes IP addresses and the Windows for Workgroups operating system recognizes names, it is necessary for a Windows for Workgroups - node running NBT to obtain the IP addresses of the computers with which it must communicate. This information is obtained during the *b* - *node NetBIOS name discovery process*. A packet containing the name of the destination computer (called the NetBIOS *name query request*, or discovery, packet) is broadcast over the network. The node that owns the name (assuming there is one) responds with a *positive name query response* packet containing its IP address, and the nodes are free to communicate via their IP addresses. This process is transparent to the user.

The discovery request is not foolproof. It is possible that b - node *name query request* packets will not be received by a functioning Windows for Workgroups - node running TCP/IP. This generally occurs when the two nodes are located on different subnets bridged by an IP router. The discovery packet is implemented as an IP broadcast, which is generally not forwarded by IP routers. Since the discovery packet cannot be heard by the destination node, it would appear that communication would be impossible. This is not the case, however. The LMHOSTS file and the **addname** utility can be used to specify NetBIOS name/IP address mappings. (For information about the LMHOSTS file, see "Understanding Name Resolution.")

# Overview

Microsoft TCP/IP for the Windows for Workgroups operating system is optimized for everyday users. The graphical installation and configuration process allows most users to configure their systems easily and effectively. If, however, you want to fine - tune options, you can change values in the Windows for Workgroups PROTOCOL.INI file with a text editor. You must then reboot for the changes to take effect. This appendix lists the available parameters that can be changed, as needed, in the **[tcpip]** section of the PROTOCOL.INI file.

**Caution:** Adjusting any PROTOCOL.INI parameters can severely degrade or impede system performance. Be sure that you fully understand the effect(s) your changes will have. You should back up your PROTOCOL.INI file before making any changes. In the event that a change "breaks" your system, revert to the defaults shown in Table 4 or use your backup settings.

Note that many of the parameters described in this appendix can be specified using the Network section of Windows Setup, as described in "Installing and Configuring Microsoft TCP/IP." In fact, using the Network option is the recommended method for updating values because the values you enter will be validated before they are modified; simply editing the PROTOCOL.INI file provides no validation.

# Parameter Definitions

Required entries in the **[tcpip]** section of the PROTOCOL.INI file are:

- **drivername = TCPIP$**
- **bindings**

Table 4 summarizes the possible entries and values in the **[tcpip]** section of the PROTOCOL.INI file:

**Table 4. TCP/IP PROTOCOL.INI Parameter Definitions**

| Entry | Units | Range | Default |
|---|---|---|---|
| **bcastaddr** | IP address | – | 255 255 255 255 |
| **bindings** | drivers | – | No default |
| **defaultgateway0** | IP address | – | No default |
| **drivername** | – | – | TCPIP$ |
| **forcepushbit** | integer | 0,1 | 0 |
| **ipaddress0** | IP address | – | No default |
| **lanabase** | integer | 0 - 255 | 0 |
| **maxlmhosts** | integer | 0-120 | 120 |
| **nbsessions** | integer | 1-22 | 6 |
| **netfiles** | path | – | *lanroot*\ETC |
| **numnames** | integer | 4 - 127 | 9 |
| **scope** | string | 64 character maximum | No default (Null) |
| **subnetmask0** | IP address | – | Default based on **ipaddress0** |
| **tcpconnections** | integer | 0 - 22 | No default |
| **tcpconntimeout** | seconds | 1-32767 | 30 |
| **tcpkeepalive** | seconds | 1-32767 | 600 |
| **tcpretries** | integer | 1-17 | 10 |
| **tcpsegmentsize** | bytes | – | 1450 |
| **tcpwindowsize** | bytes | – | 1450 |

The entries you are most likely to adjust (to enhance system performance) are **tcpconnections** and **tcpwindowsize**. Keep in mind as you adjust these entries that the more connections you have, the smaller the window size will be.

**Note:** In the PROTOCOL.INI file, IP addresses must be entered with spaces instead of periods as separators.

Entries in the **[tcpip]** section have the following meanings:

**bcastaddr**

Determines which IP address NBT uses to broadcast name requests and name queries. This entry is usually not needed since NBT uses the local IP address in conjunction with the subnet mask to determine a valid IP address to broadcast on. This parameter is used in cases where the network requires broadcasts to be issued on IP addresses 0.0.0.0 or 255.255.255.255. Keep in mind that it is possible to configure any IP address for NBT to use as a broadcast IP address, even one that is not a broadcast IP address. When such an address is used, the transport will treat it as a unique IP address and will send broadcast traffic directly to the IP address.

