

Invisible LAN Operating System

Installation Manual

© Copyright 1995 Invisible Software, Inc.

Web:

www.invisiblesoft.com

Email:

invisoft@invisiblesoft.com

Invisible Software Program License Agreement

The programs in this package are copyrighted. By using them, you indicate acceptance of the following terms and conditions. If you do not agree with these terms and conditions, return the programs to the place of purchase; and your money will be refunded.

1. License. You may use the programs on a single local area network, and you may make a reasonable number of copies of the programs for backup purposes.

2. Restrictions. You may not use the programs on more than one local area network at the same time. You may not distribute copies of the programs or documentation to others, or place them into any electronic bulletin board or information retrieval system. You may not modify or reverse engineer the programs. You may not remove, alter, or circumvent any serialization or copy protection mechanism that is built in to the programs.

3. Limited Warranty on Diskettes. If a diskette in this package is defective, Invisible Software will replace it at no charge if the defective diskette is returned to Invisible Software, shipping prepaid, within 90 days of the date of purchase.

4. Limited Warranty on Software. Invisible Software warrants that the programs will operate substantially as described in the documentation. If you report a significant software defect in writing within 90 days of the date of purchase, and Invisible Software is not able to correct the defect, then you may return the programs and documentation to the place of purchase; and your money will be refunded.

5. Disclaimer. IN NO EVENT WILL INVISIBLE SOFTWARE BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAMS EVEN IF INVISIBLE SOFTWARE OR AN AUTHORIZED REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU. This agreement will be governed by the laws of the State of California.

6. 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. Contractor/manufacturer is Invisible Software, Inc.

TRADEMARKS

Invisible LAN, Invisible Network, Invisible Software, NET/30, NET/30 For Windows, Ultra Server, Invisible LAN Remote, Invisible Ethernet, Invisible RAM, Invisible RAM 386, Invisible EMS, TransBIOS, CACHE/30, and PopMaster are trademarks of Invisible Software, Inc.

IBM, IBM PC, PC/XT, PC AT, PS/2, Micro Channel, PC Network, and PC LAN Program are trademarks of International Business Machines Corporation.

PostScript is a trademark of Adobe Systems, Inc.

SCAT, PEAK, NEAT, and AT/386 are trademarks of Chips and Technologies.

Windows, MS-DOS, and EMM386 are trademarks of Microsoft Corporation.

LaserJet and PCL are trademarks of Hewlett-Packard Company.

LaserWriter is a trademark of Apple Computer.

Novell and Netware are trademarks of Novell, Inc.

QEMM-386 and DesqView are trademarks of Quarterdeck Office Systems.

386-MAX is a trademark of Qualitas.

Contents

Preface	ix
---------------	----

Part 1 — Basic Installation

Introduction to Invisible LAN	1-1
Network Operating Systems	1-1
Components of Invisible LAN	1-3
Servers and Redirectors	1-5
Print Spoolers and Print Queues	1-5
Memory Management	1-6
Disk Cache	1-7
Remote Program Load Servers and Diskless Workstations	1-8
Electronic Mail	1-9
Command Line Utilities	1-9
Network Management	1-10
Windows Driver and Network Manager	1-11
Bulletin Board	1-13
NetBIOS and TransBIOS	1-14
Ultra Server	1-14
Remote Login and Remote Bridging	1-15
Network Configuration	2-1
Starting the Setup Program	2-1
Copying the Software onto the Hard Disk	2-2
Configuring the Software	2-4
Modifying CONFIG.SYS	2-9
Exiting from Setup	2-11
Starting the Network	3-1
Starting Invisible LAN	3-1
Using Network Disks and Printers	3-1
Network Disks	3-2
Network Printers	3-2
Discontinuing a Network Disk	3-3
Discontinuing a Network Printer	3-3
Displaying Mappings	3-3

Getting Help With Invisible LAN Commands	3-4
Changing the Configuration	4-1
Changing the Network Configuration	4-1
Multiple Configurations	4-2
Using Multiple Configuration Files	4-2
Creating Multiple Configuration Files	4-2
Windows Driver Installation	5-1
Installation Overview	5-1
Installing Windows	5-2
Installing Invisible LAN	5-2
Installing the Invisible LAN Windows Driver	5-3
Starting the Software	5-4
Memory Usage	6-1
Using Shadow RAM	6-1
Loading Invisible LAN Into Shadow RAM	6-1
Memory Manager Requirements	6-2
Loading SHARE Into Shadow RAM	6-2
Shadow RAM and Windows	6-3
Using Expanded Memory (EMS)	6-3
Loading Invisible LAN Into Expanded Memory	6-3
Memory Manager Requirements	6-4
Using Expanded Memory With Shadow RAM	6-4
HIMEM.SYS	6-5
Loading DOS High	6-5
Using the Disk Cache with HIMEM.SYS	6-5
Configuration 1: Cache Data Location is EXTENDED	6-6
Configuration 2: Cache Data Location is XMS	6-6

Part 2 — Advanced Installation

Setting Up Invisible LAN	7-1
Starting Invisible LAN	7-1
Creating an Initialization File	7-2
Step 1: Enter the File Name	7-3
Step 2: Select Parameters	7-3
Step 3: Save the File	7-4

Setting Up CONFIG.SYS	7-5
Invisible LAN Device Drivers	7-5
Disk Buffers	7-6
Drive Letters	7-6
File Control Blocks and File Handles	7-7
System Stack Resources	7-8
Advanced Network Configuration	8-1
General Parameters	8-2
TransBIOS Parameters	8-3
RAM Versus ROM	8-4
Memory Allocation	8-4
Timing	8-5
Resource Allocation	8-6
Hardware Configuration	8-6
Packet Size	8-7
NetBIOS Parameters	8-8
RAM Versus ROM	8-9
Memory Allocation	8-9
Resource Allocation	8-9
Server Parameters	8-10
Memory Allocation	8-11
Resource Allocation	8-11
Remote Program Load	8-12
Security Options	8-13
UPS Monitoring	8-13
Pipeline Parameter	8-14
Print Spooler Parameters	8-15
Defining Printers	8-15
Print Queue Configuration	8-17
Redirector Parameters	8-18
Memory Allocation	8-18
Resource Allocation	8-19
Redirector Cache Allocation	8-20
Miscellaneous Parameters	8-20
Drive Mapping Parameters	8-21
Printer Mapping Parameters	8-22
Cache Parameters	8-23
Memory Allocation	8-24
Fine Tuning	8-25
Extended Memory Access	8-26
SHARE Parameters	8-27

Memory Allocation	8-27
Resource Allocation	8-28
Mail Parameters	8-29
Memory Allocation	8-30
Ultra Server Parameters	8-30
Cache Configuration	8-31
Memory Usage	8-33
Determining What Memory Ultra Server Can Detect	8-33
Controlling the Use of Detectable Memory	8-34
Getting Ultra Server to Use Undetectable Memory	8-35
Drive Caching Options	8-36

Guide to Configuring Invisible LAN 9-1

Optimizing Performance	9-1
Minimizing Memory Usage	9-3
Large Networks	9-6

Advanced Windows Driver Installation 10-1

Components of Invisible LAN	10-1
Invisible LAN For DOS	10-2
Invisible LAN For Windows	10-2
Configuring Invisible LAN	10-3
Enabling Windows Support	10-3
Disk Cache Configuration	10-4
Shadow RAM and Expanded Memory Utilization	10-5
Invisible RAM Memory Managers	10-5
Server Configuration	10-6
Configuring Windows	10-6
Stand-Alone Windows Installation	10-7
Installing Windows	10-7
Installing the Invisible LAN Windows Driver (Quick Method) . . .	10-8
Installing the Invisible LAN Windows Driver (Automatic Method)	10-8
Installing the Invisible LAN Windows Driver (Manual Method) .	10-10
Shared Windows Installation	10-11
Copying Windows to the File Server	10-12
Setting Up Windows on a Workstation	10-13
Special Requirement for Serial Network Printers	10-14
Special Requirement for WINSTART.BAT	10-15
Special Requirement for TransBIOS ROM	10-15
Installing Invisible LAN Wallpaper (Optional)	10-16
Selecting Printer Drivers	10-16

Part 3 — Invisible RAM Memory Management

Introduction to Invisible RAM	11-1
What Invisible RAM Does For You	11-1
Hardware Requirements	11-2
Memory Configuration Information	11-3
Conventional Memory	11-3
Extended Memory	11-4
Expanded Memory	11-4
Shadow RAM And High RAM	11-6
Memory Hardware Configuration	12-1
The NEAT Chipset	12-1
Determining Total RAM	12-2
If You Have Exactly 1024K Total RAM	12-2
Option 1 — Shadow RAM	12-3
Option 2 — Expanded Memory	12-3
If You Have More Than 1024K Total RAM	12-4
Option 1 — No Expanded Memory	12-4
Option 2 — Expanded Memory	12-5
Set-Up Program Problems	12-5
The AT/386 Chipset	12-6
SCAT Chipset Configuration	12-7
Determining Total RAM	12-7
If You Have Exactly 1024K Total RAM	12-8
Option 1 — Shadow RAM	12-8
Option 2 — Expanded Memory	12-8
If You Have More Than 1024K Total RAM	12-9
PEAK Chipset Configuration	12-10
386 Virtual Mode	12-10
Microprocessor Type	12-11
Determining Extended Memory Size	12-11
Check Extended Memory Applications	12-11
Invisible EMS	12-12
Generic EMS	12-12
Install the Underlying EMS	12-12
Check the Memory Map	12-13
Checking for Frontfill	12-14
Using an "Include" Parameter	12-15
Using VGA2CGA.SYS	12-15

Advanced Memory Configuration	13-1
Page Frames	13-2
Memory Manager Parameters	13-3
Memory Manager	13-4
Configuration	13-4
Standard EMS Area Control	13-6
Frame Address	13-6
Frame Size	13-6
DOS Memory Control	13-7
Frontfill Size	13-7
Code Location	13-8
Initial VGA State	13-8
Internal Expanded Memory Parameters	13-8
Handles	13-8
Contexts	13-9
Non-Standard Page	13-9
Extended Memory Emulation	13-10
Hardware Configuration and Test	13-10
Test Level	13-10
I/O Port	13-11
System Memory Configuration	13-11
EXCLUDE	13-13
DEFAULT_RAM	13-14
INCLUDE	13-15
SHADOW_RAM	13-16
HIGH_RAM	13-17
ROM_COPY	13-17
386 Control Program Parameters	13-18
Extended Memory Allocation	13-19
High Memory Area XMS Interface	13-19
EMS Alternate Register Sets	13-20
Interrupt Stack Frame Allocation	13-21
BIOS ROM configuration	13-21
DMA Data Buffer Allocation	13-22

Part 4 — Appendices and Index

Invisible LAN CONFIG.SYS Reference	A-1
---	------------

Index	B-1
--------------------	------------

Preface

This manual tells you how to install the Invisible LAN network operating system software.

- Part 1 describes the basic installation procedure. You'll learn how to install the network software in the most commonly-used configurations, and how to establish a connection between a workstation and a file server. You'll also learn the basic procedure for installing the Windows driver, and how to use Invisible LAN in conjunction with memory management software.
- Part 2 describes the advanced installation procedure. You'll learn about all the installation options of the Invisible LAN software, and how you can configure the software to your exact requirements. You'll also learn about all the available options for installing the Windows driver.
- Part 3 describes the Invisible RAM memory managers. This memory management software gives you extra memory by increasing the size of DOS memory, creating "shadow RAM" that you can use for memory-resident programs, and providing expanded memory. You'll learn how to install and configure the Invisible RAM software.
- Part 4 contains appendices and a comprehensive index.

This manual does not describe how to operate the Invisible LAN software after it is installed. Refer to the *DOS User Manual* or *Windows User Manual* for operating instructions.

Part 1

Basic Installation

Introduction to Invisible LAN

Invisible LAN is a network operating system. This chapter presents the basic concepts of a network operating system and some of the specific features of Invisible LAN. Even if you are a computer expert, you should read this chapter to get an overview of Invisible LAN.

The topics covered in this chapter include:

- Network operating systems
- Components of Invisible LAN
- Servers and redirectors
- Print spoolers and print queues
- Memory management
- Disk cache
- Diskless workstations and the Remote Program Load Server
- Electronic mail
- Command line utilities
- Network management
- Windows driver and Network Manager
- Bulletin board
- NetBIOS and TransBIOS
- Ultra Server
- Remote login and remote bridging

Network Operating Systems

A *network operating system* is an extension of DOS. DOS provides a variety of functions for controlling the disk drives and printers on a single computer.

These functions include reading and writing disk files, organizing files into directories and subdirectories, printing files, and executing programs. A network operating system extends these functions to operate over a network.

For example, suppose that your computer has a floppy disk drive, a fixed disk drive, and a printer. In this case, DOS refers to the floppy drive as A:, the fixed disk as C:, and the printer as LPT1. DOS also organizes the files on the floppy and fixed disks.

Here are examples of some simple commands and the actions DOS takes to perform them:

COPY C:MYFILE.TXT A:

DOS finds the file named MYFILE.TXT on the fixed disk and reads it into main memory. It then writes the file onto the floppy disk.

COPY A:ACCTS.DOC LPT1

DOS reads the file named ACCTS.DOC from the floppy disk, then sends it to the printer.

TYPE C:LETTER.TXT

DOS reads the file named LETTER.TXT from the fixed disk, then displays it on the screen.

Now suppose that your computer is connected to a network that is running the Invisible LAN operating system. In this case, you may want D: to refer to another computer's fixed disk, and LPT2 to refer to another computer's printer. These are called *mappings*. Your mappings would be as follows:

A: ==> Your floppy disk
C: ==> Your fixed disk
D: ==> Another computer's fixed disk
LPT1 ==> Your printer
LPT2 ==> Another computer's printer

DOS controls A:, C:, and LPT1. Invisible LAN controls D: and LPT2. Here are examples of some simple commands and the actions that DOS and Invisible LAN take to execute them:

COPY D:MYFILE.TXT C:

Invisible LAN reads the file named MYFILE.TXT from the other computer's fixed disk. Then DOS writes the file onto your fixed disk.

COPY A:ACCTS.DOC LPT2

DOS reads the file named ACCTS.DOC from your floppy disk, then Invisible LAN sends it to the other computer's printer.

TYPE D:LETTER.TXT

Invisible LAN reads the file named LETTER.TXT from the other computer's fixed disk. DOS then displays the file on the screen.

COPY D:OLDFI D:NEWFI

Invisible LAN reads the file named OLDFI from the other computer's fixed disk. Invisible LAN then writes a copy of the file onto the other computer's fixed disk, and names the copy NEWFI.

As you can see, you access the other computer's disk or printer with the same commands you use to access your own disk or printer. DOS and Invisible LAN work together to execute your commands. DOS handles the portion of the command that refers to your own disk or printer, and Invisible LAN handles the portion of the command that refers to the other computer's disk or printer.

In this example, we say that A: and C: are *local disks*, because they are attached to your own computer. We say that D: is a *network disk*, because it is attached to another computer and accessed across the network. Similarly, we say that LPT1 is a *local printer*, and LPT2 is a *network printer*.

SUMMARY: Invisible LAN extends the functions of DOS onto the network. When Invisible LAN is installed, you use standard DOS commands to access both your own devices, and devices attached to other computers on the network.

Components of Invisible LAN

Invisible LAN includes a wide variety of functions. When you use Invisible LAN, you select which components you want to use and which functions you want to perform. You can select just the basic network functions, or you can select advanced functions such as security, memory management, network management, and disk caching.

The components included with Invisible LAN are:

- 1. Redirector** — A program that allows your computer to use disks and printers that are attached to other computers on the network.

- 2. Server** — A program that allows other computers on the network to use your disks and printers.
- 3. Print Spooler** — Stores up files waiting to be printed, and sends them to the printer one-by-one.
- 4. Memory Managers** — A collection of programs that let you utilize your computer's memory management capabilities to increase the amount of memory available to DOS applications.
- 5. Disk Cache** — A program that speeds up hard disks by storing disk data in the computer's memory.
- 6. Remote Program Load Server** — Supports the use of diskless workstations on the network.
- 7. Electronic Mail** — Sends and receives messages to other users on the network.
- 8. Command Line Utilities** — A variety of simple programs that let you control the operation of the network.
- 9. Network Manager** — A menu-driven program that lets you manage mappings, security, server usage, and other functions on the network.
- 10. Windows Network Driver** — A program that allows Microsoft Windows to function on the network, extending Windows onto the network in much the same way that Invisible LAN extends DOS onto the network.
- 11. Windows Network Manager** — A graphical "point-and-click" program for Microsoft Windows that lets you manage mappings, security, server usage, mail, and other functions on the network.
- 12. Bulletin Board** — A store-and-forward mail system that doubles as an electronic forum for exchanging ideas and information.
- 13. NetBIOS** — A standard network communication protocol that is used by many network application programs.
- 14. TransBIOS** — A special high-performance network communication protocol that is used by Invisible LAN.
- 15. Ultra Server** — A high-performance version of the Invisible LAN file server, that is used mainly for dedicated servers.
- 16. Remote Login Client** — A program that allows your computer to connect to the network through a modem.
- 17. Remote Login Host** — A program that runs on one computer in the network and transfers data between the network and a modem.

You can use the various components of Invisible LAN in any combination you want. All the components can be adjusted to match your exact requirements. The end result is a very flexible system that can be configured for any desired combination of performance, functionality, and memory usage.

Invisible LAN includes a menu-driven Setup program called **SETUP30**. The Setup program provides a convenient way to specify which components you want to use, and how to configure each component.

SUMMARY: Invisible LAN includes a variety of components that perform a wide range of functions. You can select which components to use, and tailor the system to your exact needs.

Servers and Redirectors

The server and the redirector are complementary programs. The *server* lets other computers on the network use your disks and printers. The *redirector* lets your computer use disks and printers that are attached to other computers.

Your computer can be both a server and a redirector at the same time. In this case, you can use other computers' disks and printers, while other computers can simultaneously use your disks and printers.

In Invisible LAN, you can elect to use the server, or the redirector, or both.

SUMMARY: A *redirector* can use disks and printers attached to other computers. A *server* makes its disks and printers available to other computers.

Print Spoolers and Print Queues

Suppose that your computer is a server, and that other computers can use your printer. Now suppose that two other computers both send documents to your printer at the same time. Do the documents get mixed together as they print?

The answer is no. All documents sent to your printer are handled by a special program called a *print spooler*. When the print spooler receives a document, it does not send the document directly to your printer. Instead, it stores the document in a temporary file on your disk. When the entire document has been copied into the temporary file, then the print spooler sends the temporary file to your printer.

If the print spooler receives two documents at the same time, they are stored in individual temporary files. The print spooler first sends one temporary file

to your printer, advances the paper to the top of the page, then sends the other temporary file to your printer. The documents do not get mixed together as they print.

When there are several documents waiting to be printed, they line up in the *print queue*. Newly arriving documents go to the end of the line. The document at the head of the line is sent to your printer. When the document is finished printing, then the next document in line is printed.

In Invisible LAN, the print spooler is built into the server. If you allow other computers to use your printers, the print spooler function is provided automatically.

SUMMARY: The *print spooler* stores a document in a temporary disk file, and then sends the temporary file to the printer. If there are several documents to be printed, they wait in the *print queue*, so they can be printed sequentially without getting mixed together.

Memory Management

One of the unique features of Invisible LAN is its extensive memory management capabilities. Invisible LAN includes Invisible Software's complete line of Invisible RAM™ memory managers, which are also sold as stand-alone utilities.

The memory managers are an optional part of Invisible LAN. They give you the following capabilities:

- Increase the size of DOS memory from the normal 640K up to as much as 736K.
- Load memory-resident programs and device drivers outside of DOS memory, so that they do not use up memory that you need for your application programs. This works with both the network software, and other memory-resident programs.
- Provide expanded memory, compatible with the expanded memory specification (EMS) version 4.0. The network software can be loaded into the expanded memory.

The end result is that after installing the Invisible LAN network operating system, you may have more DOS memory available than before! In addition, the memory managers are extensively configurable, so you can create the exact memory configuration that you want.

Most computers already have memory hardware that can be used by the Invisible RAM memory managers. The following types of hardware are supported:

- Any computer with a 386 or 486 microprocessor, and at least 2MB of RAM.
- Any computer with a 286, 386, or 486 microprocessor, and one of the following four chipsets made by Chips and Technologies: the NEAT chipset, the AT/386 chipset, the SCAT chipset, or the PEAK chipset.
- Any computer with an EMS 4.0 hardware expanded memory board, such as the Invisible EMS board.

Of course, if you don't have any of the above memory hardware, you can still install the network software into DOS memory. Invisible LAN is designed to use the smallest possible amount of DOS memory.

In addition to the Invisible RAM memory managers, Invisible LAN is also compatible with other memory managers, including Microsoft's EMM386, Quarterdeck's QEMM-386, and Qualitas' 386-MAX. So, regardless of whether you use one of the Invisible RAM memory managers or another memory manager, Invisible LAN automatically makes the best possible use of your memory.

SUMMARY: The *Invisible RAM* memory managers support a wide range of memory hardware. They increase the size of DOS memory, load memory-resident programs outside of DOS memory, and include an expanded memory manager. The network software can utilize the memory created by the Invisible RAM memory managers, as well as other popular memory managers.

Disk Cache

A *disk cache* is a program that speeds up the hard disk.

The basic idea of a disk cache is simple. Whenever information is read from the hard disk, the disk cache program keeps a copy of the information in memory. Then, the next time the information is needed, it can be obtained from memory instead of the disk. Since memory is much faster than the hard disk, the end result is that information can be obtained much more quickly.

The disk cache is an optional part of Invisible LAN. The cache is especially valuable on a file server, because all workstations using the file server will benefit from the increased speed. In fact, most network operations are twice as fast when the disk cache is installed on the server.

SUMMARY: The *disk cache* speeds up the hard disk. It is designed especially to speed up file server performance.

Remote Program Load Servers and Diskless Workstations

Some networks include diskless workstations. A *diskless workstation* is a personal computer with no disk drives, floppy disk or hard disk. The diskless workstation has the same keyboard, screen, and central processing unit as a regular computer.

But what use is a computer without disks? Recall that with Invisible LAN, you can make some of your drive letters refer to disk drives attached to other computers. If your computer is a diskless workstation, you make all your drive letters refer to disks on other computers. Then you can run the same programs as a regular computer; the only difference is that whenever you read or write a disk file, you access a disk that is connected to another computer on the network.

When you start up a computer, it usually reads in DOS from the disk drive. Since a diskless workstation has no disk drives, it requires a different procedure to start up. A diskless workstation reads in DOS from a network computer called the *Remote Program Load Server*.

A Remote Program Load Server is a special type of server that makes DOS available to other computers on the network. When you start a diskless workstation, it automatically contacts the Remote Program Load Server and reads in DOS from the Remote Program Load Server.

In Invisible LAN, the Remote Program Load Server function is built into the server. If you configure your server as a Remote Program Load server, then diskless workstations can load DOS from your server.

SUMMARY: A *Remote Program Load Server* makes a copy of DOS available to other computers on the network. *Diskless workstations* can read this copy of DOS and start up without the use of a disk drive.

Electronic Mail

Invisible LAN includes the capability to transmit and receive messages over the network. The *mail* program is resident on your computer but is not visible until you press a special *hot key* to make it active. You compose a message to another network user and send it via the network.

When someone else sends you a message, the mail program on your computer pops up and displays the message. It then gives you options for responding to the message.

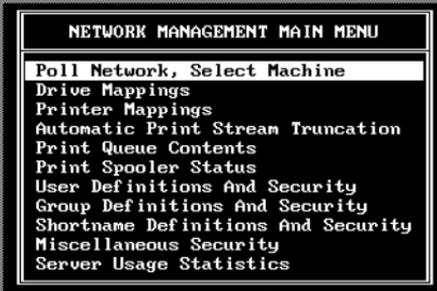
SUMMARY: The *electronic mail* system in Invisible LAN allows you to transmit messages to and receive messages from other network users.

Command Line Utilities

Invisible LAN includes a variety of useful utilities that can be run from the DOS command line, or from a batch file. Here are examples of some of the utility programs:

- **MAP** — Controls your drive and printer mappings. You can display your current mappings, create new mappings, and discontinue existing mappings.
- **NETCLOCK** — Obtains the current date and time from a server, and then sets the clock in your computer.
- **TRUNCATE** — Establishes a timeout for network printers.
- **PSETUP** — Establishes a setup procedure for network printers.
- **PQ** — Controls a server's print queue. It can display the status of the queue, pause printing, cancel print jobs, control page feeds, and set the printing priority.
- **NW** — Network software control. This program lets you pause the server, hide the network software, and control the use of the speaker.
- **CACHE** — Cache control. You can display the status of the disk cache, and enable or disable the cache.

SUMMARY: The command line utilities perform a variety of useful functions, and can be used from the DOS command line or a batch file.



Lists all the network stations, and lets you select a station to manage
HELP F1 **ACTIVATE** Enter **CANCEL** Esc **SELECT** ↑ ↓ PgUp PgDn Home End A-Z

Figure 1-1. Network Manager Main Menu

Network Management

To facilitate management of the network, Invisible LAN provides a menu-driven *Network Manager* program. The manager's Main Menu is shown in figure 1-1. The functions on the Main Menu are:

Poll Network, Select Machine — Lists the computers on the network and allows you to select a computer to manage.

Drive Mappings — Displays drive mappings of the selected computer and allows you to change the mappings on the local computer.

Printer Mappings — Displays printer mappings of the selected computer and allows you to change the mappings on the local computer.