**bindings**
Binds information taken from the PROTOCOL.INI file to with the protocol and driver modules. This entry and value are supplied during installation by the Windows for Workgroups installation program.

**defaultgateway0**
Specifies the gateway used when the IP address is not on the local network.

**drivername**
Identifies the TCP/IP driver name. This entry must be **TCPIP$** (in all capital letters) and is set during installation by TCP/IP for Windows for Workgroups installation program.

**forcepushbit**
When set to 1, forces the stack to set the push bit on every outgoing packet. Setting this entry to 0, or not specifying this entry at all (default), sets the push bit only on packets as needed. If you are having trouble connecting to an IBM mainframe, try setting **ForcePushBit=1**.

**ipaddress0**
Identifies the IP address of the local Windows for Workgroups - based workstation.

**lanabase**
Determines which network adapter number applies to NBT. This entry is used only when more than one NetBIOS driver is loaded by the computer. This entry is set during installation by the Windows for Workgroups installation program.

**maxlmhosts**
Specifies the number of entries from the LMHOSTS file that should be loaded into the cache when the workstation is booted.

**nbsessions**
Specifies the maximum number of supported NetBIOS sessions. This entry should be set to the maximum number of servers the workstation will connect to plus the number of clients expected to connect to the local workstation.

**netfiles**
Identifies the path to all ASCII database files, such as HOSTS and LMHOSTS.

**numnames**
Specifies the maximum number of supported, local NetBIOS names that the workstation can register (for example, username, domain name, and computername).

**scope**
Specifies a character string that determines the NBT scope. The default is a null scope. This entry is used to interoperate with other NBT implementations that make use of the NBT scope. Before NBT transmits any packet that contains an NBT name, the NBT scope is first appended to the name. This includes packets such as name queries, name registrations, and session requests. On the receiving end, the NBT scope in any packet must match the locally configured NBT scope. If the scopes do not match, the packet will be ignored. Therefore, only computers that have the same scope can communicate with each other. The use of the NBT scope allows two computers on the network to have the same NBT name.

**subnetmask0**
Identifies the subnet mask, which masks the IP address.

**tcpconnections**
Specifies the total of NBT sessions, sockets sessions, and Telnet sessions for the computer. Adjusting this value can enhance system performance. Changing the **nbsessions** parameter adjusts this parameter automatically, and is the recommended method.

**tcpconntimeout**

Specifies the amount of time to wait (in seconds) before dropping an unresponsive connection.

**tcpkeepalive**
Specifies the interval, in seconds, between TCP level checks to make sure that a connection is still active.

**tcpretries**
Specifies the length of time your workstation will continue attempting to send a packet. The default, 10, corresponds to approximately 50 seconds. Higher values allow more time for repeated attempts to send the packet, up to a maximum of about six minutes.

**tcpsegmentsize**
Specifies the maximum amount of data (in bytes) that can be sent by the computer in a single packet. The value depends on the number of **tcpconnections**. For maximum memory conservation, set **tcpsegmentsize** to 1024. A large segment is 1450.

**tcpwindowsize**
Specifies the maximum amount of data (in bytes) that can be accepted by a workstation into its buffer. The value depends on the number of **tcpconnections** and on the network - adapter card. The minimum size is 512 bytes. To conserve memory, use a window size less than or equal to 4350. For best performance, set the window size to a multiple of **tcpsegmentsize**. The suggested multiple is 3 or 4, depending on whether **tcpsegmentsize** is 1450 or 1024, respectively. For maximum memory conservation, set **tcpwindowsize** to 1024.

**Note:** If you use a 3Com® EtherLink® card (3C501) (instead of an EtherLink II® card), set the window size equal to the segment size for all applications. Window and segment sizes must both be equal to either 1024 or 1450. Otherwise, performance could be seriously degraded.

# Overview

The installation program for Microsoft TCP/IP for Windows for Workgroups sets the parameters in the TCPUTILS.INI file, and usually you don't need to change them. This appendix lists the parameters that can be changed, as needed, in the TCPUTILS.INI file.

**Caution:** Adjusting any TCPUTILS.INI parameters can severely degrade or impede system performance. Be sure that you fully understand the effect(s) your changes will have. You should back up your TCPUTILS.INI file before making changes. In the event that a change "breaks" your system, revert to the defaults shown in Table 5 or use your backup settings.

Table 5 shows the TCPUTILS.INI sections and their functions in TCP/IP.

**Table 5. Sections of the TCPUTILS.INI File**

| Section | Function |
|---|---|
| SOCKETS | Sockets protocol driver |
| DNR | Domain name resolver protocol driver |
| TCPGLOBAL | Section with common TCP/IP entries shared by TCP/IP drivers |

The following sections provide information about each of these sections in TCPUTILS.INI.

# SOCKETS Section

The following entry in the **[sockets]** section of the TCPUTILS.INI file is required:

**drivername = SOCKETS$**

Table 6 summarizes the possible entries and values in the **[sockets]** section:

**Table 6. [SOCKETS] Section Parameter Definitions**

| Entry | Units | Range | Default |
|---|---|---|---|
| **drivername** | – | – | SOCKETS$ |
| **maxsendsize** | bytes | 32-2048 | 1024 |
| **numsockets** | integer | 1-31 | 4 |
| **poolsize** | bytes | 3200-28800 | 3200 |

Entries in the **[sockets]** section have the following meanings:

**drivername**
Identifies the sockets driver name. This entry must be **SOCKETS$** (in all capital letters).

**maxsendsize**
Specifies the maximum send size (in bytes) allowed on user datagram protocols (UDPs) or nonblocking TCP sends.

**numsockets**
Specifies the maximum number of sockets to be supported.

**poolsize**
Specifies the buffer size (in bytes) used by the sockets driver for nonblocking send calls. This entry is set when the system is initialized.

# DNR Section

The following entry in the **[dnr]** section of the TCPUTILS.INI file is required:

**drivername = DNR$**

Table 7 summarizes the possible entries and values in the **[dnr]** section:

**Table 7. [DNR] Section Parameter Definitions**

| Entry | Units | Range | Default |
|---|---|---|---|
| **drivername** | – | – | DNR$ |
| **domain** | string | Up to 116 characters | No default |
| **nameserver0** | IP address | – | No default |
| **nameserver1** | IP address | – | No default |

Entries in the **[dnr]** section have the following meanings:

**drivername**

Identifies the DNR driver name. This entry must be **DNR$** (in all capital letters).

**domain**

Identifies the TCP/IP domain name, which helps identify the workstation to other computers on the network. The domain name can contain as many fields as will fit within 116 characters. Each field must begin with an alphanumeric character and must be followed by letters, digits, or hyphens. Each field can have between 1 and 63 characters. Adjacent fields must be separated by periods.

**nameserver0**

Specifies the IP address of the primary domain server, which maintains a database of domain names.

Specifies the IP address of the secondary domain server.

# TCPGLOBAL Section

The following entry in the **[tcpglobal]** section of the TCPUTILS.INI file is required:

**username =** *username*

Table 8 summarizes the possible entries in the **[tcpglobal]** section:

**Table 8. [TCPGLOBAL] Section Parameter Definitions**

| Entry | Units | Range | Default |
|---|---|---|---|
| **hostname** | string | – | No default |
| **username** | string | – | No default |

Entries in the **[tcpglobal]** section have the following meanings:

**hostname**

Identifies the TCP/IP name of your workstation on the network.

**username**

Identifies the local name used to logon.

# Overview

This appendix lists the error messages that might appear when running Microsoft TCP/IP for the Windows for Workgroups operating system or when using the **ping** or **addname** utilities. Where appropriate, suggested recovery methods are provided for error messages.

# TCP/IP Stack Errors

**NET0100: Incorrect value for keyword detected in PROTOCOL.INI or TCPUTILS.INI file by TCP.**

Ensure that all values in the PROTOCOL.INI and TCPUTILS.INI files are valid and correct.

**NET0101: Value for keyword not found in PROTOCOL.INI or TCPUTILS.INI file by TCP.**

Enter a value in the PROTOCOL.INI or TCPUTILS.INI file for the keyword that is missing a value.

**NET0102: Cannot load TCP 1.0: incompatible DOS version.**

Ensure that your workstation is running MS - DOS 3.0 or higher.

**NET0103: Insufficient memory to allocate by TCP.**

Your workstation does not have enough memory to allocate for TCP.

**NET0104: Insufficient memory to initialize TCP.**

Your workstation does not have enough memory to initialize TCP.

**NET0106: Open failure on PROTOCOL.INI or TCPUTILS.INI by TCP.**

Check that no other application has these files open.

**NET0107: Read failure on PROTOCOL.INI or TCPUTILS.INI by TCP.**

Check that no other application has these files open.