Automatic Print Stream Truncation — Provides a means for automatically printing files created by applications that do not signal the end of a print file.

Print Queue Contents — Lists the files in the print queue, and lets you rearrange or delete the files.

Print Spooler Status — Controls various parameters of the print spooler.

User Definitions and Security — Lets you define network users.

Group Definitions and Security — Lets you define groups of network users, and control which users belong to each group.

Shortname Definitions and Security — Provides a way to define shortnames (shorthand references to particular directories) and to control access to them.

Miscellaneous Security — Provides a means to control various security aspects of Invisible LAN not covered on other menu choices on the manager Main Menu.

Server Usage Statistics — Lets you monitor the performance of the file server and disk cache.

SUMMARY: The *network manager* provides a menu-driven means for managing the Invisible LAN.

Windows Driver and Network Manager

One of the outstanding features of Invisible LAN is the way it supports Microsoft Windows. Invisible LAN actually extends Windows so that it operates smoothly on the network. In addition, Invisible LAN includes a graphical network management program that lets you control all aspects of network operation from within Windows.

The *Windows network driver* is the program that extends Windows so that it can operate on the network. All Windows utilities and applications can access network disks, network printers, and other network resources. The network driver supports all three modes of Windows — real mode, standard mode, and 386 enhanced mode.

When Windows is running in 386 enhanced mode, the network driver even allows you to create *virtual network stations*. Each “virtual” network station acts like an independent workstation, with its own drive mappings, printer mappings, user name, and security permissions. With this powerful feature, you can run several network applications on your computer at the same time.

The *Windows Network Manager* is a graphical, “point-and-click” Windows program for managing the network. The main screen of the network manager is shown in figure 1-2. The functions appearing on the main screen are:

Poll — Lists all the stations in the network, and lets you select a station to manage.

Drives — Displays the current drive mappings. Also lets you discontinue drive mappings, and create new drive mappings.

Printers — Displays the current printer mappings. Also lets you discontinue printer mappings, and create new printer mappings.

Preferences — Adjusts the way Invisible LAN works on your own computer.

Logon — Lets you enter your user name and password.

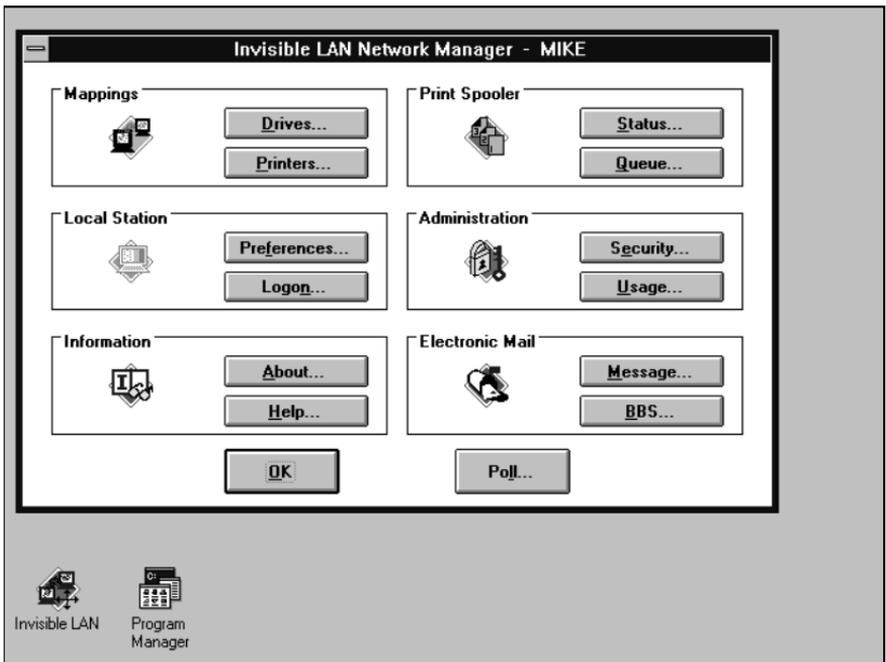


Figure 1-2. Windows Network Manager

About — Displays the Invisible LAN version number and copyright message.

Help — Starts the help system and displays help information. The Windows Network Manager includes extensive context-sensitive help, plus a complete on-line reference manual.

Status — Shows the status of the server's network printers. Lets you pause printing and configure the printers.

Queue — Lists the print jobs waiting to be printed on the server's network printers. Lets you rearrange the print queue.

Security — Displays and changes security information. Lets you define users and groups, and specify who has access to network resources.

Usage — Monitors the usage of server resources, such as open files and record locks.

Message — Sends a message to another network user.

BBS — Enters the network bulletin board system. You can leave messages for other users, and retrieve messages that others have left for you.

SUMMARY: The *Windows network driver* extends Windows to operate on the network, and gives all Windows programs access to network resources. The *Windows Network Manager* is a graphical Windows program that gives you full control over the network from within Windows.

Bulletin Board

The bulletin board is a store-and-forward mail system. *Store-and-forward* means that when you send a message, it is stored on a file server. When the recipient logs in to the bulletin board, the message is then forwarded to the recipient.

When you send a message, it does not “pop up” on the recipient’s screen. In fact, the recipient does not have to be on the network at all. Messages you send simply remain on the server’s disk until the recipient logs in and reads them.

You can send a message to either an individual user or a group. A *group* functions like a mailing list. When you send a message to a group, the message is automatically forwarded to every member of the group.

A message can be either private or public. A *private* message can be viewed only by its intended recipient. (A private message sent to a group can be viewed only by members of the group.) A *public* message can be viewed by anyone on the bulletin board. With public messages, the bulletin board acts like an “electronic forum” that allows users to exchange information and learn from each other’s experiences.

When you receive a message, you have the option of sending a reply. All the replies to a message are automatically grouped together into a *reply chain*. This allows you to quickly locate all the replies to a given message. In fact, reply chains make it possible for two or more users to carry on a “conversation” about a topic; each reply chain is a different “conversation.”

SUMMARY: The bulletin board is a store-and-forward mail system which also functions as an electronic forum. You can send messages to individuals or groups, and messages can be public or private. The bulletin board keeps track of replies, organizing all reply messages into reply chains.

NetBIOS and TransBIOS

Invisible LAN includes two complete network protocols: NetBIOS and TransBIOS. A *network protocol* is a method that programs use to communicate on the network. Users do not have to deal with the protocols.

NetBIOS is a standard network protocol defined by IBM. There are many network application programs that use the NetBIOS protocol. Since Invisible LAN includes NetBIOS, you can use any application program that is designed for NetBIOS networks.

TransBIOS is a special high-performance protocol defined by Invisible Software. All of the Invisible LAN programs use TransBIOS for network communication. This allows Invisible LAN to operate at very high speed.

By including two complete network protocols, Invisible LAN gives you the best of both worlds: the compatibility of NetBIOS, and the high performance of TransBIOS.

SUMMARY: *NetBIOS* and *TransBIOS* are the two network protocols included with Invisible LAN.

Ultra Server

Ultra Server is a high-performance version of the Invisible LAN file server. It is intended for dedicated server applications, where the highest possible performance is required. (A *dedicated server* is a computer that is used exclusively as a file server, with no local user.) However, you can also run Ultra Server in non-dedicated mode if you wish.

Ultra Server speeds up all disk operations, and it is particularly effective at speeding up CD-ROMs and other optical disk drives. In addition, Ultra Server can support large numbers of network files and record locks, so it is suitable for large networks with many users.

Ultra Server requires a 386 or 486 computer, and it is designed to take full advantage of the capabilities of the 386 and 486. The key features of Ultra Server are:

- Written in 32-bit code.
- Runs in protected mode on a 386 or 486.
- Caches all drives in the system, including CD-ROMs, SCSI drives, bus

master drives, etc.

- Supports up to 4095 megabytes of cache memory.
- Supports up to 32768 open files.
- Supports up to 65535 recored locks.
- Uses true multitasking so multiple users can access data simultaneously.
- Uses DOS for disk accesses, so it is compatible with all DOS device drivers.

When you install Ultra Server, it runs in parallel with the standard Invisible LAN file server. Ultra Server handles all file operations on cached drives. The standard server handles file operations on uncached drives, as well as printing, security, and management functions. As a result, switching from the standard server to Ultra Server, or vice-versa, is completely transparent to users. Ultra Server appears to users to be exactly the same as the standard file server, just faster.

SUMMARY: *Ultra Server* is a high-performance version of the Invisible LAN file server, that is used mainly for dedicated server applications.

Remote Login and Remote Bridging

Invisible LAN Remote is an add-on to Invisible LAN that lets you extend your network by using modems. Invisible LAN Remote provides two functions: remote login and remote bridge.

Remote login lets you connect a computer to the network through a modem. Once connected, the remote computer becomes a network workstation, and has access to all network resources.

To perform remote login, one computer on the network runs *remote login host* software. The remote computer runs *remote login client* software. Then modems are used to establish a connection between the host and the client. Once connected, the host computer is responsible for transferring data between the network and the remote computer.

Remote bridging lets you connect two networks together through modems. Once connected, any user on either network can access resources on the other network.

To set up a remote bridge, one computer on each network runs the remote login host software. Then modems are used to establish a connection between the two hosts. Once connected, each host is responsible for transferring data

between its network and the other host.

Invisible LAN Remote has its own built-in security system, which lets you control who can log in to the network remotely, and what resources can be accessed through a remote bridge. Invisible LAN Remote's security system operates independently of the Invisible LAN security system, and provides an extra layer of security protection.

Invisible LAN Remote is not described in this manual. Refer to the *Invisible LAN Remote Instruction Manual* for an explanation of how to install and use Invisible LAN Remote.

SUMMARY: *Remote login* lets you connect a computer to the network through a modem connection. Once connected, the remote computer functions as a network station. *Remote bridging* lets you connect two networks together through a modem connection. Once connected, any user on either network can access resources on the other network. Both functions are provided by Invisible LAN Remote.

Network Configuration

Installing Invisible LAN requires the following operations:

- Copy the software onto your hard disk.
- Configure the software.
- Modify CONFIG.SYS.

All these operations can be performed conveniently using the Invisible LAN Setup program. This chapter describes the procedure.

This chapter focuses on installing the network in a basic configuration, using the Setup program's **Easy Configuration** option. The Setup program also has an **Advanced Configuration** option which provides access to many advanced features of the network. The advanced features are described in later chapters of this manual.

Starting the Setup Program

To start the Setup program, insert the Invisible LAN program diskette #1 into drive A:, and then type

```
A:SETUP30
```

If your computer has a monochrome VGA display (like many laptop computers do), the display may be easier to read if you use the command

```
A:SETUP30 /B
```

In either case, the Setup program Main Menu appears, as shown in figure 2-1. The Setup program is controlled by a sequence of menus, so it is very easy to use. The keys that control the Setup program are shown in a bar at the bottom of the screen. You can get help at any time by pressing **F1**.

SUMMARY: Use the **SETUP30** command to start the Invisible LAN Setup program.

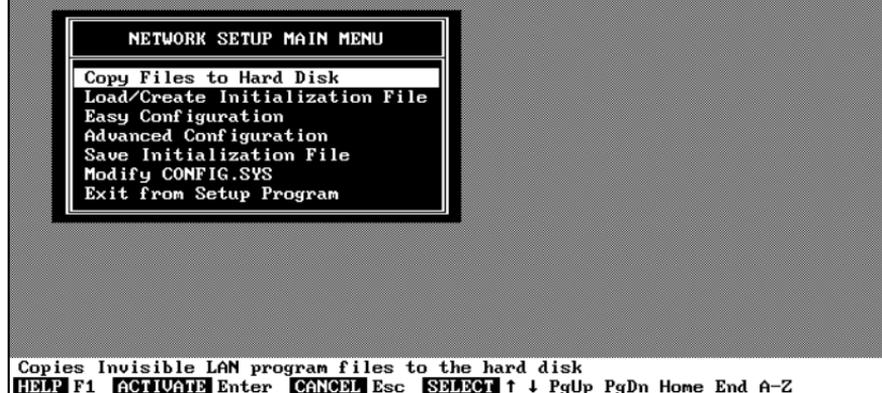


Figure 2-1. Setup Program Main Menu

Copying the Software onto the Hard Disk

The next step is to copy the software from the Invisible LAN program diskettes to your hard disk. You need to create a new directory on your hard disk, and then copy all the Invisible LAN program files into the new directory.

In addition, you need to copy the DOS program SHARE.EXE into the new directory. SHARE.EXE is supplied on your original DOS diskettes; it is not supplied as a part of Invisible LAN. Make sure you have a copy of SHARE.EXE handy, either on a diskette or on the hard disk.

Note that the files on the Invisible LAN program diskettes are stored in a compressed form. You cannot copy the files to the hard disk using the DOS COPY command. You must use **SETUP30** (or **QSETUP**) to copy the files.

Step 1. Select **Copy Files to Hard Disk** from the Setup program Main Menu. Use the up/down arrow keys to highlight **Copy Files to Hard Disk**, and then press **Enter**.

Step 2. A panel appears showing the source and destination directories for the copy, as shown in figure 2-2. By default, the source directory is A:\, and the destination directory is C:\NET30. If these are not the directories you want, you can type over them. (For example, if you prefer to install the software on drive D:, you could change the destination directory to D:\NET30.) When the directory names are correct, press **Enter** or **F9** to continue.

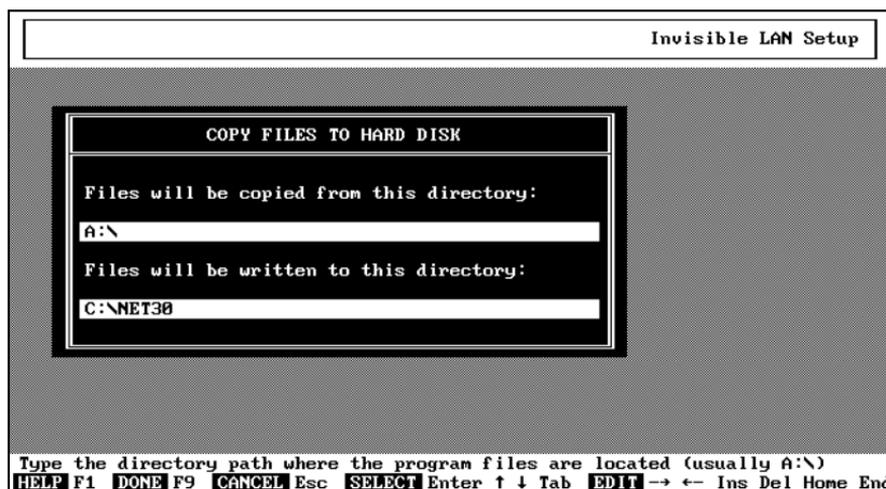


Figure 2-2. Copying Files to the Hard Disk

Step 3. The Setup program automatically copies all the files. The Setup program will prompt you to change diskettes when necessary.

Step 4. After copying the program files, the Setup program asks you for the location of SHARE.EXE. You can insert a diskette with a copy of SHARE.EXE, or you can enter the name of a directory on the hard disk that contains SHARE.EXE. When ready, press **Enter** or **F9** to continue.

Step 5. The Setup program copies SHARE.EXE into the Invisible LAN directory, and then returns to the Main Menu. The file copy is now complete.

Note — If you have a computer without a hard disk (that is, a computer with only a floppy diskette), refer to the text file FLOPPY.TXT for instructions on copying the Invisible LAN program files to the floppy diskette.

SUMMARY: Use **Copy Files to Hard Disk** on the Setup program Main Menu to copy all the Invisible LAN program files to your hard disk.

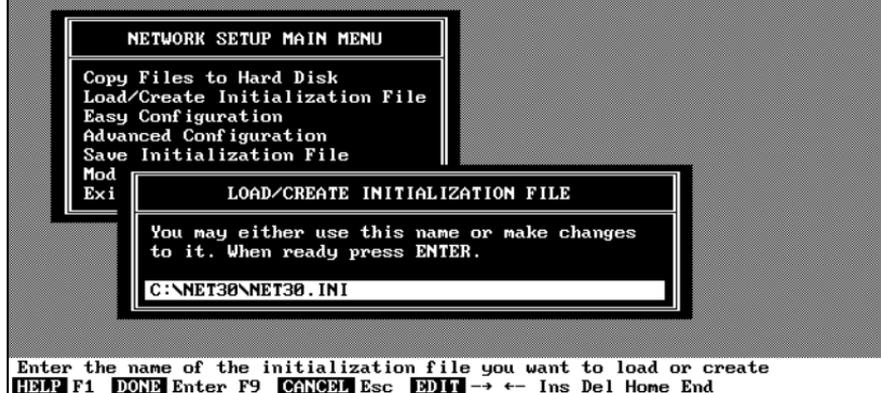


Figure 2-3. Loading or Creating an Initialization File

Configuring the Software

The configuration of the Invisible LAN software is stored in an *initialization file*. Some of the information stored in the initialization file is: your name; the type of network hardware you have; and the network programs you want to use (server, redirector, mail).

Step 1. Select **Load/Create Initialization File** from the Setup program Main Menu. The Setup program displays the name of the initialization file, as in figure 2-3. The default name is NET30.INI. For this basic installation, accept the default name by pressing **Enter** or **F9**.

Step 2. Select **Easy Configuration** from the Main Menu. The Easy Configuration panel appears, as shown in figure 2-4.

Step 3. In the **User Name** field, enter your name. This will be your name on the network. Each network user must have a different name.

Step 4. In the **Software Version** field, select which version of the Invisible LAN software you want to use. Press **F5** and **F6** to select the desired option. The available versions depend on which software you have purchased; press **F1** for a complete list.

STANDARD selects the standard version of Invisible LAN, which can be used on all computers running DOS or Windows.

ULTRA_SERVER selects the high-performance file server software, which is used mainly to set up a dedicated server. This version requires a 386 or 486 computer with extended memory. In addition, if you select this version, then you cannot use any 386 memory manager (like EMM386 or QEMM-386). Also, if possible, you should not use HIMEM.SYS.

REMOTE_LOGIN selects the remot login client software. This option is described in the *Invisible LAN Remote Instruction Manual*.

Step 5. In the **Network Hardware Type** field, select the type of network hardware you are using. Use **F5** and **F6** to select the correct option.

If you need help in making the correct selection, press **F1** for an explanation of the different hardware types. If you need additional help, refer to the text file HARDWARE.TXT.

Step 6. Enter **Y** in the **Install Electronic Mail** field if you want to install the electronic mail software; enter **N** if you don't want to install the electronic mail.

If you enter **Y**, then you can send messages to, and receive messages from, other network users. Each user that sends and receives messages must install the electronic mail software.

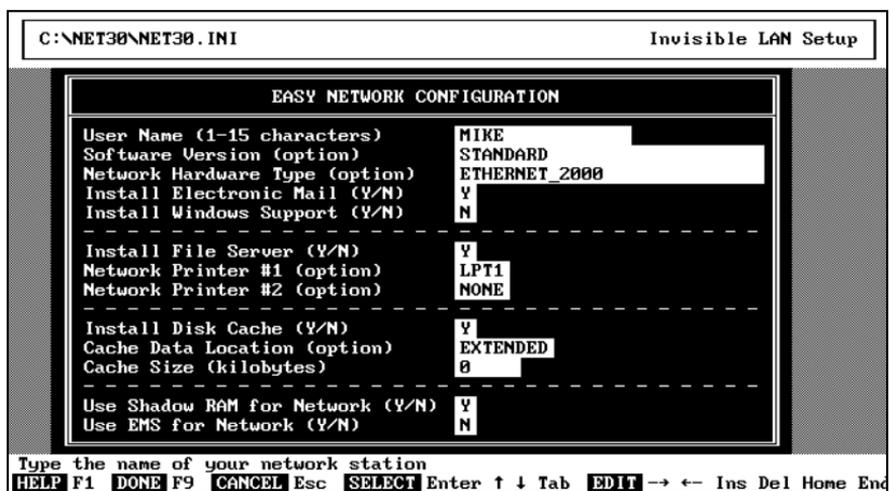


Figure 2-4. Easy Configuration Screen

Step 7. Enter **Y** in the **Install Windows Support** field if you want to run Microsoft Windows; enter **N** if you don't want to run Microsoft Windows.

Important Note — If you want to run Microsoft Windows on the network, then you must install the Invisible LAN Windows Driver. Refer to chapter 5 for instructions on installing the Windows Driver.

Step 8. Enter **Y** in the **Install File Server** field if you want to want to install the file server software; enter **N** if you don't want to install the file server.

If you enter **Y**, then other users will have access to your disks and, optionally, your printers. If you enter **N**, then other users will not have access to your disks and printers.

Step 9. If you installed the file server, you can optionally make your printers available to other network users.

- If you don't want to share your printers, choose **NONE** in the **Network Printer #1** field (use **F5** and **F6** to make your choice). The **Network Printer #2** field will be blank.
- If you want to share one of your printers, choose the printer name in the **Network Printer #1** field; using **F5** and **F6**, you can choose **LPT1**, **LPT2**, **LPT3**, **COM1**, or **COM2**. Choose **NONE** in the **Network Printer #2** field.
- If you want to share two printers, choose the name of the first printer in the **Network Printer #1** field, and the name of the second printer in the **Network Printer #2** field.

Step 10. Enter **Y** in the **Install Disk Cache** field if you want to install the disk cache software; enter **N** if you don't want to install the disk cache.

If you enter **Y**, the disk cache software will speed up your computer's hard disk by storing disk data in memory.

Step 11. In the **Cache Data Location** field, use **F5** and **F6** to choose the type of memory you want to use for the cache. You have three options:

- **EMS** — Uses expanded memory. This type of memory requires a special *expanded memory manager* program, which must conform to the Expanded Memory Specification 4.0 (also called EMS 4.0 or LIM 4.0).
- **EXTENDED** — Uses the extended memory present in many 286, 386, and 486 computers.

- **XMS** — Uses extended memory, except that the cache accesses the extended memory through an *XMS device driver* such as Microsoft's HIMEM.SYS.

If you have installed HIMEM.SYS and you want to use extended memory, you should choose **XMS**. If you have not installed HIMEM.SYS and you want to use extended memory, you should use **EXTENDED**. Also, note that if you want to run Microsoft Windows, then you cannot choose **EXTENDED**; you must use **XMS** instead.

Step 12. In the **Cache Size** field, enter the amount of expanded or extended memory that you want to use for the cache. If you leave the field blank, or enter zero, then the cache uses all available memory.

If you are using Ultra Server, this value divides extended memory between the cache and Ultra Server. As a starting point, we recommend using one-fourth of extended memory for the cache, and three-fourths for Ultra Server.

Step 13. In the **Use Shadow RAM for Network** field, enter **Y** if you want to load the network software into shadow RAM (also called *upper memory* or *above 640K memory*). Enter **N** if you don't want to load the network software into shadow RAM.

If you enter **Y** in this field, you must install a memory manager program to create shadow RAM. You can use any of the following memory managers: Invisible RAM, or Quarterdeck's QEMM-386, or Microsoft's EMM386, or Qualitas' 386MAX. (Invisible LAN can use any memory manager that supports XMS UMB function calls, or DOS 5 UMB function calls.)

Step 14. In the **Use EMS for Network** field, enter **Y** if you want to load the network software into expanded memory. Enter **N** if you don't want to load the network software into expanded memory.

If you enter **Y** in this field, you must have expanded memory in your computer, compatible with EMS version 4.0.

Note — You can specify **Y** for both **Use EMS for Network Software** and **Use Shadow RAM for Network Software**. The following table shows the results for each possible option.

Table 2-1. Network Software Memory Usage Options

<i>Use EMS?</i>	<i>Use Shadow?</i>	<i>Result</i>
N	N	The entire network software is loaded into conventional DOS memory
N	Y	The entire network software is loaded into shadow RAM
Y	N	Most of the network software is loaded into EMS, and the remainder is loaded into conventional DOS memory
Y	Y	Most of the network software is loaded into EMS, and the remainder is loaded into shadow RAM

Step 15. When your configuration is correct, press **Enter** or **F9** to complete the data entry and return to the Main Menu.

If you want to change any of your entries, use the up/down arrow keys to move the cursor to the entry you want to change, and then make your changes.

Step 16. Select **Save Initialization File** from the Main Menu. The Setup program displays the name of the initialization file, as in figure 2-5. The default name is NET30.INI. For this basic installation, accept the default name by pressing **Enter** or **F9**.

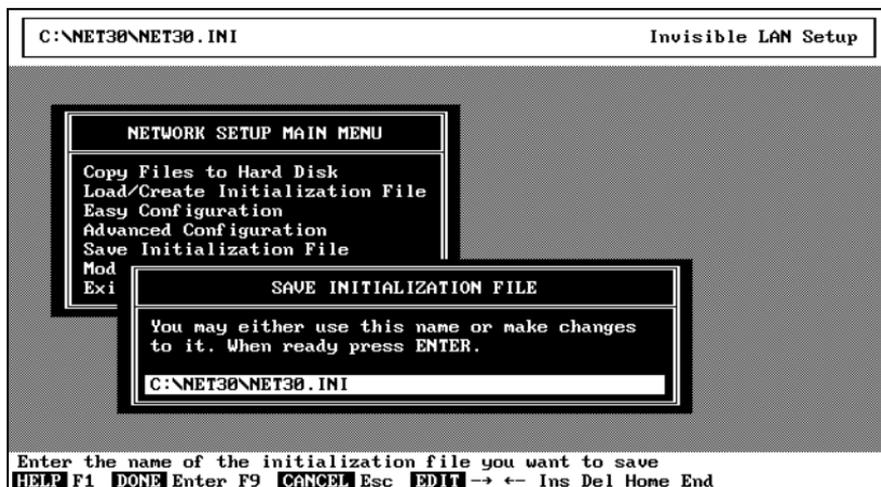


Figure 2-5. Saving the Initialization File

The network software configuration is now complete.