**NET0108: Close failure on PROTOCOL.INI or TCPUTILS.INI by TCP.**

Check that no other application has these files open.

**NET0109: TCP is not loaded - detected by TCP.**

The sequence of events in your startup files is not correct. Re - install TCP/IP on your workstation.

**NET0110: Insufficient memory to load TCP 1.0**

Your workstation does not have enough free memory to load TCP/IP.

**NET0111: Error accessing NEMM.DOS. TCP 1.0 not loaded.**

The sequence of events in your startup files is not correct. Re - install TCP/IP on your workstation.

**NET0117: Incorrect PROTOCOL.INI or TCPUTILS.INI format detected by TCP.**

Ensure that your PROTOCOL.INI and TCPUTILS.INI files are ASCII files and of the format shown in Appendix A and Appendix B.

**NET0119: PROTOCOL.INI or TCPUTILS.INI file too large.**

Delete some blank lines and comments to shrink the offending file. You might need to reboot.

**NET0120: Logical driver name not found in PROTOCOL.INI or TCPUTILS.INI file by TCP.**

Ensure that the **drivername=** parameter is correctly specified in both files.

**NET0124: TCP/IP TSR module must be loaded before WINDOWS/386.**

Ensure that the TCP/IP commands in your AUTOEXEC.BAT file have not been rearranged.

**NET0125: NETBIND must be executed before TCP/IP TSR module is loaded.**

Ensure that the NETBIND command appears in your AUTOEXEC.BAT file.

**NET0131: TCP 1.0 must be loaded before the DOS 5.0 shell is started.**

Exit the DOS shell and load TCP/IP.

**NET0133: IP - address already in use.**

You have specified an IP address that is not unique. Contact your network administrator for a unique IP address for your workstation.

# Addname Errors

**Unable to access RFC NetBIOS.**

The network is loaded, but an attempt to open the NetBIOS driver failed.

**Unable to locate default configuration file.**

The default configuration file does not seem to be used anywhere.

**Invalid computer name: *computername*.**

Syntax error in computername supplied on command line.

**Computer name *computername* not currently configured.**

You have attempted to delete a name that has not been added.

**Invalid IP address: *ip_address*.**

Syntax error in IP address supplied on command line.

**Computer name *computername* already configured.**

Cannot add the name because it has already been added.

**Name table full. A name must be deleted first.**

You must first delete one of the names in the NetBIOS name table file before you can add another one.

**Invalid number of arguments. Type "addname ?" for help.**

Check the command syntax and try again.

**Invalid argument. Type "addname ?" for help.**

Check the command syntax and try again.

**The network is not loaded.**

TCP/IP is not loaded. If you just installed Microsoft TCP/IP, you must reboot before it will take effect.

**Unable to open file: *filename*.**

The specified file could not be opened.

**Error: Bad entry on line *line#* of file: *filename***

When reading a saved table, a bad name or IP address was found.

**Too many entries in file: *filename*. No changes were made.**

The listed file is full.

**Error when attempting to read file: *filename*.**

The listed file could not be read.

**Error when attempting to write file: *filename*.**

The listed file was not updated.

# Ping Errors

**Domain name server not responding.**

The server you listed for use in hostname resolution is unavailable. Check the IP address.

**Invalid value for - t option. Range must be between 1 and 300 seconds.**

Check the value you specified for the - **t** option.

**Invalid value for - n option. Range must be between 1 and 32767.**

Check the value you specified for the - **n** option.

**Too many parameters for the PING command.**

Check your syntax for the **ping** command.

**DGN0105: Network software (NMTSR.EXE) not loaded.**

The TCP/IP transport is not loaded. Be sure that Microsoft TCP/IP for Windows for Workgroups is correctly installed on your workstation.

**DGN0108: The Internet Protocol (IP) address is invalid.**

Check your syntax in the IP address supplied on the command line.

**DGN0200: Domain name resolver (DNRTSR.EXE) not loaded. Change directory to WINDOWS\DRIVERS and run DNRTSR.EXE.**

Be sure that the "Use DNS resolution" check box is selected in the Microsoft TCP/IP – Connectivity Configuration dialog box. You must reboot your computer for this option to take effect.

**DGN0217: Remote name cannot be resolved.**

Be sure that the hostname you specified in the Microsoft TCP/IP – Connectivity Configuration dialog box is spelled correctly and is also included in your HOSTS file or in the DNS database.

**DGN0219: Remote computer not responding.**

The remote computer you specified is unavailable or not running TCP/IP.