SUMMARY: Use **Load/Create Initialization File** to create a new initialization file or load an existing file into memory. Then use **Easy Configuration** to enter your desired configuration for the network software. On the Easy Configuration panel, you specify your user name, the network hardware type, file server and print spooler options, electronic mail option, Windows support option, disk cache configuration, and desired memory usage. Finally, use **Save Initialization File** to save your selected configuration to the disk.

Modifying CONFIG.SYS

You need to modify your CONFIG.SYS file to prepare for network use. You can either use the Setup program to modify the CONFIG.SYS file, or you can use a text editor to modify the CONFIG.SYS file yourself.

We recommend that the following lines be placed into the CONFIG.SYS file:

```
BUFFERS = 20
FILES = 100
FCBS = 16,8
LASTDRIVE = Z
STACKS = 32,128
DEVICE = C:\NET30\CACHE30.SYS
DEVICE = C:\NET30\N30DEV.SYS
```

Exception — If you are using Ultra Server, the recommended value of **STACKS** is **32,256**.

Important—**CACHE30.SYS** must be the first device driver installed. **N30DEV.SYS** must be the last device driver installed. For example, if you want to install the ANSI.SYS device driver, your CONFIG.SYS file might look like this:

```
BUFFERS = 20
FILES = 100
FCBS = 16,8
LASTDRIVE = Z
STACKS = 32,128
DEVICE = C:\NET30\CACHE30.SYS
DEVICE = ANSI.SYS
DEVICE = C:\NET30\N30DEV.SYS
```

If you want to use the Setup program to change your CONFIG.SYS file, proceed as follows:

Step 1. Select **Modify CONFIG.SYS** from the Main Menu.

Step 2. The Setup program asks you which drive you boot from. Enter the drive letter of the disk you use to start the system and load DOS; in most cases, this is drive **C**. Press **Enter** to continue.

Step 3. The Setup program asks you the name of the directory where the Invisible LAN program files are installed; in most cases, this is C:\NET30. If the default value supplied by the Setup program is not correct, enter the correct value. Press **Enter** to continue.



Figure 2-6. Modifying the CONFIG.SYS File

Step 4. The Setup program displays a panel showing the proposed changes to CONFIG.SYS, as in figure 2-6. On this panel you can change the values of BUFFERS, FILES, LASTDRIVE, and STACKS, if you are not satisfied with the recommended values. When the values are acceptable, press **Enter** or **F9** to complete the operation.

If you need assistance in understanding the CONFIG.SYS commands, you can refer to chapter 7, or else refer to your DOS Reference Manual.

SUMMARY: You need to prepare your CONFIG.SYS file for network use by installing the CACHE30.SYS and N30DEV.SYS device drivers, and by setting appropriate values for BUFFERS, FILES, LASTDRIVE, and STACKS.

Exiting from Setup

To exit from the Setup program, return to the Main Menu. Then select **Exit from Setup Program**, or press **Esc**.

You may want to change your DOS PATH to include your new Invisible LAN directory, C:\NET30. You can do this by editing your AUTOEXEC.BAT file.

If you change your CONFIG.SYS or AUTOEXEC.BAT file, you need to reboot your system in order for the new CONFIG.SYS or AUTOEXEC.BAT to take effect. Press **Ctrl-Alt-Del** to reboot the system.

SUMMARY: Use **Exit from Setup Program** to leave the Setup program and return to DOS. Be sure to reboot your system if you change CONFIG.SYS or AUTOEXEC.BAT.

Starting the Network

Once you have installed and configured the software, you are ready to start the network. This chapter describes:

- Starting Invisible LAN
- Using network disks and printers
- Getting help with Invisible LAN commands

Starting Invisible LAN

To start Invisible LAN, type

NET30

The **NET30** program reads your initialization file and loads the appropriate programs.

■ **SUMMARY:** The **NET30** command starts the Invisible LAN software.

Using Network Disks and Printers

This section gives you a quick way to establish a link between your computer and a network disk or printer using the **MAP** command. The *DOS User Manual* discusses the **MAP** command in more detail. The disk or printer you want to use must be attached to a server.

Network Disks

To use a disk attached to a server, type

```
MAP d: \\server\d:\
```

Replace the first **d:** with the drive letter to be used to refer to the server's disk. Replace **server** with the network name of the server. Replace the second **d:** with the server's drive letter.

Example 1 — A server named JIM has a fixed disk named C:. To use drive letter D: to refer to JIM's fixed disk, type

```
MAP D: \\JIM\C:\
```

Example 2 — A server named TOM has a floppy disk named B:. To use drive letter E: to refer to TOM's floppy disk, type

```
MAP E: \\TOM\B:\
```

Network Printers

To use a printer attached to a server, type

```
MAP LPTx \\server\n
```

Replace **LPTx** with the name to be used for the printer: LPT1, LPT2, or LPT3. Replace **server** with the network name of the server. Replace **n** with the number of the printer: 1 or 2. Recall that the printer numbers were set when you configured the Invisible LAN server.

Example — Suppose that server MIKE was configured to use COM1 for network printer #1, and LPT1 as network printer #2.

To use LPT2 on your computer to refer to MIKE's network printer 1 and LPT3 to refer to MIKE's network printer 2, type

```
MAP LPT2 \\MIKE1  
MAP LPT3 \\MIKE2
```

Files you send to your printer LPT2 are now printed on MIKE's printer COM1 and files you send to your printer LPT3 are printed on MIKE's printer LPT1.

Discontinuing a Network Disk

After creating a drive mapping, you can use the **MAP** command to terminate the mapping.

To discontinue a drive mapping, type

```
MAP d: /D
```

Example—Suppose you are using drive letter F: to refer to a server's disk. You can terminate the mapping by typing

```
MAP F: /D
```

Now your drive letter F: no longer refers to the server's disk.

Discontinuing a Network Printer

After creating a printer mapping, you can use the **MAP** command to terminate the mapping.

To discontinue a printer mapping, type

```
MAP LPTx /D
```

Example—Suppose you are using print device LPT2 to refer to a server's printer. You can terminate the mapping by typing

```
MAP LPT2 /D
```

Now your print device LPT2 no longer refers to the server's printer.

Displaying Mappings

You can display all your drive and printer mappings by using the **MAP** command with no parameters. Simply type

```
MAP
```

Note—If you install the server and you share one or two printers, the Setup program automatically establishes printer mappings for your own printer(s). This makes your printer output go through the print spooler. Whenever you share a printer, it is important that you establish a mapping to your own printer, so that your printer output does not conflict with other users' printer output.

SUMMARY: The **MAP** command lets you access disks and printers that are attached to servers.

Getting Help With Invisible LAN Commands

All Invisible LAN commands have on-line help information. To display the help information, enter the name of the command followed by a question mark. For example, to get help using the **MAP** command, type

```
MAP ?
```

You can also use a slash followed by a question mark, for example

```
MAP /?
```

SUMMARY: To get help using a command, enter the name of the command followed by a question mark.

Changing the Configuration

This chapter describes:

- Changing the Network Configuration
- Using Multiple Configurations

Changing the Network Configuration

From time to time, you may want to change your Invisible LAN configuration. Here is how to do it:

Step 1. Start the Invisible LAN Setup program by entering the command

SETUP30

Step 2. Select **Load/Create Initialization File** from the Main Menu, and read your initialization file into memory.

Step 3. Select **Easy Configuration** (or **Advanced Configuration**) from the Main Menu, and enter your new configuration.

Step 4. Select **Save Initialization File** from the Main Menu, and save your initialization file back to disk.

The next time you start Invisible LAN, your new configuration will take effect.

SUMMARY: Use the Setup program to change the Invisible LAN configuration. You read the initialization file into memory, make the desired changes, and then write the initialization file back to disk.

Multiple Configurations

The Invisible LAN configuration is stored in an initialization file. You might find it convenient to have more than one initialization file on your disk.

For example, it may be that sometimes you want to install the file server, and other times you don't. In this case, you would have two initialization files: one file that you use when you install the server, and another file that you use when you don't install the server.

Invisible LAN allows you to have more than one initialization file. The only restrictions are that an initialization file must be located in the same directory as the Invisible LAN program files, and the name of an initialization file must end in .INI. For example, initialization files could have names like MIKE.INI, VINCE.INI, or REGINA.INI.

Using Multiple Configuration Files

When you start Invisible LAN, you can specify the name of an initialization file on the DOS command line. For example,

```
NET30 MIKE
```

would start Invisible LAN using the initialization file MIKE.INI. Notice that you don't type the ".INI" on the command line.

Creating Multiple Configuration Files

Initialization files are created using the Invisible LAN Setup program. The Setup program automatically supplies the default initialization file name, NET30.INI. However, you can tell the Setup program to use a different file name.

There are two ways you can create a new initialization file: you can create it from scratch, beginning with the default parameter values; or you can modify an existing initialization file.

To create a new initialization file from scratch:

Step 1. Select **Load/Create Initialization File** from the Setup program Main Menu.

Step 2. Enter the name of your new initialization file. The Setup program creates the file and initializes all parameters to their default values.

Step 3. Select **Easy Configuration** (or **Advanced Configuration**) from the Main Menu, and enter your configuration.

Step 4. Select **Save Initialization File** from the Main Menu, and save your new initialization file back to disk.

To create a new initialization file by modifying an existing initialization file:

Step 1. Select **Load/Create Initialization File** from the Setup program Main Menu.

Step 2. Enter the name of your existing initialization file. The Setup program reads the file into memory.

Step 3. Select **Easy Configuration** (or **Advanced Configuration**) from the Main Menu, and make the desired changes to the configuration.

Step 4. Select **Save Initialization File** from the Main Menu, and enter the name of your new initialization file. The Setup program creates the file, and the information in memory is written to disk in the new initialization file.

SUMMARY: Invisible LAN lets you have multiple initialization files, so you can switch between configurations easily. You can specify which initialization file to use on the **NET30** command line. You create your initialization files with the Setup program, either by starting from scratch or by modifying an existing initialization file.

Windows Driver Installation

If you want to run Microsoft Windows on the network, you must install the Invisible LAN Windows driver. This chapter will help you get the Invisible LAN Windows driver up and running as quickly as possible. The topics covered are:

- Installation overview
- Installing Windows
- Installing Invisible LAN
- Installing the Invisible LAN Windows driver
- Installing the Invisible LAN icons
- Starting the software

This chapter only gives basic information for installing the Invisible LAN Windows driver. Chapter 10 provides more detailed information, and describes additional installation options. **Important** — If you are using Windows For Workgroups, see README.TXT or WORKGRP.TXT for special instructions.

Installation Overview

Windows and Invisible LAN are both large programs with many functions. Both of them need to be configured during installation.

The easiest way to begin is by installing Windows and Invisible LAN separately. Installing the programs separately makes it much easier to diagnose and fix any problems that may arise. When both programs are installed and running, then you can link them together to complete your Invisible LAN Windows installation.

In the following sections, we describe how to install Windows, and how to install Invisible LAN. Then we describe how to link them together by installing the Invisible LAN Windows driver.

SUMMARY: The quickest way to install Invisible LAN for Windows is to first install Windows, then install Invisible LAN, and then link them together by installing the Windows network driver.

Installing Windows

Begin by installing Windows. This should be done without the network running.

To install Windows, you need to copy the Windows program files onto your hard disk, configure Windows for your computer, and modify your CONFIG.SYS file. All these tasks are performed with the Windows Setup program.

Follow the instructions in your Microsoft Windows User's Guide for running the Windows Setup program.

The Windows Setup program may ask you what type of network you are using. At this time, you should select "No Network." We will configure Windows for network operation later, when we install the Invisible LAN Windows driver.

SUMMARY: Install Windows in the usual way, in a "No Network" configuration. We will add the network support later.

Installing Invisible LAN

To install Invisible LAN, you need to copy the Invisible LAN program files onto your hard disk, create the Invisible LAN initialization file, and modify your CONFIG.SYS file. All these tasks are performed with the Invisible LAN Setup program (**SETUP30**).

Chapters 2 and 8 describe how to use the Invisible LAN Setup program to configure the network software. If you are going to run Windows, you must observe the following special requirements:

- You must enable the Windows Support option when configuring Invisible LAN. To do this, enter **Y** for **Windows Support** in the Invisible LAN Setup program.
- Do not install any of the Invisible RAM memory managers.

- If you install the disk cache, the Cache Data Location must be either **XMS** or **EMS**.

SUMMARY: Use **SETUP30** to perform the Invisible LAN installation. Enter **Y** for the **Windows Support** option in the Invisible LAN Setup program.

Installing the Invisible LAN Windows Driver

After installing both Windows and Invisible LAN, you are ready to install the Invisible LAN Windows driver. The *network driver* is the program that links Windows and Invisible LAN together. With the network driver installed, you have the full functionality of Invisible LAN for Windows.

The following procedure installs the Windows network driver. In addition, it also creates an Invisible LAN group in Program Manager, and installs several icons in the group. The icons provide quick access to network functions from within Windows.

The following instructions assume that your DOS PATH includes the directory where Invisible LAN is installed (usually C:\NET30).

- Step 1. Select **Run** from the Program Manager's **File** menu.
- Step 2. Type "QSETUPW" and choose **OK**.
- Step 3. A dialog box appears, asking if you want to install the Windows network driver for Invisible LAN. Choose **OK** to install the driver.
- Step 4. Wait while QSETUPW installs the driver and creates the Invisible LAN group. This will take approximately 20 seconds.
- Step 5. When the installation is complete, QSETUPW displays a dialog box informing you that you have to restart Windows for the changes to take effect. Choose **Yes** to restart Windows.

SUMMARY: Use the QSETUPW program to install the Invisible LAN Windows driver. This also creates an Invisible LAN group in Program Manager.

Starting the Software

Always start the network before starting Windows. A typical start-up sequence looks like this:

```
NET30  
WIN
```

The **NET30** command starts the Invisible LAN software, and the **WIN** command starts Windows.

■ **SUMMARY:** Always start Invisible LAN before starting Windows.

Memory Usage

One of the outstanding features of Invisible LAN is its ability to optimize the use of your computer's memory. This chapter describes how to use Invisible LAN when running a memory manager. The topics covered are:

- Using shadow RAM
- Using expanded memory
- HIMEM.SYS

Using Shadow RAM

Shadow RAM is memory that is located above 640K, and below 1M. You can use shadow RAM to hold memory-resident programs, such as the Invisible LAN network software. This frees up more conventional DOS memory for your DOS applications. (Shadow RAM is also known as *upper memory* and *high DOS memory*.)

In order to use shadow RAM, you need to install *memory manager* software. You can use one of the Invisible RAM memory managers; refer to chapters 11-13. Alternatively, you can use another memory manager such as: Microsoft's EMM386, Quarterdeck's QEMM-386, or Qualitas' 386-MAX.

Loading Invisible LAN Into Shadow RAM

Most memory managers include a "loadhigh" utility that is used to load memory-resident programs into shadow RAM. **You should never attempt to use a "loadhigh" utility to load Invisible LAN into shadow RAM.**

You use the Invisible LAN Setup program to load the network software into shadow RAM. When you configure the network software, the Setup program asks if you want to use shadow RAM. Simply enter **Y**. Then, when you start Invisible LAN it automatically loads itself into shadow RAM.

Using the Setup program is more convenient than using a "loadhigh" utility. And more importantly, this procedure yields much more efficient use of shadow RAM, because Invisible LAN can allocate exactly the amount of shadow RAM needed for the memory-resident portion of the software.

Memory Manager Requirements

In order for Invisible LAN to load itself into shadow RAM, it has to communicate with the memory manager and allocate the memory it needs. For this to work, Invisible LAN and the memory manager must speak the same “language.”

Fortunately, Invisible LAN knows how to speak four different “languages” when communicating with the memory manager:

- *The Microsoft XMS UMB Specification.* This is a standard developed by Microsoft for allocating shadow RAM (the UMB stands for Upper Memory Block).
- *The MS-DOS 5.0 UMB Specification.* This is another standard developed by Microsoft, that is provided if you run DOS 5.0 and use the command `DOS=UMB` in your `CONFIG.SYS` file.
- *The Invisible RAM Specification.* This standard for allocating shadow RAM was developed by Invisible Software, and is used in all the Invisible RAM memory managers.
- *The QEMM-386 Specification.* This standard was developed by Quarterdeck, and is used in almost all versions of QEMM-386.

In order for Invisible LAN to utilize shadow RAM, you must have a memory manager that implements at least one of the above four specifications. Luckily, almost every memory manager on the market today meets this requirement.

Loading SHARE Into Shadow RAM

SHARE is a program provided as part of DOS. It is used on the file server to help manage file sharing. Because SHARE is closely associated with the network, the Invisible LAN program loader has been designed to automatically load SHARE for you.

In most cases, Invisible LAN can load SHARE into shadow RAM along with the rest of the network software. However, there are certain unusual system configurations where Invisible LAN can't load SHARE into shadow RAM, even though it *can* load the rest of the network software into shadow RAM.

If you find that Invisible LAN isn't loading SHARE into shadow RAM, you can use your memory manager's “loadhigh” utility to load SHARE into shadow RAM. You should do this before starting Invisible LAN. For example:

```
LOADHIGH SHARE.EXE /F:2048 /L:20  
NET30
```

In this example, `LOADHIGH` is the memory manager's utility. The `/F` and `/L` are parameters to the `SHARE` program, that specify the filespace size and the

number of record locks.

Shadow RAM and Windows

You can use shadow RAM when running Microsoft Windows, provided that your memory manager is Windows-compatible.

You should be aware that if you run Windows in 386 enhanced mode, it requires some shadow RAM for its own use. Invisible LAN uses shadow RAM so efficiently that it can fill up all the shadow RAM space, leaving nothing for Windows.

If you find that Windows won't run when the network software is loaded into shadow RAM, it may be that all the shadow RAM space is filled up. If this happens, you should remove some of the network modules from shadow RAM. (The advanced configuration section of the Setup program lets you control shadow RAM usage on a module-by-module basis.)

SUMMARY: *Shadow RAM* (also called *upper memory* or *high DOS memory*) is memory that is located between 640K and 1M. To use shadow RAM, you need to install a memory manager. With the memory manager installed, you can use the Invisible LAN Setup program to configure the network software to use shadow RAM. Then, when you start Invisible LAN it automatically copies itself into shadow RAM. Never try to use a "loadhigh" utility to load Invisible LAN into shadow RAM.

Using Expanded Memory (EMS)

Expanded memory is a way of making many megabytes of additional memory that is accessible to DOS program. (Expanded memory is also known as *EMS memory* and *LIM memory*.)

In order to use expanded memory, you need to install *expanded memory manager* software. Most of the Invisible RAM programs can function as expanded memory managers. Alternatively, you can use another memory manager such as: Microsoft's EMM386, Quarterdeck's QEMM-386, or Qualitas' 386-MAX.

Loading Invisible LAN Into Expanded Memory

Most of the network software can be loaded into expanded memory. You do this with the Invisible LAN Setup program. When you configure the network

software, the Setup program asks if you want to use expanded memory. Simply enter **Y**. Then, when you start Invisible LAN it automatically loads most of itself into expanded memory.

Note — Loading the network software into expanded memory makes the network run significantly slower.

Memory Manager Requirements

In order for Invisible LAN to load itself into expanded memory, it has to communicate with the expanded memory manager and allocate the memory it needs.

Invisible LAN requires that the expanded memory manager be compatible with the Expanded Memory Specification (EMS) version 4.0. Since virtually every expanded memory manager conforms to EMS 4.0, you should be able to use any expanded memory manager that you happen to have.

Using Expanded Memory With Shadow RAM

It is possible to use both expanded memory and shadow RAM at the same time. Table 6-1 shows the result of each possible selection.

<i>Use EMS?</i>	<i>Use Shadow?</i>	<i>Result</i>
N	N	The entire network software is loaded into conventional DOS memory
N	Y	The entire network software is loaded into shadow RAM
Y	N	Most of the network software is loaded into EMS, and the remainder is loaded into conventional DOS memory
Y	Y	Most of the network software is loaded into EMS, and the remainder is loaded into shadow RAM

SUMMARY: *Expanded memory* (also called *EMS memory* or *LIM memory*) is a way of adding several megabytes of memory to a DOS system. To use expanded memory, you need to install an expanded memory manager, compatible with EMS 4.0. With the expanded memory manager installed, you can use the Invisible LAN Setup program to configure the network software to use expanded memory. Then, when you start Invisible LAN it automatically copies most of itself into expanded memory, leaving the remainder in conventional DOS memory or shadow RAM.

HIMEM.SYS

HIMEM.SYS is a device driver that can be used with 286, 386, and 486 systems. HIMEM.SYS controls access to the computer's extended memory.

Loading DOS High

With DOS 5.0, if you install HIMEM.SYS you can load most of DOS into the *high memory area* (the first 64K of extended memory). To do this, place the following lines into your CONFIG.SYS file:

```
DEVICE = C:\DOS\HIMEM.SYS
DOS = HIGH
```

The DOS=HIGH statement tells DOS to load itself into the high memory area. This frees up more conventional memory for DOS applications.

Invisible LAN doesn't care if DOS is loaded high or not. Invisible LAN works either way.

Using the Disk Cache with HIMEM.SYS

The Invisible LAN disk cache supports two methods for accessing extended memory. The method used depends on how you configure the Cache Data Location in the Invisible LAN initialization file:

- If the Cache Data Location is **EXTENDED**, the cache uses the BIOS (INT 15) interface for accessing extended memory.

- If the Cache Data Location is **XMS**, the cache uses the XMS (eXtended Memory Specification) interface for accessing extended memory.

If you are using HIMEM.SYS, the configuration of HIMEM.SYS must match the configuration of the disk cache, as described below:

Configuration 1: Cache Data Location is EXTENDED

If the Cache Data Location is **EXTENDED**, then you have to use the /INT15 parameter on HIMEM.SYS to allocate extended memory for the cache.

Example — Suppose you want to use 2048K of extended memory for the cache. If you set the Cache Data Location to **EXTENDED**, you need to install HIMEM.SYS as follows:

```
DEVICE = C:\DOS\HIMEM.SYS /INT15=2048
```

Configuration 2: Cache Data Location is XMS

If the Cache Data Location is **XMS**, then you must *not* use the /INT15 parameter to allocate extended memory for the cache. If the Cache Data Location is **XMS**, the cache makes direct calls to HIMEM.SYS to obtain the extended memory it needs.

Example — Suppose you want to use 2048K of extended memory for the cache. If you set the Cache Data Location to **XMS**, you need to install HIMEM.SYS as follows:

```
DEVICE = C:\DOS\HIMEM.SYS
```

In this example, you have to set the Cache Data Size to 2048, to tell Invisible LAN that you want to use 2048K of extended memory for the cache. (If you don't specify a value for Cache Data Size, the cache automatically uses all available extended memory.)

SUMMARY: HIMEM.SYS is the Microsoft device driver that controls access to extended memory. It allows DOS 5.0 to load itself into the high memory area. When using the cache together with HIMEM.SYS, the two programs must be configured so they cooperate in using extended memory.

Part 2

Advanced Installation

Setting Up Invisible LAN

Invisible LAN can be configured in a wide variety of ways, to fully exploit the capabilities of your computer system and meet your exact requirements. Chapters 2 through 6 gave quick-start procedures for installing some basic configurations. In this chapter, we begin the discussion of Invisible LAN's advanced configuration options.

The topics covered in this chapter are:

- Starting Invisible LAN
- Creating an Initialization File
- Setting Up CONFIG.SYS

Starting Invisible LAN

The configuration of Invisible LAN is controlled by a wide variety of *parameters*. Each parameter controls some specific aspect of Invisible LAN. For example, one parameter specifies whether or not to load the server; another parameter specifies the type of network hardware to use; another specifies the number of network files that the redirector can support; and so on.

You are probably familiar with computer programs that accept parameters on the DOS command line. Invisible LAN has too many parameters to be entered on the DOS command line. Therefore, the Invisible LAN parameters are stored in an *initialization file*. When you give the **NET30** command to start up Invisible LAN, it reads the initialization file and takes its parameters from the file.

The default name of the initialization file is NET30.INI. However, you can specify a different name if you want to. The procedure for starting Invisible LAN is shown below.

To start Invisible LAN, type

```
NET30 [ init_file ]
```

The command-line parameter has the following meaning:

init_file The name of an initialization file. The file must be located in the same directory as the Invisible LAN program files. The name of the initialization file must end in .INI, however, you should not type .INI on the command line. The square brackets indicate that *init_file* is optional; if omitted, it defaults to NET30.

Example—Suppose you have an initialization file named MICHAEL.INI. You would tell Invisible LAN to take its parameters from the initialization file MICHAEL.INI by entering

```
NET30 MICHAEL
```

SUMMARY: The *initialization file* contains the parameters that define the configuration of Invisible LAN. With the **NET30** command, you can specify which initialization file to use.

Creating an Initialization File

To create an initialization file, you use the Invisible LAN Setup program.

To start the Invisible LAN Setup program, type

```
SETUP30
```

If your computer has a monochrome VGA display (like many laptop computers do), the display may be easier to read if you use the command

```
SETUP30 /B
```

The program loads into memory and displays its Main Menu as shown in figure 7-1. The keys used to control the Setup program are shown in a line at the bottom of the screen. The Setup program also has on-line help. To access the on-line help, press **F1**.

NETWORK SETUP MAIN MENU

```

Copy Files to Hard Disk
Load/Create Initialization File
Easy Configuration
Advanced Configuration
Save Initialization File
Modify CONFIG.SYS
Exit from Setup Program

```

Copies Invisible LAN program files to the hard disk

HELP F1 ACTIVATE Enter CANCEL Esc SELECT ↑ ↓ PgUp PgDn Home End A-Z

Figure 7-1. Setup Program Main Menu

Step 1: Enter the File Name

Select **Load/Create Initialization File** from the Setup program's Main Menu. Then type in the name of the initialization file that you want to create. The Setup program provides a default file name of NET30.INI, but you can type over this name.

- If the file you specify does not exist, the Setup program creates the file and sets all parameters to their default values.
- If the file you specify already exists, the Setup program loads the file into memory so you can modify the parameters.

Remember that the name of the initialization file must end in .INI, and the file must be stored in the same directory as the Invisible LAN program files.

After you enter the file name, the Setup program displays the file name in the upper left corner of the screen.

Step 2: Select Parameters

Select **Advanced Configuration** from the Setup program's Main Menu. The Advanced Configuration Menu appears as shown in figure 7-2. From this menu, you can select which parameters you want to modify.

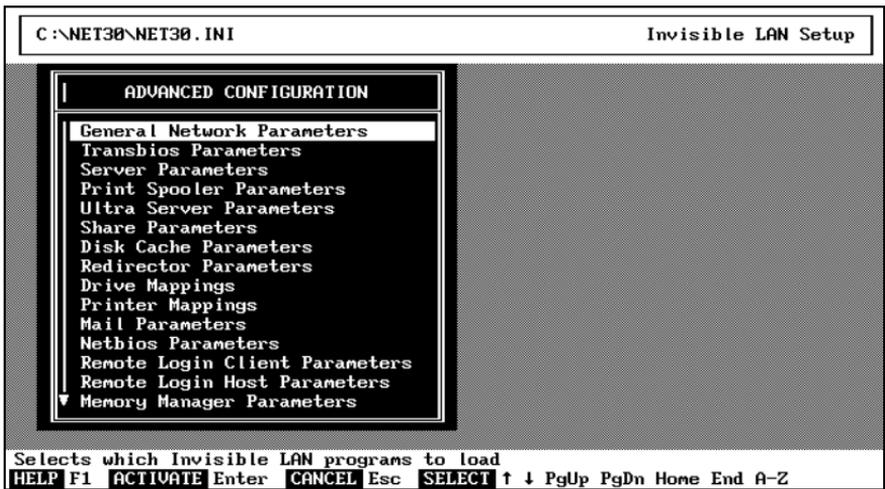


Figure 7-2. Advanced Configuration Menu

For convenience, the parameters are divided into seventeen categories. You need to select a category, and then the Setup program lets you modify the parameters in that category.

The first twelve categories control the network software, and are described in chapter 8. The next two categories control the remote login software, and are described in the *Invisible LAN Remote Instruction Manual*. The last three categories control the Invisible RAM memory managers, and are described in chapter 13.

After you have modified the parameters, press **Esc** to return to the Main Menu.

Step 3: Save the File

Select **Save Initialization File** from the Setup program's Main Menu. The Setup program writes the initialization file back out to the disk, with your parameters in the file.

You can use the new initialization file when you start Invisible LAN.

SUMMARY: With the Setup program, you can enter the name of an initialization file, modify the Invisible LAN parameters, and then save the initialization file with the new parameter values.

Setting Up CONFIG.SYS

DOS reads the text file CONFIG.SYS at startup and uses the information there to set the system configuration. In effect, CONFIG.SYS is the initialization file for DOS, just as NET30.INI is the initialization file for Invisible LAN.

When using Invisible LAN, you need to place some statements into your CONFIG.SYS file. There are two ways you can modify your CONFIG.SYS file. One way is to load the Invisible LAN Setup program (**SETUP30**) and select **Modify CONFIG.SYS** from the Main Menu. The other way is to use a text editor to edit the CONFIG.SYS file. Either method is acceptable.

Invisible LAN Device Drivers

To use Invisible LAN, you need to install two *device drivers* in the CONFIG.SYS file. The device drivers are called **CACHE30.SYS** and **N30DEV.SYS**. These device drivers use very little memory (about 100 bytes each), and it is safe to leave them in your CONFIG.SYS file all the time, even when you are not using the network.

To install the Invisible LAN device drivers, place the following two lines in the CONFIG.SYS file:

```
DEVICE=C:\NET30\CACHE30.SYS
```

```
DEVICE=C:\NET30\N30DEV.SYS
```

You can specify a drive and path different from C:\NET30, if you wish. **CACHE30.SYS** *must* be the *very first* device driver installed in your CONFIG.SYS file. Similarly, **N30DEV.SYS** *must* be the *very last* device driver installed in your CONFIG.SYS file. If there are any other **DEVICE=** statements in your CONFIG.SYS file, they must come *after* **DEVICE=CACHE30.SYS** and *before* **DEVICE=N30DEV.SYS**.

Disk Buffers

DOS uses *buffers* in memory to temporarily hold data being sent to or from the disk. In general, the more buffers you have, the faster the system runs. Having a large number of buffers is particularly important for a computer that is used as a server. Each buffer uses approximately 528 bytes of memory.

If you are using the disk cache, then the number of DOS buffers has only a slight effect on system performance. Therefore, if you use the disk cache you may want to save memory by reducing the number of buffers.

To specify the number of disk buffers, the CONFIG.SYS file should contain the command

BUFFERS=number

The parameter has the following meaning:

number The desired number of buffers (1 to 99). In general, the recommended command for Invisible LAN is **BUFFERS=20**. If you use the disk cache, you can save memory by specifying **BUFFERS=4**.

Drive Letters

The **LASTDRIVE** command in the CONFIG.SYS file specifies the number of different drive letters that you can use. The command syntax is:

LASTDRIVE=drive_letter

The parameter has the following meaning:

drive_letter An alphabetic character ranging from A to Z. It represents the last valid drive letter. For the redirector, it is important to specify a large number of different drive letters, because you need a different letter for each network disk you access. The recommended command for Invisible LAN is **LASTDRIVE=Z**.

File Control Blocks and File Handles

There are two ways to open a disk file under DOS: using a *file control block*, and using a *file handle*. The **FCBS** command in the CONFIG.SYS file specifies the number of files that can be open using file control blocks. The command syntax is:

FCBS=*max_number,prot_number*

The parameters have the following meanings:

- max_number** The maximum number of files that can be opened by file control blocks (1 to 255).
- prot_number** The number of file control blocks that cannot be closed automatically by DOS (0 to 255). If a program tries to open too many files with file control blocks, DOS automatically closes the least recently used file control block. The first **prot_number** file control blocks are protected from being closed. The recommended command for Invisible LAN is **FCBS=16,8**.

The **FILES** command in the CONFIG.SYS file specifies the number of files that can be open using file handles. The command syntax is:

FILES=*number*

The parameter has the following meaning:

- number** The maximum number of files that can be opened using file handles (8 to 255). Each handle uses approximately 48 bytes of memory. For a server, it is important to specify a large number in the **FILES** command, because each file opened by another computer on the network uses a file handle on the server. (Exception — If you configure the server to allocate a private file table, then the value of **FILES** has no effect on the number of files that can be opened by network workstations.) The recommended value for Invisible LAN is **FILES=100**.

System Stack Resources

The **STACKS** command in the CONFIG.SYS file specifies the number and size of the stacks that are used for processing interrupts. **STACKS** can only be used with DOS version 3.2 or later. The command syntax is:

STACKS=number,size

The parameters have the following meanings:

- number** The number of stacks to allocate (8 to 64).
- size** The size (in bytes) of each stack (32 to 512). The recommended command for Invisible LAN is **STACKS=32,128**. Exceptions: If you use Ultra Server (the high-performance file server), the recommended command is **STACKS=32,256**. If you use the 386 Virtual Mode memory manager (Invisible RAM 386), you can save memory by specifying **STACKS=0,0**.

SUMMARY: You use the CONFIG.SYS file to install the two device drivers required by Invisible LAN. You also use CONFIG.SYS to configure DOS so that it works smoothly in combination with Invisible LAN.

Advanced Network Configuration

This chapter describes all the parameters that are available for configuring the network software.

Invisible LAN has a large number of parameters, so you can configure it to match your exact needs. By changing parameters, you can optimize the network's performance, control the use of system memory, and configure the software to match the size of the network — anything from 2 stations to 100 or more.

All the parameters can be adjusted using the Invisible LAN Setup program (**SETUP30**). The parameters are stored in the Invisible LAN initialization file, and they go into effect the next time you start Invisible LAN.

The parameters covered in this chapter are:

- General parameters
- TransBIOS parameters
- NetBIOS parameters
- Server parameters
- Print Spooler parameters
- Redirector parameters
- Drive Mapping parameters
- Printer Mapping parameters
- Cache parameters
- SHARE parameters
- Mail parameters
- Ultra Server parameters

Parameters that control the Invisible RAM memory managers are described separately, in Chapter 13.

General Parameters

The General Parameters select which programs you want to load. They also define your user name on the network.

The general parameters are entered on the General Parameters screen, as shown in figure 8-1.

- **User Name** is your name on the network. You can enter a name from 1 to 15 characters in length. Each network user must have a different name.
- **Software Version** is the Invisible LAN software version you want to use. Use **F5** and **F6** to select the version. The available versions depend on which software you have purchased; press **F1** for a complete list.

STANDARD selects the standard version of Invisible LAN, which can be used on any computer running DOS or Windows.

ULTRA_SERVER selects the high-performance file server software, which is used mainly to set up a dedicated server. This version requires a 386 or 486 computer with extended memory. In addition, if you use Ultra Server, you cannot use any 386 memory manager (such as EMM386 or QEMM-386). Also, if possible, you should not use HIMEM.SYS. These restrictions allow Ultra Server to have direct access to extended memory.

REMOTE_LOGIN selects the remot login client software. This option is described in the *Invisible LAN Remote Instruction Manual*.

- **Network Hardware** is the type of network hardware you are using. Use **F5** and **F6** to select the type that corresponds to your network hardware.

If you need help selecting the correct hardware type, press **F1** for an explanation of the different hardware types. If you need additional help, refer to the text file `HARDWARE.TXT`, which contains an extensive description of the supported network hardware.

- **Load Server** specifies if you want to load the Invisible LAN server. The server allows other users to make use of your disks and printers. Enter **Y** or **N**.
- **Load SHARE** indicates whether to load the DOS SHARE program. You *must* load SHARE if you are loading the server. Enter **Y** or **N**. **Note** — There are two ways to load SHARE: you can enter **Y** for this parameter and let Invisible LAN load SHARE; or you can load the SHARE program from the DOS command line (or batch file) before starting Invisible LAN.
- **Load Cache** specifies if you want to load the Invisible LAN disk cache. The disk cache speeds up hard disk operation by keeping recently used disk data in memory. The disk cache is very beneficial for servers, however it can be used on workstations as well. Enter **Y** or **N**.

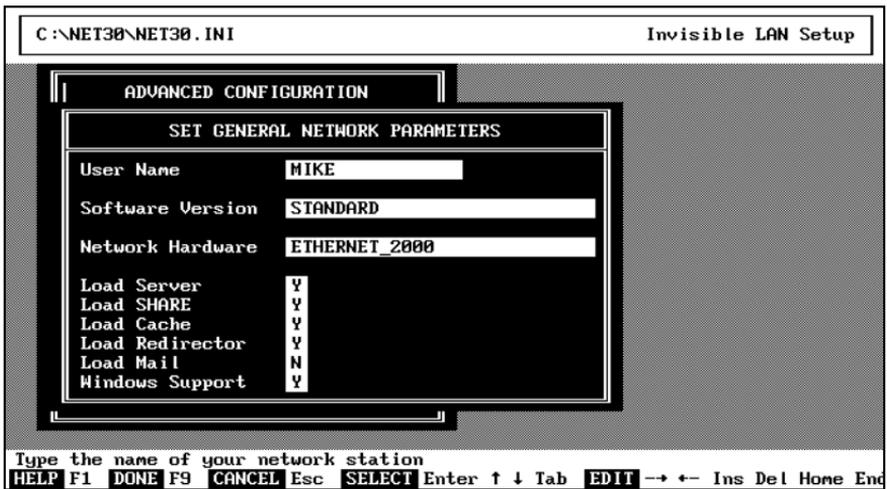


Figure 8-1. General Parameters screen

- **Load Redirector** specifies if you want to load the Invisible LAN redirector. The redirector lets you use disks and printers that are attached to servers. Enter **Y** or **N**.
- **Load Mail** indicates whether or not to load the electronic mail program. Mail lets you send brief, one-line messages to other users. Enter **Y** or **N**. Note that if you load mail, then you must also load the redirector.
- **Windows Support** indicates whether or not to load the Windows version of Invisible LAN. You must load the Windows version if you want to run Microsoft Windows. Enter **Y** or **N**.

SUMMARY: The General Parameters screen lets you select which Invisible LAN programs to load, and define your user name on the network.

TransBIOS Parameters

TransBIOS is the basic network communication software used by Invisible LAN. *TransBIOS* is responsible for controlling the network hardware and providing communication services used by all the other Invisible LAN programs.

The *TransBIOS* parameters are entered on the *TransBIOS* Parameters screen, as shown in figure 8-2.

ADVANCED CONFIGURATION

SET TRANSMIBIOS PARAMETERS

Code Memory Location	DOS	Applications (4-256)	16
Data Memory Location	DOS	Data Size (0-256)	
Retry Limit	32	Hardware I/O Port	
Sessions (4-256)	32	Hardware IRQ Level	
Names (4-256)	18	Hardware DMA Channel	
Polls (2-16)	4	Hardware Memory Address	
Responses (4-256)	16	Maximum Packet Size	

Press F5 or F6 to select the type of memory to use for TransBIOS program code
 HELP F1 DONE F9 CANCEL Esc SELECT Enter ↑ ↓ Tab OPTION F5=Back F6=Next A-Z

Figure 8-2. TransBIOS Parameters Screen

RAM Versus ROM

There are two ways to start TransBIOS. One way is to load TransBIOS into memory from disk, using the **NET30** command. The second way is to use the TransBIOS/NetBIOS ROM located on the network card; if you use the BIOS ROM, then TransBIOS starts automatically when you turn on your computer. (**Note** — Not all network cards have TransBIOS/NetBIOS ROMs.)

If your computer is a 286, 386, or 486, then the RAM version of TransBIOS is considerably faster than the ROM version. For this reason, it is generally a good idea to disable the TransBIOS/NetBIOS ROM and use the RAM version.

All the parameters described in this section are effective with the RAM version of TransBIOS. The parameters are *not* effective with the ROM version of TransBIOS, except for the **Retry Limit** and **Data Location** parameters. The ROM version of TransBIOS simply uses the default values for all the parameters (except the two mentioned).

Memory Allocation

With the memory allocation parameters, you can control the amount and type of memory (conventional, shadow, or EMS) used by TransBIOS. These parameters also control the type of memory used by NetBIOS.

The TransBIOS program is divided in two pieces: a *code segment* and a *data segment*. The code segment contains the TransBIOS program code, and is

approximately 16K in size. The data segment contains TransBIOS data tables and network packet buffers, and is variable size. (On Ethernet hardware, a typical size for the data segment is 24K.) Separate parameters are provided to control these two segments.

- **Code Memory Location** specifies the type of memory used for the TransBIOS code segment: **DOS**, **SHADOW**, or **EMS**. To use **SHADOW**, you must load a memory manager that provides shadow RAM or upper memory. To use **EMS**, you must have expanded memory in your system compatible with EMS version 4.0. The default value for this parameter is **DOS**. **Note** — For the ROM version of TransBIOS, the code segment is located in the ROM.
- **Data Memory Location** specifies the type of memory used for the TransBIOS data segment: **DOS** or **SHADOW**. To use **SHADOW**, you must load a memory manager that provides shadow RAM or upper memory. The default value for this parameter is **DOS**. **Note** — This parameter is effective for both RAM and ROM versions of TransBIOS.
- **Data Size** specifies the size of the TransBIOS data segment, in kilobytes. The value can range from 0 to 256. The default value varies depending on the type of network hardware you use; the default is 24 for most Ethernet boards. To specify the default value, leave this field blank. Entering 0 does not give a zero-size data segment; instead, it gives the minimum allowed segment size (which varies depending on the network hardware, and is only a little smaller than the default size). On a dedicated file server, you can improve performance by increasing this parameter so that there are more network packet buffers available to handle workstation requests.

Timing

There is one parameter to control the session timeout used by TransBIOS.

When you reboot a server, all the workstations automatically re-map their drives and printers by creating new network sessions with the server. However, before this can happen, the old network sessions have to be aborted. The following parameter controls how long it takes to abort old network sessions.

- **Retry Limit** is the number of times that TransBIOS retries a network transmission before aborting a session. Each retry takes one or two seconds. Thus, this parameter controls the amount of time it takes for network sessions to die. The default value is 32. The allowable range of values varies depending on the network hardware; however, if the value you enter is out of range, TransBIOS simply uses the closest legal value.

Resource Allocation

These parameters are very technical in nature, and most users will not have to modify them. They control TransBIOS resources, which determine how many different things TransBIOS can do simultaneously (for example, how many network sessions there can be).

In general, increasing these parameters increases the memory consumed by TransBIOS data tables (which are part of the TransBIOS data segment).

- **Sessions** is the number of network sessions that TransBIOS supports. The value can range from 4 to 256, and the default is 32. A *session* is a “connection” between two programs on the network, that allows them to transfer data. For example, the server has a session with each redirector that is mapped to it, and the mail program uses a session to send mail messages across the network. Thus, this parameter determines how many programs you can be “connected” to simultaneously. If you are making a server that needs to support more than 30 workstations, you need to increase this parameter.
- **Names** is the number of network names that TransBIOS supports. The value can range from 4 to 256, and defaults to 18. Most network programs identify themselves on the network by name. So this parameter indirectly determines how many network programs you can use at one time. **Note** — TransBIOS uses 2 names internally, and Invisible LAN uses 2 or 3 names.
- **Applications** is the number of TransBIOS application programs that can be running on your computer at one time. Allowed values are 4 to 256, and the default is 16. The server, the redirector, the network manager, NetBIOS, and TransBIOS each count as one application.
- **Polls** is the number of network poll commands that can be processed concurrently. Allowed values are 2 to 16, with a default of 4. **Note** — There is no known reason to change this parameter.
- **Responses** is the number of network poll responses that can be pending concurrently. Allowed values are 4 to 256, with a default of 16. **Note** — There is no known reason to change this parameter.

Hardware Configuration

When TransBIOS loads, it automatically detects the configuration of the network hardware. This includes such things as the hardware’s I/O port address or interrupt level.

In some instances, you may want to override the auto-detection mechanism and manually specify the hardware configuration. You might want to do this if the auto-detect mechanism is interfering with other equipment in the computer; or if your hardware has an unusual configuration that the auto-detect mechanism can't handle; or if you have two network boards installed and you want to direct TransBIOS to use a particular board. In these cases, you can use the following parameters to specify the hardware configuration. Refer to `HARDWARE.TXT` for a description of how these parameters affect the particular type of network hardware you're using.

- **Hardware I/O Port** is the address of the network hardware I/O port, in hexadecimal. The allowed values depend on the type of hardware you're using. If you leave this field blank, TransBIOS attempts to detect the port address automatically.
- **Hardware IRQ Level** is the interrupt level used by the network hardware. It can be a decimal number from 2 to 15. (On XT-type computers, it can be from 2 to 7.) If you leave this field blank, TransBIOS attempts to detect the interrupt level automatically.
- **Hardware DMA Channel** is the DMA (direct memory access) channel used by the network hardware. It can be a decimal number from 0 to 7. (On XT-type computers, it can be from 1 to 3.) Most network hardware does not use a DMA channel. If you leave this field blank, TransBIOS attempts to detect the DMA channel automatically.
- **Hardware Memory Address** is the segment address of the network hardware's shared memory, in hexadecimal. The allowed values depend on the type of hardware you are using. Most hardware does not use any shared RAM. If you leave this field blank, TransBIOS attempts to detect the memory address automatically.

Packet Size

There is one parameter you can use to specify the size of the packets that TransBIOS sends on the network.

Normally, TransBIOS sends the largest possible packets, since these are the most efficient. This can create a problem if your network includes devices like bridges and routers that cannot handle large packets. For example, many Ethernet bridges and routers cannot handle packets larger than 1514 bytes. In this case, you can use this parameter to force TransBIOS to limit its packets to a size that the bridges and routers can handle.

- **Maximum Packet Size** is the largest packet that TransBIOS can send on the network, in bytes. You can enter a decimal number to specify the maximum packet size, or leave this field blank to use the default. The

default packet size varies depending on the type of network hardware. This parameter can only reduce the packet size below the default, it cannot increase the packet size.

Note — On most Ethernet hardware, TransBIOS uses packets that are larger than the official Ethernet specification allows. Enter 1514 if you want TransBIOS to adhere strictly to the official Ethernet specification.

SUMMARY: The TransBIOS parameters control the type and amount of memory used by TransBIOS, the timing of TransBIOS sessions, the amount of resources that TransBIOS makes available, the network hardware configuration, and the network packet size.

NetBIOS Parameters

NetBIOS is a standard network communication interface defined by IBM. Many network application programs use NetBIOS to establish communication between different computers on the network.

Invisible LAN comes with two network protocols: NetBIOS and TransBIOS. All the Invisible LAN programs (server, redirector, etc.) use TransBIOS, since TransBIOS is faster. However, NetBIOS is included in order to make Invisible LAN compatible with all the application programs that require NetBIOS.

The NetBIOS parameters are entered on the NetBIOS Parameters screen, as shown in figure 8-3.

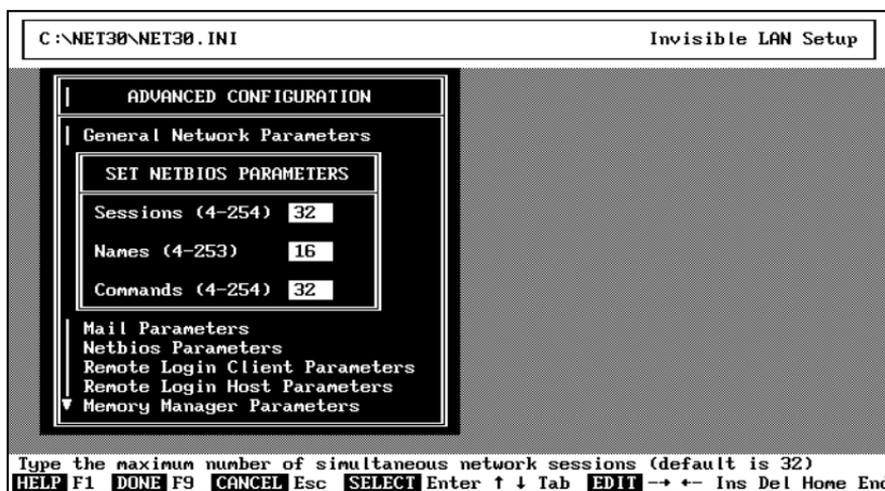


Figure 8-3. NetBIOS Parameters Screen

RAM Versus ROM

There are two ways to start NetBIOS. One way is to load NetBIOS into memory from disk, using the **NET30** command. The second way is to use the TransBIOS/NetBIOS ROM located on the network card; if you use the ROM, then NetBIOS starts automatically when you turn on your computer. (**Note** — Not all network cards have TransBIOS/NetBIOS ROMs.)

If your computer is a 286, 386, or 486, then the RAM version of NetBIOS is considerably faster than the ROM version. For this reason, it is generally a good idea to disable the TransBIOS/NetBIOS ROM and use the RAM version.

All the parameters described in this section are effective with the RAM version of NetBIOS. The parameters are *not* effective with the ROM version of NetBIOS.

Memory Allocation

There are no parameters to control the type of memory (conventional, shadow, or EMS) used by NetBIOS. NetBIOS automatically uses the same type of memory as TransBIOS. Refer to the previous section in this chapter for details.

Resource Allocation

These parameters are very technical in nature, and most users will not have to modify them. They control NetBIOS resources, which determine how many different things NetBIOS can do simultaneously (for example, how many network sessions there can be).

Invisible LAN itself makes very little use of NetBIOS (it uses TransBIOS instead). Therefore, NetBIOS resources are needed only for application programs that use NetBIOS.

- **Sessions** is the number of network sessions that NetBIOS supports. The value can range from 4 to 254, and the default is 32. A *session* is a “connection” between two programs on the network, that allows them to transfer data. **Note** — Each active NetBIOS session requires one TransBIOS session, so if you increase this parameter you should also increase the TransBIOS sessions parameter.
- **Names** is the number of network names that NetBIOS supports. The value can range from 4 to 253, and defaults to 16. Most network programs identify themselves on the network by name. **Note** — Invisible LAN uses one NetBIOS name, plus one additional name if the Remote Program

Load Server is started. **Note** — Each active NetBIOS name requires one TransBIOS name, so if you increase this parameter you should also increase the TransBIOS names parameter.

- **Commands** is the number of NetBIOS commands that can be outstanding at one time. Allowed values are 4 to 254, and the default is 32.

SUMMARY: The NetBIOS parameters control the amount of resources that NetBIOS makes available.

Server Parameters

The *server* is the program that makes your disks and printers available to other users on the network. The parameters described in this section let you configure the server.

There are two main reasons for changing the server configuration. One is to control the type of memory that the server uses (conventional, shadow, or EMS). The second reason is to control the number of workstations that the server can support, and the resources that the server provides to those workstations.

The server parameters are entered on the Server Parameters screen, as shown in figure 8-4.

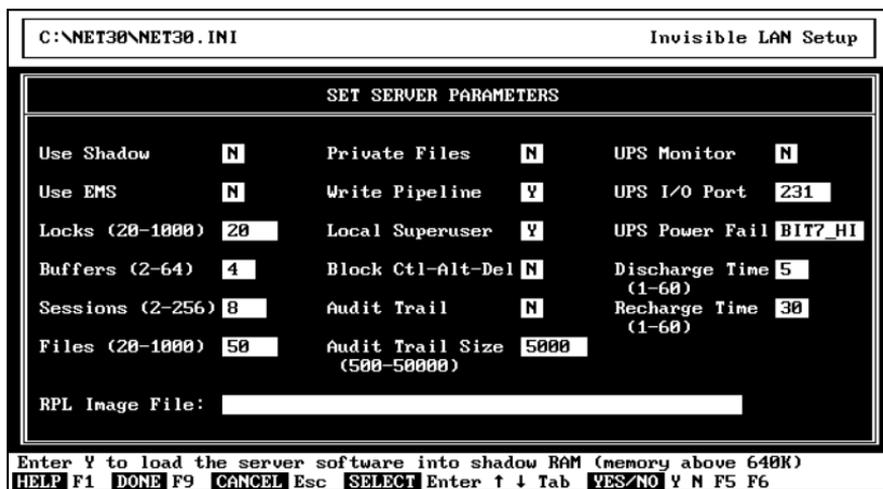


Figure 8-4. Server Parameters Screen

Memory Allocation

The server can use conventional DOS memory, shadow RAM, or EMS (expanded memory). In order to use shadow RAM, you must install a memory manager that provides shadow RAM or upper memory. To use EMS, you must have expanded memory compatible with the Expanded Memory Specification version 4.0.

- **Use Shadow** specifies whether or not the server should use shadow RAM. Enter **Y** if you want the server to load itself into shadow RAM. Enter **N** if you want the server to load itself in conventional DOS memory. The default value for this parameter is **N**.
- **Use EMS** specifies whether or not the server should use expanded memory. If you enter **Y**, then most of the server software is loaded into expanded memory, and the remainder is loaded into conventional DOS memory. If you enter **N**, the server does not use expanded memory. The default value for this parameter is **N**. Note that loading the server into expanded memory significantly reduces performance.

Note — You can specify **Y** for both **Use Shadow** and **Use EMS**. If you do, then most of the server software is loaded into expanded memory, and the remainder is loaded into shadow RAM.

Resource Allocation

These parameters control the amount of resources that the server makes available to the network. They determine the number of workstations that can use the server. They also determine the number of network files and record locks that the server supports. Additionally, they determine the amount of buffer space that the server allocates for network buffers.

The default values of these parameters are generally adequate for supporting up to six workstations. You will want to increase these parameters if more than six workstations are using the server. You will also want to increase these parameters if your workstations make unusually heavy demands on the server.

- **Buffers** is the number of network buffers allocated by the server. You can enter a number from 2 to 64; the default value is 4. Each buffer uses 2K bytes of memory. Increasing this value will increase the speed of the server (however, there is no benefit to providing more than 4 buffers per workstation). For a dedicated server, we strongly recommend increasing the value of this parameter.
- **Sessions** is the number of network sessions that the server supports. This determines the number of workstations that can use the server. You can enter a number from 2 to 256; the default value is 8. It is generally a good

idea to allow at least two more sessions than the number of workstations; for example, if you have 12 workstations then you should provide at least 14 sessions. **Note** — If you enter a number larger than 30, you will need to increase the number of sessions that TransBIOS supports.

- **Files** is the number of network files that can be open concurrently. This limits the total number of files that can be opened by all workstations using the server. You can enter a value from 20 to 1000; the default value is 50. **Note** — The number of network files is also limited by the **Filespace** parameter of the DOS SHARE program. In addition, if you specify **N** for **Private Files**, then the number of network files is also limited by the FILES= statement in the server's CONFIG.SYS file. (Therefore, **Files** values larger than 250 are not useful unless you specify **Y** for **Private Files**.)
- **Private Files** specifies whether or not the server should allocate a private file table. If you specify **N**, then the number of open files is limited by the FILES= statement in the server's CONFIG.SYS file. If you specify **Y**, then the number of open files is *not* limited by the FILES= statement in the server's CONFIG.SYS file. **Note** — If you allocate a private file table, then you cannot run Windows in 386 enhanced mode on the server.
- **Locks** is the number of record locks that the server supports. The value can range from 20 to 1000, with 20 as the default. Record locks are used mainly by database applications. **Note** — The number of record locks is also limited by the **Locks** parameter of the DOS SHARE command.

Remote Program Load

This parameter is only used if you have diskless workstations in the network. With diskless workstations, you need to set up one computer in the network to act as a Remote Program Load Server. The *Remote Program Load Server* supplies the diskless workstations with a copy of DOS and other software that they need.

Refer to the *DOS User Manual* for further information on setting up a diskless workstation.

- The **RPL Image File** parameter is used to make the server into a Remote Program Load (RPL) Server. If you want to enable the RPL Server, you need to enter the name of a disk file that contains a *boot diskette image*. You must enter the complete drive and path for the file (for example, C:\NET30\BOOT.IMG). You need to use the **MAKEIMG** program to create the boot diskette image file; refer to the *DOS User Manual* for details. If you leave this parameter blank, then the RPL Server is not enabled.

Note — If you are using Ultra Server, then the RPL image file must be located in the network directory (usually C:\NET30), or in a subdirectory of the network directory, or on a drive that is not cached by Ultra Server.

Security Options

These parameters control the operation of the security system on the server.

- **Local Superuser** specifies if a person working locally at the server should be a *superuser* who has full access to all server resources, regardless of security permissions. If you enter **Y** (the default), the local user is a superuser. If you enter **N**, then the local user is subjected to the same security checking as any other user.
- **Block Ctl-Alt-Del** specifies whether or not to deactivate the **Ctrl-Alt-Del** key. If you enter **Y**, then you can't reboot the server by pressing **Ctrl-Alt-Del**.
- **Audit Trail** specifies whether or not the server should maintain an *audit trail*, which is a disk file that records all server activities, such as opening files or submitting print jobs. Refer to the *DOS User Manual* or *Windows User Manual* for additional information on audit trails.
- **Audit Trail Size** is the desired size of the audit trail file. You should enter a number from 500 to 50000; the default is 5000. The number represents the number of records in the audit trail file; each record occupies 48 bytes of disk space. This parameter is only used when the server creates a new audit trail file. If there is an existing audit trail file, the server uses the size of the existing file.

UPS Monitoring

The server can monitor a UPS (*Uninterruptable Power Supply*). This allows the server to automatically shut itself down in an orderly manner when there is a power failure.

- **UPS Monitor** specifies whether or not the server monitors a UPS. Enter **Y** if you want the server to monitor a UPS; otherwise enter **N**.
- **UPS I/O Port** is the address of the I/O port that the server uses to read the status of the UPS. You must enter the address in hexadecimal.
- **UPS Power Fail** is the signal that the UPS generates when the power fails. By pressing **F5** and **F6**, you can select **BIT0_LO** through **BIT7_LO** and **BIT0_HI** through **BIT7_HI**. Selecting **BIT0_LO** means that bit 0 in

the I/O port is low when power has failed; selecting **BIT0_HI** means that bit 0 in the I/O port is high when power has failed; selecting **BIT1_LO** means that bit 1 in the I/O port is low when power has failed; and so on.

- **Discharge Time** is the amount of time, in minutes, that the server waits before shutting down when there is a power failure. If power is restored within this time, the server does not shut down. You can enter a number from 1 to 60; the default is 5. You should select a time that ensures the server will shut down long before the UPS batteries are exhausted.
- **Recharge Time** is the amount of time, in minutes, to wait before restarting the server when power is restored after a shutdown. You can enter a number from 1 to 60; the default is 30. You should try to set this time large enough to allow the UPS batteries to recharge.

Note — For a “Novell Type 1” UPS monitoring card, the I/O address is usually **231**, and the power fail signal is usually **BIT7_HI**.

Pipeline Parameter

This parameter is highly technical in nature. You will probably not have to change its value.

- The **Write Pipeline** parameter specifies whether or not the server can use pipelined writes during file I/O. This means that write data is cached in the server, and the workstation is immediately given permission to proceed while the data is written to the file at a later time. This speeds up file writes significantly. However, the write pipeline may prevent some error conditions from being reported to the workstation (for example, a totally full hard disk). Also, it is conceivable that some software may not work properly with the write pipeline enabled (although we do not know of any such software). You can enter **Y** to enable write pipeline, or **N** to disable write pipeline; the default is **Y**. **Note** — The server’s write pipeline is totally separate from the disk cache’s write-back feature.

SUMMARY: The server parameters control the type and amount of memory used by the server, and the amount of resources that the server makes available to users. They also enable the remote program load function, security options, UPS monitoring, and file write optimization.

Print Spooler Parameters

The *print spooler* controls the printers that you share with other users on the network. It stores user's print jobs in the *print queue*, and sends the jobs to the printer one-by-one.

The print spooler is not actually a separate program; it is part of the server. Using the parameters described in this section, you can define which printers you want to make available for other network users. You can also define the size and location of the print queue.

The print spooler parameters are entered on the Print Spooler Parameters screen, as shown in figure 8-5.

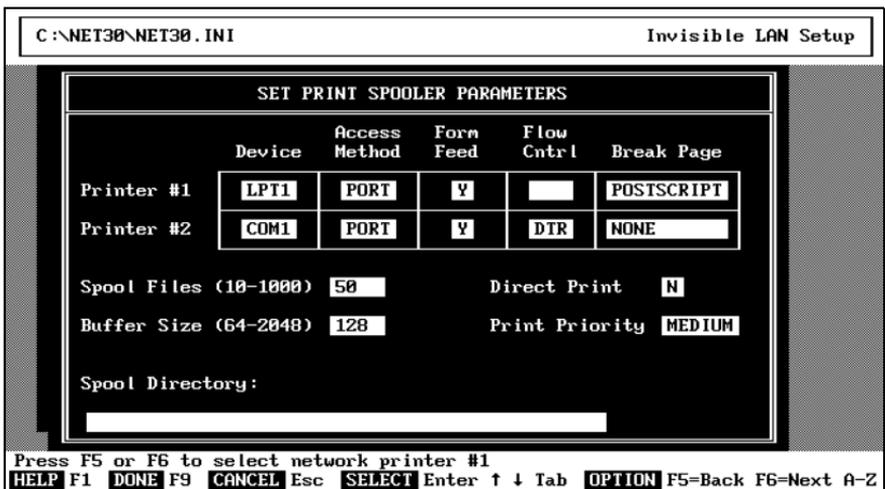


Figure 8-5. Print Spooler Parameters Screen

Defining Printers

You can define one or two network printers. Each printer that you define is accessible to other users on the network. If you define two network printers, then the first printer is called *network printer #1*, and the second printer is called *network printer #2*. If you don't define any network printers, then other network users cannot use your printers.

For each network printer, you have to specify the following five parameters:

- **Device** is the parallel port or serial port where the printer is connected. Using **F5** and **F6**, you can select from **LPT1**, **LPT2**, **LPT3**, **COM1**, or

COM2. If you don't want to define a network printer, select **NONE**.
Note — You need to enter the port name where the printer is physically connected; in particular, for a serial printer you must enter **COM1** or **COM2**, even if you normally use the DOS MODE command to redirect LPT output to the serial printer.

- **Access Method** specifies the method used to access the printer port. This parameter is used only for a parallel printer; it is ignored for a serial printer. You can select **PORT** or **BIOS**. If you select **PORT**, the print spooler sends data directly to the printer port hardware. If you select **BIOS**, the print spooler uses the printer BIOS (INT 17h).
- **Form Feed** specifies whether or not to insert a page feed command at the end of each print job. Enter **Y** or **N**; the default value is **Y**. You should always enter **N** if you are using a PostScript printer.
- **Flow Control** specifies the method used to send data to a serial printer. This parameter is ignored for a parallel printer. You can select **DTR** (for *data terminal ready*), or **XON** (for *transmit-on/transmit-off* or *XON/XOFF*). It is outside the scope of this manual to describe the technical differences between these two methods. However, the default value of **DTR** should work with most printers.
- **Break Page** tells the print spooler to print a break page at the start of each print job. The *break page* contains the user name, job title, date and time, and a graphic design. If you want a break page, select **ASCII** if you are using an ASCII printer, or **POSTSCRIPT** if you are using a PostScript printer, or **PCL** if you are using a PCL printer. If you don't want a break page, select **NONE**.

Important — Whenever you share a printer, you should always establish a mapping to the shared printer, so that your own printer output will go through the print spooler. Use the Printer Mapping Parameters screen, described later in this chapter, to establish the mapping.

Note — You can change the form feed and flow control settings while Invisible LAN is running. You can also change the design of the break page. Refer to the *DOS User Manual* or *Windows User Manual* for details.

Note — When using a serial printer, you need to use the DOS MODE command to configure the serial port. Using MODE, you set the baud rate, parity, number of data bits, and number of stop bits. For example, to set COM1 to 9600 baud, no parity, 8 data bits, and 1 stop bit, you would use

```
MODE COM1:9600,N,8,1,P
```

With a shared printer, you should never use the MODE command to redirect LPT output to the serial printer (as in **MODE LPT1=COM1**). Instead, use a printer mapping to make LPT output go to the serial printer, through the print spooler.

Print Queue Configuration

Print jobs are stored on the disk in temporary files. These files are in the *print queue*. The files stored in the print queue are sent to the printer one-by-one.

You can use the following parameters to control the configuration of the print queue.

- **Spool Files** determines the maximum number of files that can be in the print queue. You can enter a number from 10 to 1000; the default value is 50.
- **Spool Buffer Size** is the size of the memory buffer that is used to transfer data from the temporary print files to the printer. You can enter a number from 64 to 2048; the default value is 128. If you are printing ordinary text files, the default value should be adequate. If you are printing graphics, or using a page description language (such as PostScript), you may be able to increase printing speed significantly by increasing this parameter.
- **Spool Directory** is the disk drive and directory where the temporary print files are stored. By default, the print spooler uses the same directory where the Invisible LAN program files are stored. However, you can specify a different directory by entering a full drive and path. It is highly recommended that you use a RAMDISK for the temporary files, as this will speed up the printing.

Note — If you are using Ultra Server, then the temporary print files must be located in the network directory (usually C:\NET30), or in a subdirectory of the network directory, or on a drive that is not cached by Ultra Server.

- **Direct Print** indicates whether or not you want the print spooler to use *direct printing*. If you enter **N** (the default value), then the print spooler does not send a job to the printer until the entire job has been received from the workstation. If you enter **Y**, then the print spooler can send a job to the printer at the same time that the job is being received from the workstation.
- **Print Priority** determines the amount of computer processing time that is devoted to sending jobs to the printer. You can select **HIGH**, **MEDIUM**, or **LOW**. The default is **MEDIUM**. A higher priority gives more time to printing and less time to the user working locally at the server.

SUMMARY: The Print Spooler Parameters let you define which printers to share with the network, and the characteristics of the print queue.

Redirector Parameters

The *redirector* is the program that lets you use disks and printers belonging to other users on the network. The parameters described in this section let you configure the redirector.

There are two main reasons for changing the redirector configuration. One is to control the type of memory that the redirector uses (conventional, shadow, or EMS). The second reason is to control the number of servers that the redirector can support, and the resources that the redirector provides to application programs.

The redirector parameters are entered on the Redirector Parameters screen, as shown in figure 8-6.

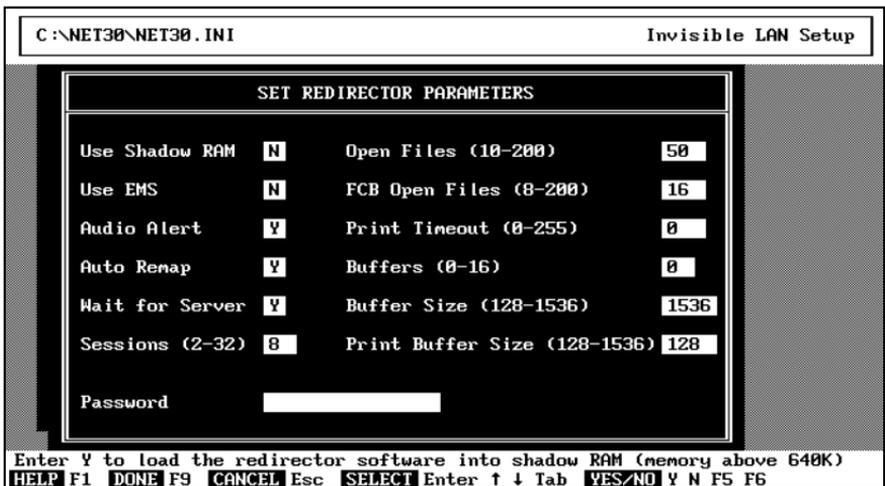


Figure 8-6. Redirector Parameters Screen

Memory Allocation

The redirector can use conventional DOS memory, shadow RAM, or EMS (expanded memory). In order to use shadow RAM, you must install a memory manager that provides shadow RAM or upper memory. To use EMS, you must have expanded memory compatible with the Expanded Memory Specification version 4.0.

- **Use Shadow RAM** specifies whether or not the redirector should use shadow RAM. Enter **Y** if you want the redirector to load itself into shadow RAM.

shadow RAM. Enter **N** if you want the redirector to load itself in conventional DOS memory. The default value for this parameter is **N**.

- **Use EMS** specifies whether or not the redirector should use expanded memory. If you enter **Y**, then most of the redirector software is loaded into expanded memory, and the remainder is loaded into conventional DOS memory. If you enter **N**, the redirector does not use expanded memory. The default value for this parameter is **N**. Note that loading the redirector into expanded memory significantly reduces performance.

Note — You can specify **Y** for both **Use Shadow RAM** and **Use EMS**. If you do, then most of the redirector software is loaded into expanded memory, and the remainder is loaded into shadow RAM.

Resource Allocation

These parameters control the amount of resources that the redirector makes available to application programs. They determine the number of servers that the redirector can be connected to simultaneously. They also determine the number of network files available.

The default values of these parameters are generally adequate for most networks. You will want to increase these parameters if you need to use more than six servers simultaneously. You will also want to increase these parameters if your application software opens an unusually large number of network files.

- **Sessions** is the number of network sessions that the redirector supports. This determines the number of servers that you can use simultaneously. You can enter a number from 2 to 32; the default value is 8. It is generally a good idea to allow at least two more sessions than the number of servers; for example, if you need to use 12 servers, then you should provide at least 14 sessions.
- **Open Files** is the number of network files that can be open concurrently. You can enter a value from 10 to 200; the default value is 50. **Note** — The number specified for this parameter, plus the value specified in the workstation's CONFIG.SYS file, must not exceed 250. For example, if you have FILES=100 in your CONFIG.SYS file, then you should not enter a number larger than 150 for this parameter.
- **FCB Open Files** is the maximum number of network files that can be open concurrently using the FCB method. The value can range from 8 to 200, and defaults to 16.

Redirector Cache Allocation

These parameters let you allocate a cache for network files. Allocating a cache can improve the performance of applications that perform sequential file access using a small record size. It can also improve performance for applications that do repeated polling, such as repeatedly attempting to lock a record or open a file until the record or file becomes available.

- **Buffers** is the number of buffers to allocate for caching network files. You can enter a number from 0 to 16; the default value is 0. Allocating buffers can improve the performance of applications that perform sequential file access with a small record size.
- **Buffer Size** is the size of the buffers used for caching network files, in bytes. You can enter a value from 128 to 1536; the default is 1536. In general, larger buffers make the caching more effective.
- **Print Buffer Size** is the size of the buffers used for caching output to network printers, in bytes. There are always three of these buffers, one each for LPT1, LPT2, and LPT3. You can enter a value from 128 to 1536; the default is 128. Increasing this value may speed up printing.

Miscellaneous Parameters

The following parameters are used to control miscellaneous aspects of the redirector configuration.

- **Print Timeout** is the automatic print stream truncation time, measured in seconds. Some application programs do not send an end-of-file at the end of a print job. With this parameter, you can specify that the redirector automatically end the print job whenever there is no printer activity for the specified time. You can enter a number from 0 to 255. If you enter 0, then the automatic print stream truncation is disabled. The default value for this parameter is 0.
- **Password** is your password. You have three options: (1) You can enter your password, in which case the password is established automatically when you start Invisible LAN. (2) You can enter "*" into the field, in which case the redirector prompts you to enter your password when you start Invisible LAN. (3) You can leave the field blank, in which case no password is established for you when you start Invisible LAN.
- **Audio Alert** determines whether or not Invisible LAN can use the computer's speaker. If you enter **Y**, the redirector beeps the speaker when a file server is slow in responding, and the mail program beeps the speaker when you receive mail. If you enter **N**, then Invisible LAN does not use the speaker at all. The default value for this parameter is **Y**.

- **Auto Remap** tells the redirector what to do when a server is re-booted. If you enter **Y**, the redirector automatically re-establishes its drive and printer mappings. If you enter **N**, the redirector does not re-establish the mappings; instead, it returns the “Abort, Retry, Fail?” error message. The default for this parameter is **Y**.
- **Wait for Server** tells the redirector whether or not it should wait for the server to be started when the initial drive and printer mappings are established. This parameter affects *only* the initial mappings established when Invisible LAN is first started (see the next two sections of this chapter). Enter **Y** or **N**; the default value is **Y**.

Note — After Invisible LAN is started, you can change the settings for print timeout, password, and audio; refer to the *DOS User Manual* or *Windows User Manual* for details.

SUMMARY: The Redirector Parameters let you control the type of memory (DOS, shadow, or EMS) used by the redirector. They also let you allocate resources for application programs to use, allocate a cache for network files, and configure miscellaneous aspects of the redirector.

Drive Mapping Parameters

A *drive mapping* makes one of your drive letters refer to a server’s disk.

With the drive mapping parameters, you can define the *initial* drive mappings. When you start Invisible LAN, it automatically establishes the initial drive mappings. After Invisible LAN is started, you can change the initial drive mappings or establish new mappings; refer to the *DOS User Manual* or *Windows User Manual* for instructions.

The initial drive mappings are shown on the Drive Mapping Parameters screen, as in figure 8-7. For each possible drive letter from A: to Z:, the screen shows the initial mapping for the drive letter. If there is no mapping for a given drive letter, the word “Local” is shown, indicating that the drive letter refers to a local drive rather than a network drive. By default, all drive letters are initially local.

To change the initial drive mappings, use the arrow keys to select a drive letter, and then press **Enter**. You are prompted to enter the following information:

- Server’s name
- Server’s drive letter or shortname
- Directory path (optional)

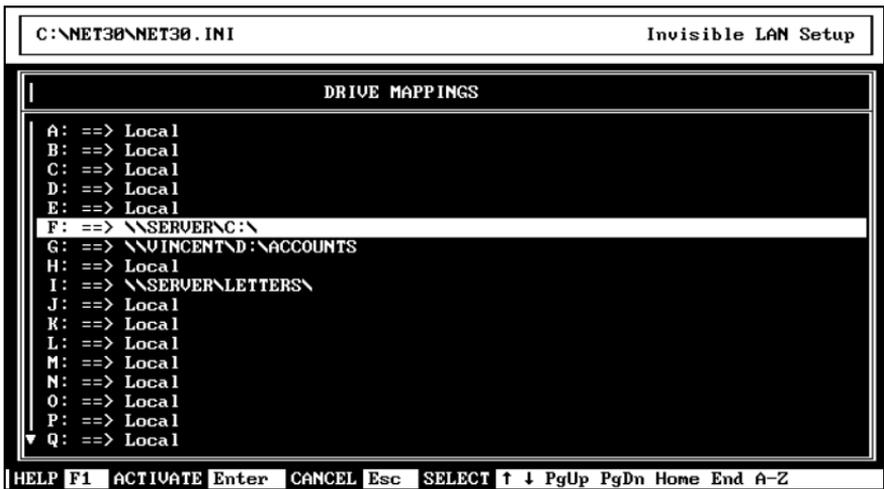


Figure 8-7. Drive Mapping Parameters Screen

SUMMARY: The Drive Mapping Parameters let you define the initial drive mappings that are automatically established when Invisible LAN is started.

Printer Mapping Parameters

A *printer mapping* makes one of your print devices (LPT1, LPT2, or LPT3) refer to a server's printer.

With the printer mapping parameters, you can define the *initial* printer mappings. When you start Invisible LAN, it automatically establishes the initial printer mappings. After Invisible LAN is started, you can change the initial printer mappings or establish new mappings; refer to the *DOS User Manual* or *Windows User Manual* for instructions.

The initial printer mappings are shown on the Printer Mapping Parameters screen, as in figure 8-8. For each possible print device, the screen shows the initial mapping. If there is no mapping for a given print device, the word "Local" is shown, indicating that the print device refers to a local printer rather than a network printer. By default, all print devices are initially local.

To change the initial printer mappings, use the arrow keys to select a print device, and then press **Enter**. You are prompted to enter the following information:

- Server's name
- Server's network printer number (1 or 2)

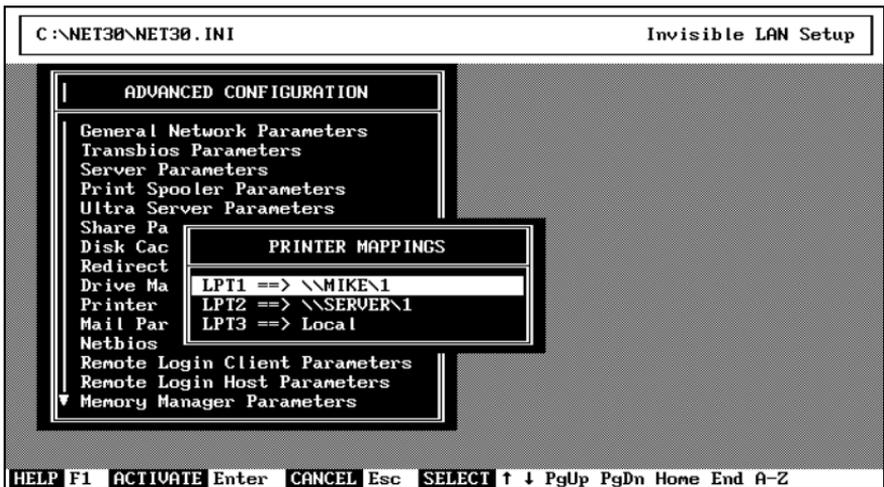


Figure 8-8. Printer Mapping Parameters Screen

Important — Whenever you share a printer, you should always establish a mapping to the shared printer, so that your own printer output will go through the print spooler. Use the Printer Mapping Parameters screen to establish the mapping. For example, suppose your name is ESTHER, and you are sharing your printer LPT1 as network printer #2. You would use the Printer Mapping Parameters screen to establish a mapping for LPT1; you would enter ESTHER as the server's name, and 2 as the server's network printer number.

SUMMARY: The Printer Mapping Parameters let you define the initial printer mappings that are automatically established when Invisible LAN is started.

Cache Parameters

The *disk cache* is a program that speeds up the operation of hard disk drives. It speeds up disk reads by storing disk data in memory, where it can be accessed very quickly. It speeds up disk writes by storing write data in memory, and then writing it out at a later time, in background. In addition, the cache uses a track buffering technique that speeds up disk operations by reading or writing complete disk tracks, rather than individual sectors. Also, the cache uses an elevator seek technique that reorders disk operations to minimize the movement of the disk head.

The cache is particularly valuable when installed on a file server, because all the workstations using the server benefit from the increased performance. The parameters described in this section let you configure the cache.

There are two main reasons for changing the cache configuration. One is to control the type and amount of memory that the cache uses. The second reason is to fine-tune the cache to get the best performance on your system.

The cache parameters are entered on the Cache Parameters screen, as shown in figure 8-9.

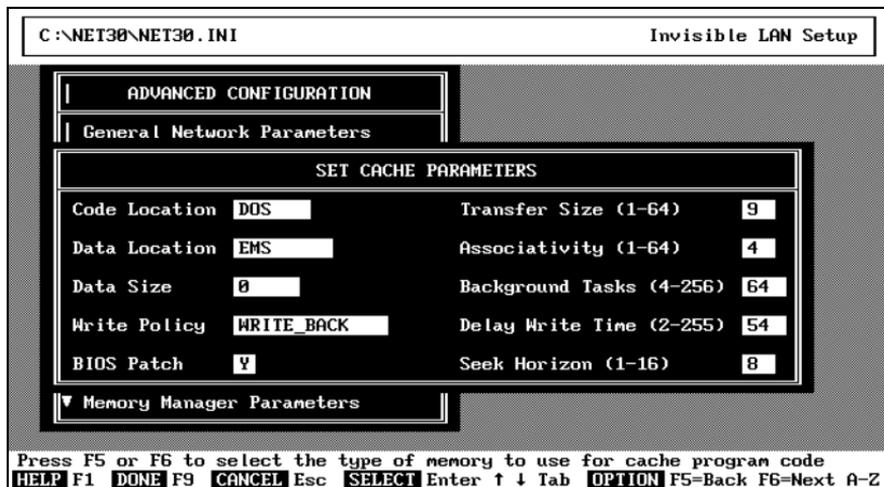


Figure 8-9. Cache Parameters Screen

Memory Allocation

The cache uses two blocks of memory: cache data and cache code. The *cache data* is where disk data is stored; it can range in size up to several megabytes. The *cache code* is the actual program code, plus the track buffer; it is typically 20K bytes in size.

With these parameters, you can control the cache's memory usage.

- **Cache Data Location** specifies the type of memory used for the disk cache data. You can select **EMS**, or **EXTENDED**, or **XMS**. To use **EMS**, you need to have expanded memory compatible with EMS version 4.0. **EXTENDED** and **XMS** both select extended memory, which requires a 286, 386, or 486 computer. If you choose **EXTENDED**, the cache accesses extended memory using the BIOS (INT 15) interface. If you choose **XMS**, you must have an XMS device driver, such as HIMEM.SYS; the cache accesses extended memory by calling the XMS device driver.

Note — If you are running Windows, you cannot use **EXTENDED** as the cache data location. You should use **XMS** instead.

- **Cache Data Size** specifies the amount of memory to use for the cache data, in kilobytes. If you enter a value of 0, or if you leave the field blank, then the cache uses all available memory. The default value of this parameter is 0.
- **Cache Code Location** specifies the type of memory to use for the cache code. You can select either **DOS** or **SHADOW**. To use shadow RAM, you need to install a memory manager that provides shadow RAM or upper memory. The default value of this parameter is **DOS**.

Fine Tuning

The following parameters let you fine-tune the operation of the cache.

- **Associativity** controls the organization of the cache data. Because the cache data may contain hundreds of disk tracks, it would take a long time to search through the entire cache to determine if a given track is in the cache or not. Therefore, the cache data is grouped into *associations*, with a small number of buffers in each association. Each track on the disk can be cached in only one of the associations. This way, any track can be located by searching just one association, instead of searching the entire cache. The value of this parameter determines the number of buffers in each association. You can enter a number from 1 to 64; the default value is 4.
- **Write Policy** determines the method that the cache uses to write data to the disk. You can select **WRITE_THROUGH**, **WRITE_BACK**, or **DELAYED_WRITE**; the default value is **WRITE_BACK**.

Write-through means that when a program performs a disk write, the data is physically written to the disk immediately. The program must wait until the physical disk write is complete.

Write-back means that when a program performs a disk write, the data is stored in the cache. The physical disk write is started as soon as possible, but then completes in background. The program does not have to wait for the physical disk write to complete. This is the default write policy.

Delayed-write means that when a program performs a disk write, the data is stored in the cache. The physical disk write does not even begin until several seconds later. The physical disk write is performed entirely in background, and the program does not have to wait for the disk write.

Write-back is much faster than write-through, since programs do not have to wait until disk writes are physically completed. Delayed-write is even

faster, but you should use delayed-write with caution; you will lose data if you turn off or reboot your computer before the delayed write is completed. **Note** — You can change the write policy while Invisible LAN is running; refer to the *DOS User Manual* for details.

- **Background Tasks** is the maximum number of disk tracks waiting to be written to the hard disk. For example, entering 8 would limit the cache to at most 8 disk tracks waiting to be written. This parameter is effective only when the write policy is write-back or delayed-write. You can enter a number from 4 to 256; the default value is 64.
- **Delay Write Time** is the amount of time that the cache waits before writing data to the disk, when the write policy is delayed-write. The time is measured in units of 1/18 second. The default value is 54 (3 seconds).
- **Seek Horizon** specifies the number of operations that the cache looks at when deciding how to reorder disk operations. When there are several disk operations waiting to be performed, the cache reorders the operations so as to minimize the amount of disk head movement. This improves disk performance by reducing the amount of time that the system must wait for the disk head to move into position.

This process is called *elevator seek* because it makes the disk head move smoothly across the disk, just as an elevator in a tall building will first go all the way to the top and then all the way to the bottom, picking up and dropping off people as it goes.

In deciding how to reorder operations, the disk cache does not check all the waiting operations. Instead, this parameter specifies the number of operations to check. You can enter a number from 1 to 16. The default value is 8.

Extended Memory Access

If the cache data is installed in extended memory, the following parameters determine the manner in which the cache program accesses extended memory. These parameters are ignored if the cache data is installed in expanded memory.

- **Transfer Size** is the maximum number of disk sectors that can be transferred to or from extended memory in one operation. Since interrupts are disabled during transfers to or from extended memory, specifying too large a value can cause the system to lose interrupts. You can enter a value from 1 to 64; the default value is 9.
- **BIOS Patch** determines the method that the cache uses to access extended memory. If you enter **N**, the cache uses the computer's built-in BIOS routines to access extended memory. If you enter **Y**, the cache

program substitutes its own BIOS routines in place of the computer's built-in routines. This is useful because, on many computers, the cache program's BIOS routines are faster than the built-in BIOS routines. Also, some 386-based computers have bugs in the built-in BIOS that may interfere with the operation of the cache. The default value of this parameter is **Y**.

SUMMARY: The Cache Parameters let you specify the amount and type of memory that the cache uses. They also let you fine-tune the cache and control the method used to access extended memory.

SHARE Parameters

The SHARE program is not actually a part of Invisible LAN. SHARE is supplied as part of DOS. However, since SHARE is used almost exclusively by networks, Invisible LAN has been designed to handle the loading of SHARE. The function of SHARE is to enforce the file sharing and record locking rules. SHARE is mandatory if you load the server program.

There are two reasons for changing the SHARE configuration. One is to control the type of memory that SHARE uses. The second reason is to control the amount of resources that SHARE makes available for the file sharing and record locking functions.

The SHARE parameters are entered on the SHARE Parameters screen, as shown in figure 8-10. Note that these parameters are effective only if SHARE is loaded by the Invisible LAN loader (**NET30.EXE**). If SHARE is loaded from the DOS command line, then the parameters must be placed on the DOS command line.

Memory Allocation

The SHARE program can use conventional DOS memory or shadow RAM. In order to use shadow RAM, you must install a memory manager that provides shadow RAM or upper memory.

- **Use Shadow RAM** specifies whether or not the SHARE should be loaded into shadow RAM. Enter **Y** if you want SHARE to be loaded into shadow RAM. Enter **N** if you want SHARE to be loaded into conventional DOS memory. The default value for this parameter is **N**.

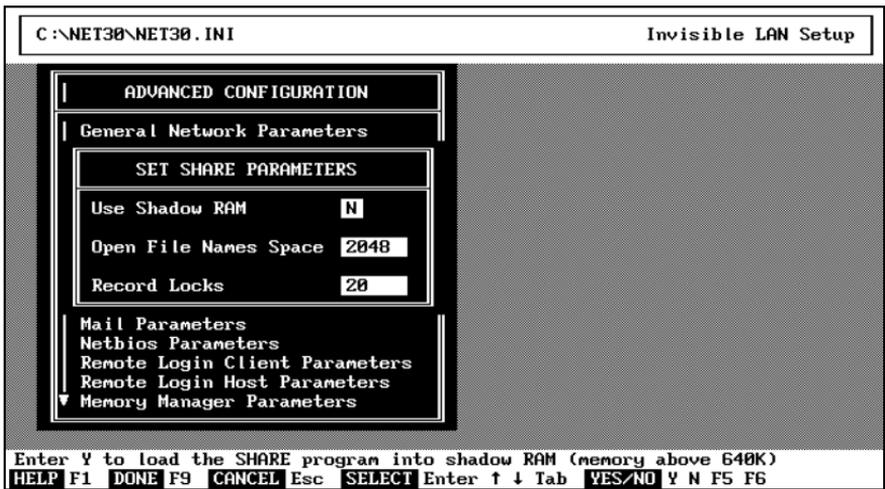


Figure 8-10. SHARE Parameters Screen

Resource Allocation

These parameters control the amount of resources that the SHARE makes available to programs. On a file server, these parameters help determine the number of network files and file locks that the server supports.

The default values of these parameters are generally adequate for supporting up to six workstations. You will want to increase these parameters on a server that supports more than six workstations. You will also want to increase these parameters if your workstations make unusually heavy demands on the server.

- **Open File Names Space** is the amount of memory, in bytes, that SHARE allocates to hold the names of all open files. The default value is 2048. This block of memory must be large enough to hold the full path names of all files that are concurrently open. Thus, on a file server, this parameter places a limit on the number of network files that the server can support (along with the server's **Files** parameter, and the FILES= statement in the server's CONFIG.SYS file). **Note** — This parameter performs the same function as the /F parameter on the SHARE command line.

Note — If you are using Ultra Server, this parameter must be large enough to accommodate all the DOS channels used by Ultra Server (see the Ultra Server Parameters section later in this chapter). Generally, a value of 6000 is adequate. For Ultra Server, unlike the standard server, this parameter does *not* limit the number of open files.

- **Record Locks** is the total number of record locks that SHARE supports. The default value is 20. Record locks are used mainly by database applications. On a file server, this parameter places a limit on the number

of network file locks that the server can support (along with the server's **Locks** parameter). **Note** — This parameter performs the same function as the **/L** parameter on the **SHARE** command line.

Note — For Ultra Server, unlike the standard server, this parameter does *not* limit the number of concurrent record locks.

SUMMARY: The **SHARE** Parameters let you specify the type of memory that **SHARE** uses, and the amount of resources that **SHARE** makes available.

Mail Parameters

Mail is the program that lets you send and receive short messages to or from other users on the network. The parameters described in this section let you configure the mail program. This lets you control the type of memory that mail uses (conventional, shadow, or EMS).

The mail parameters are entered on the Mail Parameters screen, as shown in figure 8-11.

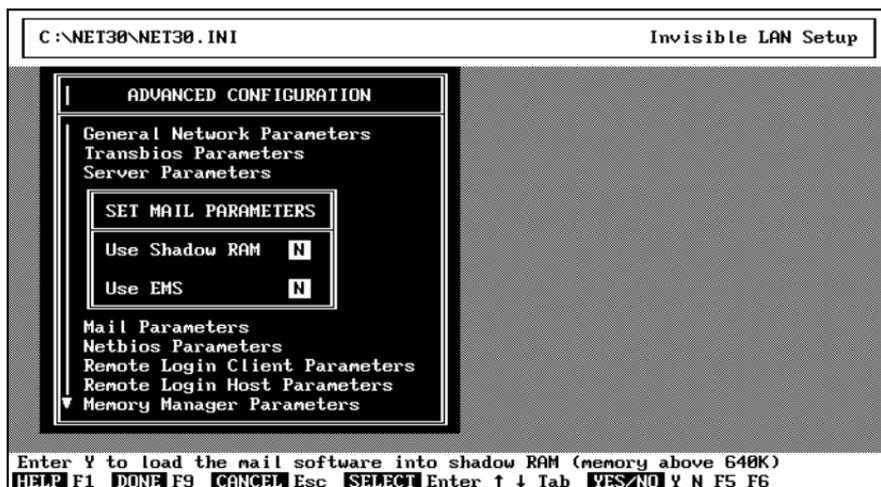


Figure 8-11. Mail Parameters Screen

Memory Allocation

The mail program can use conventional DOS memory, shadow RAM, or EMS (expanded memory). In order to use shadow RAM, you must install a memory manager that provides shadow RAM or upper memory. To use EMS, you must have expanded memory compatible with the Expanded Memory Specification version 4.0.

- **Use Shadow RAM** specifies whether or not the mail program should use shadow RAM. Enter **Y** if you want mail to load itself into shadow RAM. Enter **N** if you want mail to load itself in conventional DOS memory. The default value for this parameter is **N**.
- **Use EMS** specifies whether or not the mail program should use expanded memory. If you enter **Y**, then most of the mail software is loaded into expanded memory, and the remainder is loaded into conventional DOS memory. If you enter **N**, mail does not use expanded memory. The default value for this parameter is **N**.

Note — You can specify **Y** for both **Use Shadow RAM** and **Use EMS**. If you do, then most of the mail software is loaded into expanded memory, and the remainder is loaded into shadow RAM.

SUMMARY: The Mail Parameters let you specify the type of memory that Mail uses.

Ultra Server Parameters

Ultra Server is a special high-performance version of the file server. Ultra Server requires a 386 or 486 computer. It is intended mainly for use as a dedicated server, however you can use it in non-dedicated mode if you wish.

Ultra Server uses extended memory to maintain a cache of file data. Unlike the disk cache, which merely stores disk sectors, the Ultra Server cache stores file data, directory entries, file handles, and record locks. Also, the Ultra Server cache is multitasking, which allows multiple users to access the cache data simultaneously.

The Ultra Server parameters are entered on the Ultra Server Parameters screen, as shown in figure 8-12. These parameters control the configuration of the Ultra Server cache.

Note — When you install Ultra Server, it runs in parallel with the standard server. In other words, both Ultra Server and the standard server are running

in the computer at the same time. Ultra Server handles cached drives, while the standard server handles uncached drives, printing, security, and management. This means that you can use *both* the standard server parameters *and* the Ultra Server parameters to configure your system.

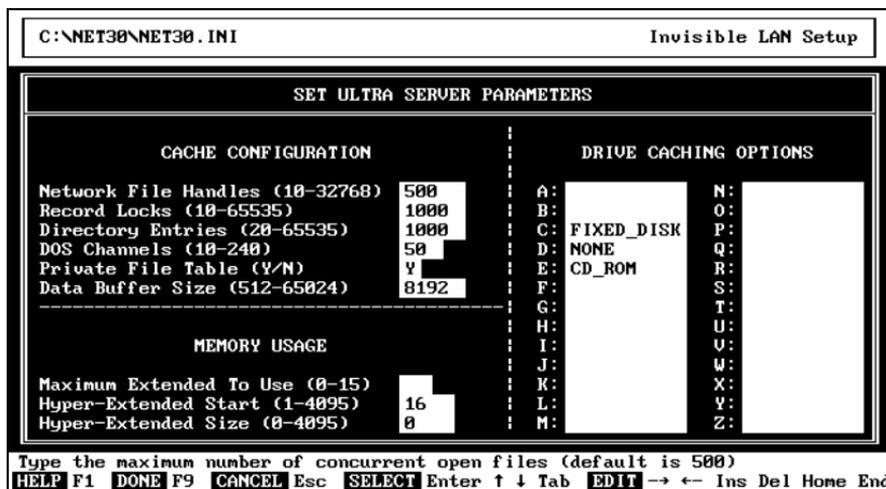


Figure 8-12. Ultra Server Parameters Screen

Cache Configuration

These parameters control the allocation of memory within the Ultra Server cache. They determine the number of network files and record locks that Ultra Server supports. Additionally, they determine the number of directory entries that can be retained in the cache, and they specify how Ultra Server uses DOS file handles. And they also set the size of the buffers used for holding file data.

The default values of these parameters are generally adequate for most installations. You only have to increase these parameters in the case of a very heavily used server.

- **Network File Handles** is the number of files that network users can open concurrently. This limits the total number of files that can be opened by all workstations using Ultra Server. You can enter a value from 10 to 32768. The default value is 500.

Note that the **FILES=** value in CONFIG.SYS, and the **Open File Names Space** parameter of the SHARE program, have no effect on the number of file handles supported by Ultra Server.

- **Record Locks** is the number of record locks that Ultra Server supports. You can enter a value from 10 to 65535. The default value is 1000. Record locks are used mainly by database applications.

Note that the **Record Locks** parameter of the SHARE program has no effect on the number of record locks supported by Ultra Server.

- **Directory Entries** is the number of directory entries that Ultra Server holds in its cache. The value can range from 20 to 65535, with a default value of 1000.
- **DOS Channels** is the number of DOS file handles that Ultra Server can use to read and write the disk. You can enter a value from 10 to 240; the default value is 50.

Whenever Ultra Server needs to read or write a disk file, it calls DOS. Since Ultra Server allows thousands of open files, while DOS only allows 255 file handles, it is not possible for Ultra Server to use a different DOS handle for each network file. Instead, Ultra Server only opens DOS file handles for the most recently used files. This parameter determines how many DOS file handles Ultra Server may keep open.

Important — The **Open File Names Space** parameter of the SHARE program must include sufficient space for all of the DOS channels. We recommend that you set **Open File Names Space** to at least 100 times the **DOS Channels** value.

- **Private File Table** specifies whether or not Ultra Server should allocate a private file table for the DOS channels. If you specify **Y**, then the **FILES=** value in CONFIG.SYS does not have to allow for the DOS channels used by Ultra Server. If you specify **N**, then the **FILES=** value in CONFIG.SYS must be big enough to accommodate all the DOS channels used by Ultra Server, in addition to any other files that are open at the server. The default value is **Y**.
- **Data Buffer Size** is the size of each data buffer in the cache, in bytes. You can enter a number from 512 to 65024; the default value is 8192.

For best results, the buffer size should be a multiple of the disk allocation unit (also called the cluster size). You can use **CHKDSK** to find out your disk's allocation unit. For CD-ROMs, the allocation unit is always 2048 bytes.

When Ultra Server reads data from the disk, it always reads a complete buffer, even if the entire buffer is not immediately needed. This "read-ahead" is very effective at speeding up CD-ROMs and other optical disks, because these types of disk are much more efficient at doing large reads than small reads. You can use this parameter to adjust the amount of "read-ahead" that Ultra Server performs.

Memory Usage

These parameters control the amount of extended and hyper-extended memory that Ultra Server uses. As a rule, you should try to give Ultra Server as much memory as possible.

Memory below the 1 megabyte boundary is called *conventional memory*. Memory between 1 megabyte and 16 megabytes is called *extended memory*. Memory above the 16 megabyte boundary is called *hyper-extended memory*. Conventional memory is the only memory that is available to DOS applications. Extended memory and hyper-extended memory are only available to programs such as Ultra Server that run in protected mode.

When Ultra Server loads, it attempts to detect how much extended and hyper-extended memory you have available, and it automatically uses all the memory it can detect (unless you specify otherwise). Ultra Server can always detect how much extended memory you have. However, if you have hyper-extended memory then Ultra Server may or may not be able to detect it.

The method Ultra Server uses to detect hyper-extended memory does not work on all computers. Also, Ultra Server can detect at most the first 49M of hyper-extended memory (that is, memory below the 65M boundary).

The parameters described in this section let you do two things. First, you can control how Ultra Server uses the memory that it is able to detect automatically. Second, if you have hyper-extended memory that Ultra Server is unable to detect automatically, you can manually tell Ultra Server to use that memory.

Determining What Memory Ultra Server Can Detect

You can use the **SHADOW** program to see how much hyper-extended memory Ultra Server will be able to detect. With no network software running, give the command

```
SHADOW /E
```

The program displays a message similar to this:

```
Extended memory size: xxxxxK
```

In the message, **xxxxx** indicates the total amount of extended and hyper-extended memory that Ultra Server can detect. The following table shows some typical values:

SHADOW Display	Amount of Hyper-Extended Memory Detected Automatically	Start of Undetected Hyper-Extended Memory
15360K	0M	16M
19456K	4M	20M
23552K	8M	24M
27648K	12M	28M
31744K	16M	32M
35840K	20M	36M
39936K	24M	40M
44032K	28M	44M
48128K	32M	48M
52224K	36M	52M
56320K	40M	56M
60416K	44M	60M
64512K	48M	64M
65535K	49M	65M

Controlling the Use of Detectable Memory

The following parameter controls the usage of all the memory that Ultra Server can detect automatically, regardless of whether it is above or below the 16M boundary. If you leave the parameter blank, then Ultra Server uses all the memory that it can detect.

- Maximum Extended To Use** is the maximum amount of detectable extended and hyper-extended memory that Ultra Server can use, in megabytes. You can enter a value from 0 to 64. If you leave the field blank, then Ultra Server uses all available extended and hyper-extended memory that it is able to detect. If you enter 0, then Ultra Server does not use any of the extended and hyper-extended memory that it is able to detect.

Example — Suppose that **SHADOW /E** returns 31744K. This means that Ultra Server can detect 15M of extended memory and 16M of hyper-extended memory, for a total of 31M. Let's say that you specify a **Maximum Extended To Use** of 24. Then Ultra Server will use 24M of the memory that it can detect, leaving 7M free. On the other hand, if you leave **Maximum Extended To Use** blank, then Ultra Server will use the entire 31M.

Getting Ultra Server to Use Undetectable Memory

The following two parameters specify a region of hyper-extended memory that Ultra Server can use. This is in addition to whatever hyper-extended memory Ultra Server can detect automatically. The region must not overlap with the hyper-extended memory that can be detected automatically. You can use these parameters to make Ultra Server use a block of hyper-extended memory that it is unable to detect automatically.

- **Hyper-Extended Start** is the starting location of undetectable hyper-extended memory, in megabytes. You can enter a number from 1 to 4095. The default value is 16, which is the starting location of hyper-extended memory.
- **Hyper-Extended Size** is the amount of undetectable hyper-extended memory you have, in megabytes. You can enter a value from 0 to 4095; the default is 0. If you enter 0, then Ultra Server only uses the memory that it is able to detect automatically.

Example — Suppose that your computer has a total of 48M RAM, which means it has 1M of DOS memory, 15M of extended memory, and 32M of hyper-extended memory. Suppose that **SHADOW /E** returns 15360K. This means that Ultra Server cannot detect any of your hyper-extended memory. In this case, you should set **Hyper-Extended Start** to 16 and **Hyper-Extended Size** to 32. Notice that the far right-hand column of the table above gives you the value to use for **Hyper-Extended Start**.

Example — Suppose that your computer has a total of 128M RAM, which means it has 1M of DOS memory, 15M of extended memory, and 112M of hyper-extended memory. Suppose that **SHADOW /E** returns 65535K. This means that Ultra Server can only detect the first 49M of hyper-extended memory. In this case, you should set **Hyper-Extended Start** to 65 and **Hyper-Extended Size** to 63. Notice again that the right-hand column of the table gives you the value to use for **Hyper-Extended Start**. The value for **Hyper-Extended Size** is the amount of hyper-extended memory you have, minus the amount that is detected automatically ($63 = 112 - 49$).

Drive Caching Options

Ultra Server maintains a cache of file data. For each drive in the system from A: to Z:, there are three possibilities: (1) the drive can be uncached; (2) the drive can be cached as a fixed disk, which allows both reading and writing; or (3) the drive can be cached as a CD-ROM, which allows only reading.

Files located on uncached drives are handled by the standard server. Ultra Server only handles files that are located on cached drives.

As a special exception to these rules, Ultra Server establishes a cache *blackout zone* in the network directory (usually C:\NET30). Files located in the network directory are never cached, even if the network directory is on a cached drive. Also, files located in subdirectories of the network directory are never cached.

When Ultra Server is started, it automatically determines which drives are present in the system, and which of those drives are CD-ROMs. By default, drives A: and B: are uncached. Each installed drive from C: to Z: is cached as either a fixed disk or CD-ROM, depending on the type of drive installed.

Using the drive caching options, you can override the default configuration of Ultra Server and specify how Ultra Server is to cache each drive. For each drive from A: to Z:, you can press **F5** and **F6** to select one of the following options:

- **BLANK** selects default caching. For A: and B:, the default is no caching. For each installed drive from C: through Z:, the default is to cache the drive as either a fixed disk or CD-ROM, depending on the type of drive that is installed.
- **NONE** tells Ultra Server not to cache the drive.
- **FIXED_DISK** tells Ultra Server to cache the drive as a standard read/write drive. Users can read from the drive and write to the drive (subject to the usual security permissions). Standard rules for file sharing and record locking are enforced.
- **CD_ROM** tells Ultra Server to cache the drive as a read-only drive. Users can read from the drive, but they cannot write to it. Rules for file sharing and record locking are suspended; all data on the disk is available to all users at all times (subject to the usual security permissions).

SUMMARY: The Ultra Server Parameters let you specify the configuration of the Ultra Server cache, the amount of extended and hyper-extended memory that Ultra Server uses, and the caching options for each drive.

Guide to Configuring Invisible LAN

Invisible LAN has many parameters that you can use to customize its configuration. If you are new to Invisible LAN, you may need assistance to determine which parameters you ought to change. In this chapter, we provide that assistance by explaining what parameters should be changed to achieve various goals.

The topics covered are:

- Optimizing performance
- Minimizing memory usage
- Large networks

Optimizing Performance

When properly configured, Invisible LAN can deliver outstanding performance. In this section, we give a number of suggestions that should help you make the network run at the fastest possible speed.

The five most important things you can do to improve network performance are:

- Install Ultra Server on the server. Ultra Server is optimized to give the fastest possible server performance. It uses fast 32-bit code, true multi-tasking, and file-level caching.
- Install the disk cache on the server. A server with a disk cache is much faster than one without. For fastest performance, use the delayed-write write policy; if you're not comfortable with delayed-write, then use the write-back write policy. (You can use both the disk cache and Ultra Server on the same computer.)

- Don't use the TransBIOS/NetBIOS ROM on the network card. You get much better performance by disabling the ROM and using the RAM version of TransBIOS/NetBIOS. (Not all network cards have a TransBIOS/NetBIOS ROM.)
- Don't use EMS (expanded memory) for the network software; use DOS memory or shadow RAM instead. The network software runs slower in EMS because of the extra overhead of page mapping. Using shadow RAM does not affect the speed of the network.
- Use a fast computer for the server, preferably a 386 or 486. This may seem obvious, but a 386 or 486 server gives much better performance than a 286 or 386SX server. The network software uses fast 32-bit instructions when installed on a 386 or 486. An 8088-based computer should not be seriously considered for use as a server, due to its slow speed.

The following steps also improve the performance of the network, although not as much as the four steps listed above.

- Provide as much memory as possible to Ultra Server. If your computer has hyper-extended memory (memory above 16 megabytes), configure Ultra Server to use the hyper-extended memory.
- Allocate as much memory as possible to the disk cache. You should try to allocate at least 2 megabytes for the cache data. However, if you don't have this much memory, use whatever you have; even a small cache is better than no cache at all. (If you are using both the disk cache and Ultra Server, we recommend allocating one-fourth of your extended memory to the disk cache, and three-fourths to Ultra Server.)
- If you have a choice of using either extended or expanded memory for the disk cache data, use extended. Expanded memory is slower than extended memory because of the extra overhead of mapping pages. If you have a choice of **XMS** or **EXTENDED** (both of which refer to extended memory), choose **EXTENDED**.
- On a heavily-used server, you can increase performance by increasing the size of the TransBIOS data segment (on the TransBIOS Parameters screen).
- On a server, you can increase performance by increasing the number of network buffers (on the Server Parameters screen). If you can spare the memory, allocate four buffers for each workstation. If you're low on memory, try to allocate at least one buffer for each workstation.
- On a server, use the **DEDICATE** command to set up a dedicated server. (Refer to the *DOS User Manual* for information about the **DEDICATE** command.)

- The print spooler can be speeded up by placing the spool directory in a RAM disk (on the Print Spooler Parameters screen).
- If you print graphics or page description languages (such as PostScript), you can speed up the print spooler by increasing the size of the spool buffer (on the Print Spooler Parameters screen).
- The server runs faster when the write pipeline is enabled (on the Server Parameters screen).
- 386 control programs (such as Invisible RAM 386, Microsoft's EMM386, Quarterdeck's QEMM-386, or Qualitas' 386-MAX) slow down the system, because the microprocessor runs slower in virtual mode. Don't use a 386 control program if you don't have to.
- Some memory managers have the ability to replace BIOS ROMs with faster RAM, thereby speeding up the computer system. With Invisible RAM, use the System Memory Configuration screen to enable this feature; with other memory managers, refer to the memory manager's instruction manual.
- On a redirector, allocate buffers for local caching of network files, and make them as big as possible. Also, make the print buffer size as big as possible (on the Redirector Parameters screen).

SUMMARY: The most important things you can do to improve network performance are: install Ultra Server; install the disk cache; don't use the TransBIOS/NetBIOS ROM; don't put the network software in EMS; and use fast 386 or 486 computer for the server.

Minimizing Memory Usage

A problem that many network users experience is that after loading the network software, there isn't enough memory left to run their applications. Invisible LAN comes with outstanding memory management capabilities, that let you free up memory for DOS applications. In this section, we give suggestions for freeing up as much memory as possible.

The nine things you can do that are most effective in freeing up memory are:

- Install a memory manager — either one of the Invisible RAM memory managers, or a third-party memory manager such as Microsoft's EMM386, Quarterdeck's QEMM-386, or Qualitas' 386-MAX. The memory manager actually increases the size of DOS memory, to 736K on a color system, or 704K on a monochrome system. It also provides *shadow*

RAM or *upper memory* that can be used to load memory-resident programs outside DOS memory (shadow RAM works with both the network software and other memory-resident programs).

- Use shadow RAM for the network software. This is the best way to get the network software out of DOS memory, since the entire network software can be placed into shadow RAM, and there is little or no degradation in performance.
- Use EMS (expanded memory) for the network software. This is not as good as using shadow RAM, since a portion of the network software remains in DOS memory, and there is a reduction in performance.
- If you have shadow RAM available, but not enough to hold all of the network software, you can use *both* EMS and shadow RAM. Then most of the software goes into EMS, and the remainder into shadow RAM.
- Use shadow RAM for third-party memory resident programs and device drivers. This frees up more DOS memory.
- If you are using the disk cache, reduce your DOS BUFFERS (in CONFIG.SYS) to the minimum value of 4. With a cache installed, it is not necessary to have a lot of DOS BUFFERS.
- If you are using the **386_VIRTUAL** memory manager (Invisible RAM 386), put STACKS=0,0 in your CONFIG.SYS file.
- If you aren't using any memory managers, enable the TransBIOS/NetBIOS ROM on the network board. This frees up the DOS memory that would otherwise be used to hold the TransBIOS program code. (Not all boards have a TransBIOS/NetBIOS ROM.)
- Some memory managers have the ability to replace unnecessary ROMs with shadow RAM (for example, many computers have a 32K ROM that contains the computer's setup program). These memory managers are also able to replace unnecessary video RAM with shadow RAM (many video adapters have 16K of video RAM that is never used). This creates a substantial amount of additional shadow RAM, that can be used for the network software or other memory-resident programs.

Here are some additional suggestions for reducing the network's memory requirements. They are not as effective as the previous nine suggestions.

- Make the TransBIOS data segment as small as possible, by entering 0 for the data segment size (on the TransBIOS Parameters screen).
- Reduce the number of TransBIOS sessions (on the TransBIOS Parameters screen) to the minimum you actually need. Here is how to calculate what you need: If you load the server, you need one session for each workstation that uses the server. If you load the redirector, you need one

session for each server that the redirector uses. If you load the Remote Program Load Server, you need one session for each diskless workstation. Finally, allow two “extra” sessions. (If any of your applications open NetBIOS sessions, you need to allow for their sessions as well; remember that each NetBIOS session requires one TransBIOS session.)

- Reduce the number of TransBIOS names (on the TransBIOS Parameters screen) to the minimum you actually need. TransBIOS uses 2 names internally, and Invisible LAN never uses more than 4 names; so a value of 6 is always enough to run Invisible LAN. (If any of your applications create NetBIOS names, you need to allow for their names as well; remember that each NetBIOS name requires one TransBIOS name.)
- Reduce the number of NetBIOS names, commands, and sessions (on the NetBIOS Parameters screen) to the minimum you actually need. Invisible LAN never uses more than 2 NetBIOS names or 3 NetBIOS commands. If you don’t load the Remote Program Load Server, then Invisible LAN doesn’t use any NetBIOS sessions at all; if you load the Remote Program Load Server, you need to allow one NetBIOS session for each diskless workstation. (If any of your applications use NetBIOS, you need to allow whatever number of names, commands, and sessions they require.)
- Reduce the number of network buffers that the server allocates (on the Server Parameters screen). Each buffer uses 2K bytes of RAM. Reducing the number of buffers does not affect the number of workstations or files that the server can support; however, it may reduce the server’s speed.
- Reduce the number of server sessions (on the Server Parameters screen) to the minimum you actually need. You need one session for each workstation that uses the server, plus two “extra” sessions.
- Reduce the number of files and record locks supported by the server and SHARE programs (on the Server and SHARE Parameter screens). The minimum you need depends on the application programs you run.
- Reduce the size of the print queue (on the Print Spooler Parameters screen) to the minimum you actually use. Also, you can try reducing the size of the print spool buffer (also on the Print Spooler Parameters screen).
- Reduce the number of redirector sessions (on the Redirector Parameters screen) to the minimum you actually need. You need one session for each server you use, plus two “extra” sessions.
- Reduce the number of files and FCBs that the redirector supports (on the Redirector Parameters screen). The minimum you need depends on the application programs you run.
- Don’t allocate any redirector buffers for caching network files, and reduce the redirector’s print buffers to the minimum size (on the Redirector Parameters screen).

SUMMARY: The most important things you can do to minimize memory usage are: install a memory manager; use shadow RAM and/or EMS for the network software; use shadow RAM for memory-resident programs; minimize DOS BUFFERS and STACKS; use the TransBIOS/NetBIOS ROM; and disable unnecessary ROMs.

Large Networks

The default parameter values are generally adequate for networks with six stations or less. If you have more than six network stations, you may need to increase some parameter values. The following guidelines should help you determine what parameters you need to change if you have a large number of stations.

- If a server needs to support more than 6 workstations, increase the number of network sessions that the server can handle (on the Server Parameters screen). Allow one session for each workstation, plus 2 “extra” sessions. If there are more than 25 workstations, then you also need to increase the number of network sessions that TransBIOS can handle (on the TransBIOS parameters screen).
- If a server is supporting many workstations, consider increasing the number of network buffers that the server allocates (on the Server Parameters screen). This is not actually necessary to make the server work; but it helps maintain high performance as you add workstations.
- If a server is supporting many workstations, you may need to increase the number of files that the server can support. This has to be done in three places: the files parameter in the Server Parameters screen; the filespace parameter in the SHARE Parameters screen; and the FILES= statement in the server’s CONFIG.SYS file. If you need more than 255 files, you need to configure the server to create a private file table.

For Ultra Server, if you want to increase the number of files, you only need to change the files parameter in the Ultra Server parameters screen. The number of files that Ultra Server supports is not affected by the SHARE filespace parameter or the FILES= value in CONFIG.SYS.

- On a server that handles a lot of database operations, you may need to increase the number of record locks that the server can support. This has to be done in two places: the record locks parameter in the Server Parameters screen, and the record locks parameter in the SHARE Parameters screen.

For Ultra Server, if you want to increase the number of record locks, you only need to change the record locks parameter in the Ultra Server parameters screen. The number of record locks that Ultra Server supports is not affected by the SHARE record locks parameter.

- If a redirector needs to be connected to more than 6 servers simultaneously, increase the number of network sessions that the redirector can handle (on the Redirector Parameters screen). Allow one session for each server, plus 2 “extra” sessions.
- If your application software uses an unusually large number of network files, you may need to increase the number of files that the redirector can support (on the Redirector Parameters screen).
- On a large network, you should consider using a dedicated server. A dedicated server can devote all its time to servicing the workstations, since it does not need to devote any time to a local user. With a dedicated server you can allocate lots of resources, since you don’t have to conserve memory for a local user. Also, a dedicated server can make good economic sense on a large network, because the cost of the server is spread over many users.

SUMMARY: For networks with more than six stations, you will probably have to increase the number of resources provided by the server and redirector.

Advanced Windows Driver Installation

Chapter 5 described the basic procedures for installing Invisible LAN For Windows. This chapter gives more detail about installation, and fully describes all the available options.

The topics covered are:

- Components of Invisible LAN
- Configuring Invisible LAN
- Configuring Windows
- Installing the Invisible LAN icons
- Special requirement for serial network printers
- Special requirement for WINSTART.BAT
- Special requirement for TransBIOS ROM
- Installing Invisible LAN wallpaper
- Selecting printer drivers

Components of Invisible LAN

Invisible LAN is actually two programs in one: Invisible LAN For DOS, and Invisible LAN For Windows.

Invisible LAN For DOS is a network operating system for DOS-based computers. It allows DOS computers to communicate over a network, sharing disks, printers, and other resources.

Invisible LAN For Windows is a network operating system for both DOS-based and Windows-based computers. It allows both DOS and Windows computers to communicate over a network, sharing disks, printers, and other resources.

When you configure Invisible LAN, you select which of the two versions of

the operating system you want to use.

From the user's standpoint, Invisible LAN For Windows is an extension of Invisible LAN For DOS. Just as Windows extends DOS to provide a multi-tasking graphical interface, Invisible LAN For Windows extends Invisible LAN For DOS to provide support for graphics and multitasking.

Invisible LAN For Windows is fully compatible with Invisible LAN For DOS. Both operating systems can exist on the network at the same time, and they can communicate with each other to share data and resources.

Invisible LAN For DOS

Invisible LAN For DOS has six memory-resident components. *Memory-resident* means that the components remain in your computer's memory while you run other programs. The six memory-resident DOS components are:

- **1. Redirector** — Lets you use disks and printers that are attached to other computers on the network.
- **2. Server** — Allows other computers on the network to use your disks and printers.
- **3. Disk Cache** — Speeds up hard disks by storing disk data in the computer's memory.
- **4. Mail** — Sends and receives messages to other users on the network.
- **5. Memory Manager** — Uses your computer's memory-management hardware to provide more memory for DOS applications.
- **6. TransBIOS and NetBIOS** — Controls the network hardware and allows programs to communicate on the network.

You can use these components in any combination you want. When you configure Invisible LAN For DOS, you select which components you want to use. The selected components are loaded into memory automatically when you start Invisible LAN.

Invisible LAN For Windows

Invisible LAN For Windows also uses the six memory-resident components listed above, although most of them are replaced with special Windows-compatible versions. The redirector, server, disk cache, and mail programs all have special versions for Windows. When you configure Invisible LAN to support Windows, these special Windows versions are automatically loaded in place of the standard DOS versions.

In addition, Invisible LAN For Windows adds a seventh memory-resident component:

- **7. Windows Network Driver** — A program that links Windows to Invisible LAN. The network driver lets Windows communicate with the network software and access network functions. The network driver also contains the Windows Network Manager, a graphical program that you use to manage the network.

Unlike the other memory-resident components, the Windows network driver does *not* load when you start Invisible LAN. The network driver loads when you start Windows, and remains in memory as long as Windows is running. When you exit from Windows, the network driver automatically unloads from memory.

Since the network driver is only loaded while Windows is running, the network driver does not take up any DOS memory.

SUMMARY: Invisible LAN is two operating systems in one. *Invisible LAN For DOS* works on DOS-based computers, while *Invisible LAN For Windows* works on both DOS-based and Windows-based computers. When you configure Invisible LAN, you select which operating system you want to use. Invisible LAN For DOS includes six memory-resident components: redirector, server, disk cache, mail, memory manager, and Trans-BIOS/NetBIOS. Invisible LAN For Windows adds a seventh component: the Windows network driver. The network driver enables Windows to communicate with Invisible LAN, providing full compatibility between Windows and the network. The network driver also contains the Windows Network Manager.

Configuring Invisible LAN

When installing Invisible LAN, you specify which components you want to use. You also specify whether or not to load the Windows version of Invisible LAN.

You use the **SETUP30** program to configure Invisible LAN. Your configuration is stored on disk in the Invisible LAN initialization file, which is usually called NET30.INI. The following paragraphs explain how to configure Invisible LAN properly so that it can run with Windows.

Enabling Windows Support

If you want to run Windows, you have to enable the Windows support in Invisible LAN. This is done with the **SETUP30** program. There are two places

in **SETUP30** where you can enable Windows support:

- In the **Easy Configuration** section of **SETUP30**.
- In the **Advanced Configuration** section of **SETUP30**, under **General Network Parameters**.

The **Windows Support** option selects which version of Invisible LAN you want to use. Enter **Y** for **Windows Support** to select Invisible LAN For Windows. Enter **N** for **Windows Support** to select Invisible LAN For DOS.

If you enter **N**, then you cannot run Windows; you can only run DOS. If you enter **Y**, then you can run both DOS and Windows. Be aware that entering **Y** increases the amount of memory used by Invisible LAN, because of the extra program code required to support Windows.

It is not necessary to enable Windows support on all your network computers. You only have to enable Windows support on the computers that run Windows.

Disk Cache Configuration

When you configure the Invisible LAN disk cache, you have to specify the location of the cache data. There are three possible locations: **EMS**, **EXTENDED**, and **XMS**.

(**EMS** refers to expanded memory. **EXTENDED** and **XMS** both refer to extended memory. **EXTENDED** and **XMS** differ in the method that the cache uses to access the extended memory. **EXTENDED** specifies that the cache should access extended memory via the computer's BIOS. **XMS** specifies that the cache should access extended memory via an XMS device driver, such as Microsoft's HIMEM.SYS.)

When you install Invisible LAN For DOS, you can use any of the three possible locations.

When you install Invisible LAN For Windows, you can only use **EMS** or **XMS**. You cannot use **EXTENDED**. This is not a problem, because you must install Microsoft's HIMEM.SYS in order to run Windows, and HIMEM.SYS provides XMS memory that you can use for the cache.

When you install the cache in XMS memory, make sure that you leave enough XMS free to run Windows. By default, the cache leaves 1 megabyte of XMS free, which is the bare minimum required to run Windows in 386 enhanced mode. You will probably want to reduce the cache size, to leave more XMS memory for Windows.

Shadow RAM and Expanded Memory Utilization

You can load some or all of Invisible LAN into shadow RAM (also called upper memory) and/or expanded memory (also called EMS).

Invisible LAN For Windows offers the same memory utilization features as Invisible LAN For DOS. When configuring Invisible LAN for Windows, you can load some or all of the network software into shadow RAM and/or expanded memory.

The only restriction is that the Windows version of the redirector cannot load into expanded memory. (It *can* load into shadow RAM.) If you try to load the Windows redirector into expanded memory, it automatically uses DOS memory instead.

Note — To run Windows in 386 enhanced mode, you need to have some free shadow RAM (also called *upper memory*). Invisible LAN can squeeze itself into shadow RAM so well, that there may be none left for Windows. In this case, you must either configure the memory manager to create more shadow RAM, or else move some of the Invisible LAN modules into conventional DOS memory.

Invisible RAM Memory Managers

Invisible RAM is memory-management software that increases the memory available to DOS programs. Depending on your hardware, Invisible RAM may be able to frontfill (expand DOS memory beyond 640K), create shadow RAM (extra memory located between 640K and 1M), and/or create expanded memory. Invisible LAN comes with seven versions of Invisible RAM, each designed for different memory hardware.

Invisible RAM is intended to be used with DOS. Invisible RAM is not a Windows memory manager. In most cases, you should not use Invisible RAM if you are going to run Windows. However, you *can* use Invisible RAM with Windows subject to the following restrictions:

- You cannot use Invisible RAM if you run Windows in 386 enhanced mode.
- If you run Windows in standard mode, you cannot use Invisible RAM 386 (the version of Invisible RAM that uses the 386 virtual mode). The other six versions of Invisible RAM can be used.
- You cannot increase DOS memory beyond 640K while Windows is running — not even within a DOS session. You must configure Invisible RAM to disable frontfill, or else run **VGAON** before starting Windows.
- You can load memory-resident programs into shadow RAM before starting Windows. You cannot load programs into shadow RAM after starting Windows.

If you have a 386 or 486 computer, and you need to use a memory manager, you should use Microsoft's EMM386 (or another Windows memory manager). Using EMM386 together with MS-DOS 5.0 provides most of the capabilities of Invisible RAM, and is compatible with all modes of Windows.

Server Configuration

If you load the file server and you allocate a private file table, or if you load Ultra Server, then you cannot run Windows in 386 enhanced mode. You *can* run Windows in standard mode.

The default server configuration does *not* allocate a private file table. Therefore, this restriction will only arise if you have taken action to explicitly allocate a private file table or select Ultra Server. (Private file tables are used to increase the number of file handles beyond the DOS limit of 255.)

SUMMARY: When configuring Invisible LAN, you have to enable the **Windows Support** option if you want to run Windows. This is done in the **SETUP30** program. In general, Invisible LAN For Windows has the same configuration options as Invisible LAN For DOS. There are, however, a few restrictions: (1) If you load the disk cache, the cache data location must be **XMS** or **EMS**. (2) The Windows version of the redirector cannot load into expanded memory. (3) Invisible RAM cannot be used with Windows 386 enhanced mode. (4) Except for Invisible RAM 386, it is possible to use Invisible RAM with Windows standard mode, subject to certain limitations. (5) If you load the server and allocate a private file table, or if you load Ultra Server, then you cannot run Windows in 386 enhanced mode.

Configuring Windows

When you install Windows, you configure it for your system. As part of the configuration process, you specify which network you are using, and install the Windows network driver.

You use the Windows Setup program to configure Windows. Your configuration is stored on disk, in two files named WIN.INI and SYSTEM.INI.

There are two ways to install Windows on a network: *stand-alone* and *shared*.

A stand-alone installation is just like a normal, non-network installation of Windows. In a stand-alone installation, you copy the Windows program files onto your own hard disk. This gives you a private copy of Windows for your own exclusive use.

In a shared installation, you copy the Windows program files onto a file server's disk. Several users can share this copy of Windows. With a shared installation, it is not necessary for each user to have a private copy of Windows.

Shared installations generate a lot of network traffic. Therefore, we recommend using stand-alone installations whenever possible.

It is not necessary to install Windows the same way on every network computer. Some computers can run stand-alone installations, while others run shared installations.

Important — The procedures described in this section are for Windows versions 3.0, 3.1, and 3.11. They are not for Windows For Workgroups. If you are using Windows For Workgroups, refer to README.TXT or WORKGRP.TXT for instructions.

Stand-Alone Windows Installation

A stand-alone Windows installation is performed in two steps. First, you install Windows on your hard disk and configure Windows for your system. Second, you install the Windows network driver.

The stand-alone installation can be performed either with or without the network running. We recommend doing it without the network running.

Note — In Windows version 3.0, there is a bug in the Windows Setup program that makes it impossible to install the Windows network driver at the same time you install the rest of Windows. You must install the network driver separately from the rest of Windows, as described here. In Windows version 3.1, it is possible to install the network driver at the same time you install the rest of Windows.

Installing Windows

To install Windows, you need to copy the Windows program files onto your hard disk, configure Windows for your computer, and modify your CONFIG.SYS file. All these tasks are performed with the Windows Setup program.

Follow the instructions in your Microsoft Windows User's Guide for running the Windows Setup program.

The Windows Setup program may ask you what type of network you are using. At this time, you should select "No Network."

Installing the Invisible LAN Windows Driver (Quick Method)

After installing Windows, you are ready to install the Invisible LAN Windows network driver. The *network driver* is the program that links Windows and Invisible LAN together. With the network driver installed, you have the full functionality of Invisible LAN For Windows.

Installing the network driver involves copying several files from your Invisible LAN directory to your Windows directory, and modifying your WIN.INI and SYSTEM.INI files. It also creates an Invisible LAN group in Program Manager. The Invisible LAN group contains icons that you can use to access network functions while running Windows.

There are three ways to install the network driver. You can use the Invisible LAN Quick Setup program, or you can use the Windows Setup program, or you can install the driver manually. The following steps describe how to install the driver using the Invisible LAN Quick Setup program. The next two sections describe the other methods.

The following instructions assume that your DOS PATH includes the directory where Invisible LAN is installed (usually C:\NET30).

- **Step 1.** Select **Run** from the Program Manager's **File** menu.
- **Step 2.** Type "QSETUPW" and choose **OK**.
- **Step 3.** A dialog box appears, asking if you want to install the Windows network driver for Invisible LAN. Choose **OK** to install the driver.
- **Step 4.** Wait while QSETUPW installs the driver and creates the Invisible LAN group. This will take approximately 20 seconds.
- **Step 5.** When the installation is complete, QSETUPW displays a dialog box informing you that you have to restart Windows for the changes to take effect. Choose **Yes** to restart Windows.

Installing the Invisible LAN Windows Driver (Automatic Method)

After installing Windows, you must install the Invisible LAN Windows network driver. The previous section described how to use the Invisible LAN Quick Setup program to install the driver.

It is also possible to install the network driver using the Windows Setup program, instead of using the Invisible LAN Quick Setup program. To install using the Windows Setup program, proceed as follows:

- **Step 1.** Make sure that Windows is not running. If Windows is running, exit Windows and return to DOS.

- Step 2. Change to the directory where Windows is installed. For example, if Windows is installed in C:\WINDOWS, you would enter the following commands:

```
C:  
CD WINDOWS
```

- Step 3. Start the Windows Setup program by typing:

```
SETUP
```

- Step 4. The Windows Setup program displays a list of the equipment in your computer. Select **Network** from the list and press **Enter**.
- Step 5. The program displays a list of networks. Select **Other** and press **Enter**.
- Step 6. The program asks you to insert a disk containing the network driver. Type the name of the directory where you have installed Invisible LAN (usually C:\NET30) and press **Enter**. You do not have to insert a disk.

Note — Setup requires the following files that are included with Invisible LAN: OEMSETUP.INF, ISNET.DRV, ISNET.HLP, and NET30APP.EXE. Make sure your Invisible LAN directory contains these files.

- Step 7. The program displays another list of networks. Select **Invisible LAN (for Windows 3.0, 3.1, and 3.11)** and press **Enter**.
- Step 8. The program may ask you to insert the Invisible LAN Driver Disk. Type the name of the directory where you have installed Invisible LAN (usually C:\NET30) and press **Enter**. You do not have to insert a disk.
- Step 9. Select **Accept Configuration**, and follow the instructions on-screen to complete the setup.

The following instructions create the Invisible LAN group in Program Manager. They assume that your DOS PATH includes the directory where Invisible LAN is installed (usually C:\NET30).

- Step 10. Start Windows and go to Program Manager. Select **Run** from the Program Manager's **File** menu.
- Step 11. Type "SETUPW" and choose **OK**.
- Step 12. A dialog box appears, asking if you want to create a Program Manager group for Invisible LAN. Choose **OK** to create the group and install the icons.

Installing the Invisible LAN Windows Driver (Manual Method)

After installing Windows, you must install the Invisible LAN Windows network driver. The previous two sections described how to use the Invisible LAN Quick Setup program or the Windows Setup program to install the driver.

It is also possible to install the network driver manually, instead of using one of the Setup programs. To install the driver manually, proceed as follows:

- Step 1. Copy the files listed below into your WINDOWS or WINDOWS\SYSTEM directory (you can use either directory):

ISNET.DRV
ISNET.HLP
NET30APP.EXE

- Step 2. Add or change the following lines in your SYSTEM.INI file:

[boot]
network.driv = isnet.driv

[boot.description]
network.driv = Invisible LAN (for Windows 3.0, 3.1, and 3.11)

[386enh]
network = *vnetbios, *dosnet
FileSysChange = off
InDOSPolling = TRUE
INT28Critical = TRUE
ReflectDosInt2A = FALSE
UniqueDOSPSP = FALSE
LPT1AutoAssign = 0
LPT2AutoAssign = 0
LPT3AutoAssign = 0
PerVMFILES = 0
TimerCriticalSection = 1000

- Step 3. Add or change the following lines in your WIN.INI file:

```
[windows]
Spooler = yes
```

```
[spooler]
netupdate = yes
netspool = yes
```

If you don't know how to modify your WIN.INI and SYSTEM.INI files, refer to the documentation that comes with Microsoft Windows.

The following instructions create the Invisible LAN group in Program Manager. They assume that your DOS PATH includes the directory where Invisible LAN is installed (usually C:\NET30).

- Step 4. Start Windows and go to Program Manager. Select **Run** from the Program Manager's **File** menu.
- Step 5. Type "SETUPW" and choose **OK**.
- Step 6. A dialog box appears, asking if you want to create a Program Manager group for Invisible LAN. Choose **OK** to create the group and install the icons.

Shared Windows Installation

In a shared Windows installation, you copy all the Windows program files onto a file server's disk, so that several users can access them. This eliminates the need for each user to have his or her own copy of Windows. This also makes it possible to run Windows on floppy-based computers and diskless workstations.

A shared Windows installation is accomplished in two steps. First, you copy all the Windows files to a directory on the file server's disk. Second, you set up Windows on each workstation that accesses the shared copy.

Important — On a shared Windows installation, it is necessary for the Windows swap file to be on a local drive. During Windows Setup, you can specify the location of the Windows swap file. If you need to change the location of the Windows swap file, click on the Control Panel icon in Program Manager, then click on the 386 Enhanced icon, and then choose Virtual Memory.

Copying Windows to the File Server

Follow these steps to copy the Windows files to the file server:

- Step 1. Create a directory on the server's disk. This directory will be used to hold the shared copy of Windows.
- Step 2. Copy all the files from your Windows distribution diskettes to the directory on the server's disk.

You also have to expand the files, since Microsoft supplies the files in compressed format. A program called EXPAND.EXE is included with Windows. You use EXPAND.EXE to expand the files as you copy them. Refer to the Microsoft Windows User's Guide for instructions on how to expand the Windows program files.

If you are using Windows 3.1, you can copy and expand the Windows files with the command **SETUP /A**. Refer to the Microsoft Windows User's Guide for instructions on using **SETUP /A**.

- Step 3. Copy the files listed below from your Invisible LAN directory to the directory that contains the shared copy of Windows. (These files are not compressed, so no expansion is necessary.)

DRVMAP.EXE
ICW.HLP
ICW.EXE
ICWLB.DLL
ISNET.DRV
ISNET.HLP
NET30APP.EXE
PRTMAP.EXE
PSDEMON.EXE
QSETUPW.EXE
SETUPW.EXE
WALL30.BMP

- Step 4. Make all the files in the shared copy of Windows read-only. This can be done with the following DOS command:

```
ATTRIB +R *.*
```

Setting Up Windows on a Workstation

After installing the shared copy of Windows on the file server, you are ready to set up Windows on the workstations.

Each workstation must have a private disk directory. The private directory can be on the file server, or it can be on the workstation's local disk. The private directory holds Windows files that contain information relevant to the particular workstation. For example, the private directory contains the workstation's WIN.INI and SYSTEM.INI files.

Follow this procedure to install Windows on a workstation:

- Step 1. Start the network and establish a drive mapping that gives you access to the shared copy of Windows.
- Step 2. Change to the directory where the shared copy of Windows is located.
- Step 3. Start the Windows Setup program with the following command:

SETUP /N

- Step 4. If you are running Windows version 3.1, the Setup program gives you the option of *Express Setup* or *Custom Setup*. Select **Custom Setup**.
- Step 5. The Setup program displays a list of the equipment in the workstation. Select **Network** from the list and press **Enter**.
- Step 6. The program displays a list of networks. Select **Other** and press **Enter**.
- Step 7. The program asks you to insert a disk containing the network driver. Type the name of the directory where you have installed Invisible LAN and press **Enter**. You do not have to insert a disk.

Note — Setup requires the file OEMSETUP.INF that is included with Invisible LAN. Make sure your Invisible LAN directory contains this file.

- Step 8. The program displays another list of networks. Select **Invisible LAN** and press **Enter**.
- Step 9. Select **Accept Configuration**, and follow the instructions on-screen to complete the setup.

After completing the setup, change the workstation's DOS PATH to include both the private Windows directory and the shared Windows directory, in that order.

Then use the following instructions create the Invisible LAN group in Program Manager.

- Step 10. Start Windows and go to Program Manager. Select **Run** from the Program Manager's **File** menu.
- Step 11. Type "SETUPW" and choose **OK**.
- Step 12. A dialog box appears, asking if you want to create a Program Manager group for Invisible LAN. Choose **OK** to create the group and install the icons.

SUMMARY: There are two ways to install Windows: stand-alone and shared. In a stand-alone installation, you have a copy of the Windows program files on your own hard disk. To perform a stand-alone installation, you first perform a normal non-network installation of Windows, and then install the Invisible LAN Windows network driver. The network driver can be installed either by using the Invisible LAN Quick Setup program, or by using the Windows Setup program, or by manually copying some files and modifying the WIN.INI and SYSTEM.INI files.

In a shared installation, a copy of Windows is installed on a file server so that several users can share it. To install Windows on the server, you copy and expand all the Windows files onto the server's disk, and then copy some of the Invisible LAN files. Then you use the Windows Setup program with the /N parameter to install Windows on each workstation that needs to use the shared copy of Windows.

Special Requirement for Serial Network Printers

If you are going to run the server, and you plan to share COM1 as a network printer, you must manually add the following line to your SYSTEM.INI file:

```
[386enh]
COM1AutoAssign = 0
```

If you are going to run the server, and you plan to share COM2 as a network printer, you must manually add the following line to your SYSTEM.INI file:

```
[386enh]
COM2AutoAssign = 0
```

If you don't know how to modify your SYSTEM.INI file, refer to the documentation that comes with Microsoft Windows.

SUMMARY: If you install the server and you want to share a serial printer, you must place a line into the SYSTEM.INI file as shown above.

Special Requirement for WINSTART.BAT

WINSTART.BAT is a batch file that Windows executes automatically when it starts. If you have a WINSTART.BAT file, you must place the following command into the WINSTART.BAT file:

WINSTRT

You can place the WINSTRT command anywhere in the WINSTART.BAT file. This is necessary to allow the network redirector to process WINSTART.BAT correctly.

SUMMARY: If you have a WINSTART.BAT file, you must insert the **WINSTRT** command into WINSTART.BAT.

Special Requirement for TransBIOS ROM

If you are using the TransBIOS/NetBIOS ROM on your network board, you must install a special program called RIPLMEM.EXE. RIPLMEM.EXE is a program written by Microsoft that allows Windows to run on diskless workstations. Although RIPLMEM.EXE is designed for diskless workstations, it is actually required whenever you use the TransBIOS/NetBIOS ROM, regardless of whether or not you actually boot from the network.

There is a copy of RIPLMEM.EXE on your Invisible LAN distribution disks, in a subdirectory called RIPLMEM. You will also find a text file called RIPLMEM.TXT which contains instructions for installing RIPLMEM.EXE.

Note — Since diskless workstations always use the TransBIOS/NetBIOS ROM, you must always install RIPLMEM.EXE if you are running Windows on a diskless workstation.

SUMMARY: If you are using the TransBIOS/NetBIOS ROM, you must install RIPLMEM.EXE.

Installing Invisible LAN Wallpaper (Optional)

Optionally, you can use the Invisible LAN logo as wallpaper for your Windows desktop.

To install the Invisible LAN wallpaper, proceed as follows:

- Step 1. Copy the file WALL30.BMP into your Windows disk directory.
- Step 2. Open the Windows Control Panel and double-click on the **Desktop** icon.
- Step 3. Select WALL30.BMP in the **Wallpaper** section, and select the **Tile** option.
- Step 4. Choose **OK**.

On a VGA display, your screen background is filled with 80 copies of the Invisible LAN logo.

SUMMARY: You can use the Invisible LAN logo as wallpaper for your Windows desktop.

Selecting Printer Drivers

When you configure Windows, you must install the correct printer drivers for the printers that you plan to use. Then, when you print from a Windows application, the print output is converted into a form appropriate for the particular printer.

When you print to a network printer, you must use the printer driver that corresponds to the printer on which the actual printing will occur.

For example, suppose that a server has a PostScript printer. Suppose that a workstation has LPT2 mapped to the server's printer. Then, when you install Windows on the workstation you must install the PostScript printer driver and connect it to port LPT2.

SUMMARY: When printing from Windows, you must use the appropriate printer driver for the printer on which the document is ultimately printed. In the case of a network printer, you must install and activate the printer driver that corresponds to the server's printer.

Part 3

Invisible RAM Memory Management

Introduction to Invisible RAM

Invisible RAM is a shadow RAM manager. It is a software package that provides additional memory to DOS applications. It also lets you load memory-resident programs outside of DOS memory.

In addition, Invisible RAM is an expanded memory manager. It lets you run application programs that are designed to take advantage of expanded memory.

There are seven different versions of Invisible RAM, each designed for a different type of memory hardware. Each version of Invisible RAM is optimized to fully exploit the capabilities of your memory hardware.

This chapter covers the following topics:

- What Invisible RAM does for you
- Hardware requirements
- Memory configuration information

What Invisible RAM Does For You

The Invisible RAM software does three things for you:

- Invisible RAM increases the size of DOS memory from the normal 640K up to 736K on a color system, or 704K on a monochrome system. This process is called *frontfill*.
- Invisible RAM lets you load memory-resident programs and device drivers (including the Invisible LAN software) outside of DOS memory, so that they do not use up memory that you need for your application programs. As much as 256K of memory can be made available to memory-resident programs and device drivers outside of DOS memory; this memory is called *shadow RAM*.
- If your hardware permits, Invisible RAM acts as an expanded memory manager, compatible with the expanded memory specification (EMS)

version 4.0. You can make many megabytes of expanded memory available to applications such as Lotus 1-2-3, Desqview, and Ventura Publisher.

The exact amount of additional memory you get varies depending on what hardware you have installed in your computer. Invisible RAM includes programs that let you determine what hardware you have installed, and configure the system to match your exact needs.

SUMMARY: Invisible RAM increases the size of DOS memory, creates shadow RAM that is used to load memory-resident programs outside of DOS memory, and acts as an expanded memory manager.

Hardware Requirements

There are seven versions of Invisible RAM, each of which supports a different type of hardware. The seven versions, and their hardware requirements, are:

- *Invisible RAM for the NEAT Chipset* — Requires a 286 or 386SX based computer with the NEAT chipset, and at least 1 megabyte of RAM. The NEAT chipset is manufactured by a company called Chips and Technologies. It provides shadow RAM, expanded memory, and extended memory.
- *Invisible RAM for the AT/386 Chipset* — Requires a 386 based computer with the AT/386 chipset, and at least 1 megabyte of RAM. The AT/386 chipset is manufactured by a company called Chips and Technologies. It provides shadow RAM and extended memory.
- *Invisible RAM 386* — Requires a 386, 386SX, or 486 based computer with at least 2 megabytes of RAM. Invisible RAM 386 works by switching the microprocessor into the *386 virtual mode*; thus, Invisible RAM 386 is also called the 386 virtual mode memory manager. Invisible RAM 386 converts part of your computer's extended memory into expanded memory and shadow RAM.
- *Invisible RAM for Invisible EMS* — Requires an Invisible EMS memory board. The Invisible EMS board is manufactured by Invisible Software, and it provides expanded memory and shadow RAM. Invisible EMS features true EMS 4.0 hardware, and is a good way to add memory management to computers with no other memory management hardware built-in.

- *Invisible RAM for Generic EMS* — Requires any expanded memory system compatible with EMS 4.0. Invisible RAM for Generic EMS is designed to work in conjunction with your existing expanded memory hardware. Invisible RAM adds shadow RAM capability to your expanded memory.
- *Invisible RAM for the SCAT Chipset* — Requires a 286 based computer with the SCAT chipset, and at least 1 megabyte of RAM. The SCAT chipset is manufactured by a company called Chips and Technologies. It provides shadow RAM, expanded memory, and extended memory.
- *Invisible RAM for the PEAK Chipset* — Requires a 386 or 486 based computer with the PEAK chipset, and at least 1 megabyte of RAM. The PEAK chipset is manufactured by a company called Chips and Technologies. It provides shadow RAM and extended memory.

SUMMARY: The seven versions of Invisible RAM support seven different hardware configurations, so you can select the version that matches your hardware.

Memory Configuration Information

In this section we present information about memory in the PC. This information will help you to understand what Invisible RAM does, and it will help you to understand the Invisible RAM parameters that are described in later chapters.

Conventional Memory

The IBM PC can address 1024K bytes of memory. This 1024K is divided into *segments*, each containing 16 bytes. Thus, there are 65,536 different segments. It is conventional to number the segments using a *hexadecimal* numbering system (a *base-16* number system with digits that range from 0 to 9 and A to F). Thus, segments are numbered from 0000 to FFFF.

Segments 0000-9FFF are reserved for DOS. This is the *standard DOS 640K area*, also known as *conventional DOS memory*.

Segments B000-BFFF are reserved for the video system. Segments B000-B7FF are used for monochrome screens, and segments B800-BFFF are used for color screens. Segments A000-AFFF are used by the high-resolution

graphics modes of VGA and EGA video systems.

The highest-numbered segments (FFFF down) are used for the *system BIOS*, a ROM which is physically located on the motherboard. The size of the system BIOS can range from a little as 8K to as much as 128K. Most AT-type computers have either a 64K BIOS which occupies segments F000-FFFF, or a 32K BIOS which occupies segments F800-FFFF. Most XT-type computers have an 8K BIOS which occupies segments FE00-FFFF.

Segments from C000 to the bottom of the BIOS are available for use by plug-in adapter boards. Many adapters locate ROM in this area.

Extended Memory

The 286, 386, 386SX, and 486 microprocessors can address more than one megabyte of memory. Memory beyond one megabyte is called *extended memory*.

Like conventional memory, extended memory is numbered in segments. On the 286 and 386SX, there can be up to 15 megabytes of extended memory, occupying segments 10000 to FFFFFF. On the 386 and 486, there can be up to 4095 megabytes of extended memory, occupying segments 10000 to FFFFFFFF.

Unfortunately, extended memory is not directly accessible to DOS. This is because the microprocessor must be switched into a special mode of operation, called *protected mode*, in order to access the extended memory. DOS cannot operate in protected mode, and so DOS cannot use extended memory.

Note — With Invisible RAM 386, your computer's extended memory is converted into memory that DOS programs can use. Some of the extended memory is used to increase the size of DOS memory (*frontfill*), and create shadow RAM that can be used by memory-resident programs and device drivers. The rest of the extended memory can be converted into *expanded memory*, which can be used by many application programs.

Expanded Memory

Expanded memory is a method for adding as much as 32 megabytes of memory, while remaining within the 1024K addressing limit of the PC. Expanded memory is also called *EMS memory* (for Expanded Memory Specification) or *LIM memory* (for Lotus-Intel-Microsoft, the companies that originally defined expanded memory).

The expanded memory is divided into *pages*, each 16K bytes in size. Since

expanded memory can be as large as 32 megabytes, there can be as many as 2048 pages.

The PC's 1024K address space is divided into 64 *page frames*, each 16K bytes in size. Each page frame acts as a "window" into the expanded memory. The memory hardware allows any page frame to map into any page. Software can change the mapping at any time, so that every page frame can view any part of the expanded memory. As a result, a handful of 16K byte page frames can provide access to many megabytes of memory.

Not all of the 64 page frames can be used to access the expanded memory. For example, some of the page frames are required for the system BIOS, plug-in adapter boards, and the video system. A page frame may be used to access expanded memory only if it does not conflict with any equipment installed in the computer. Typically, page frames available for expanded memory are located in the area from segment C000 to the start of the BIOS, and, if the video system permits, in the area A000-AFFF.

Application programs do not exercise direct control over the expanded memory page frames. Instead, they make requests to a program called the *expanded memory manager*. The expanded memory manager is responsible for manipulating the expanded memory hardware.

Every expanded memory system provides at least four 16K page frames, arranged in a contiguous 64K block of memory. This is called the *standard EMS* area. For example, the standard EMS area may consist of four 16K page frames located in segments D000-D3FF, D400-D7FF, D800-DBFF, and DC00-DFFF; notice that the page frames form a contiguous 64K block from D000 to DFFF.

Some expanded memory systems provide additional page frames beyond the four standard page frames. The additional page frames are called the *enhanced EMS* area. They may not be contiguous. Enhanced EMS page frames can be created wherever there is a 16K block of memory that is not being used for some other purpose. For example, there may be enhanced EMS windows in segments A000-A3FF, A400-A7FF, A800-ABFF, AC00-AFFF, C800-CBFF, and CC00-CFFF; notice that they are not all contiguous.

Some expanded memory systems also create enhanced EMS page frames within the DOS 640K area. This allows certain programs, such as Desqview, to provide multitasking by changing the mapping of conventional DOS memory.

Note — The NEAT and SCAT chipsets have a built-in expanded memory controller, so you do not need a separate expanded memory board. The NEAT chipset's expanded memory controller only supports the four standard EMS page frames. The SCAT chipset's expanded memory controller allows 32 EMS page frames, in segments 4000-9FFF and D000-EFFF.

Note — The AT/386 and PEAK chipsets do not have a built-in expanded memory controller.

Note — Invisible RAM 386 can create expanded memory page frames anywhere in the PC's address space. Therefore, it supports both standard and enhanced EMS page frames, and it creates enhanced EMS page frames within the DOS memory space.

Note — Invisible EMS supports both standard and enhanced EMS page frames.

Shadow RAM And High RAM

We have explained how a page frame acts as a window into expanded memory. By calling the expanded memory manager, software can use a page frame to access any 16K byte page of expanded memory.

With Invisible RAM, it is possible to make some page frames access a *fixed* 16K page of memory. Such page frames are called *shadow RAM* or *high RAM*. A page frame used for shadow RAM or high RAM cannot access any 16K page of expanded memory; it only accesses one, fixed, unchangeable 16K page of memory.

From a hardware standpoint, shadow RAM is exactly the same as high RAM. The difference between shadow RAM and high RAM lies in the way they are utilized by software.

Shadow RAM is used by the **LSHADOW** programs supplied with Invisible RAM (see the *DOS User Manual* for details), and by the Invisible LAN operating system. The **LSHADOW** programs let you load TSR's (terminate-stay-resident programs) and DOS device drivers into the shadow RAM. Also, the Invisible LAN software has the ability to load itself into the shadow RAM. In either case, more DOS memory is freed up for application programs. Notice that you don't need specially-designed TSR's or device drivers in order to use shadow RAM; **LSHADOW** works with ordinary TSR's and device drivers.

High RAM is used only by specially-designed programs. It cannot be used by **LSHADOW** or by Invisible LAN. If you don't have any programs that are designed to work with high RAM, then there is no use in creating any high RAM.

Note — All versions of Invisible RAM can create shadow RAM or high RAM in any of the 24 page frames that occupy segments A000-FFFF. In the case of the NEAT, AT/386, SCAT, and PEAK chipsets, this is done using special hardware that is built-in to the chipset. In the case of Invisible RAM 386, Invisible EMS, and Generic EMS, this is done using the same page-mapping mechanism that is used to create expanded memory page frames.

SUMMARY: The PC's one-megabyte address space is divided into 64 *page frames*, each 16K bytes in size. The first 640K is always used for DOS. The remaining 384K is used for BIOS ROMs, video RAM, expanded memory "windows", shadow RAM, and high RAM. Expanded memory "windows" are *paged*, so any window can access any part of the expanded memory. Shadow RAM is used for memory-resident programs, including the Invisible LAN software.

Memory Hardware Configuration

This chapter describes how to configure your memory hardware so it is ready to run the Invisible RAM memory manager. In most cases, all you'll have to do is verify that you have the correct hardware.

We describe the configuration process separately for each of the seven types of Invisible RAM:

- The NEAT chipset
- The AT/386 chipset
- The SCAT chipset
- The PEAK chipset
- 386 virtual mode
- Invisible EMS
- Generic EMS

Please refer to the appropriate section of this chapter for your memory hardware.

The NEAT Chipset

The NEAT chipset supports shadow RAM, expanded memory, and extended memory. Before installing Invisible RAM, you must configure the NEAT chipset to specify how much of each type of memory you want to have. The actual configuration is done using your computer's set-up program. After the NEAT chipset is configured, you can install Invisible RAM.

The configuration you select depends on the following two factors:

- The total amount of RAM installed in your motherboard.
- The amount RAM you want to use for expanded memory.

The following sections describe how to select the appropriate configuration.

Determining Total RAM

In order to configure the NEAT chipset correctly, you need to know how much RAM you have installed on the motherboard.

To determine the amount of RAM on the motherboard, type the command

```
SHADOW /C
```

The **SHADOW** program produces a display similar to the following:

```
Chips and Technologies chipset: NEAT version B.  
Total RAM on motherboard: 1024K.  
640K to 1M memory relocation: Disabled.  
Shadow RAM: Available.  
Expanded memory controller: Disabled.
```

The second line shows the amount of RAM on the motherboard; it can range from 512K to 8192K. In the above example, there is 1024K RAM on the motherboard.

Stop — You need at least 1024K RAM to use Invisible RAM. If there is less than 1024K total RAM, you can't use Invisible RAM.

If You Have Exactly 1024K Total RAM

This section describes the options available to you if you have *exactly* 1024K RAM on the motherboard. If you have *more* than 1024K, skip ahead to the next section.

With exactly 1024K RAM on the motherboard, you can elect to have *either* shadow RAM *or* expanded memory, but not both:

- If you elect to have shadow RAM, you can load memory-resident programs and device drivers into the shadow RAM and you can increase DOS memory size above 640K, but you can't use expanded memory applications.
- If you elect to have expanded memory, you can use expanded memory applications, but you can't load memory-resident programs and device drivers outside DOS memory, and you can't increase DOS memory size above 640K.

The following paragraphs describe how to use your computer's set-up program to select the option you want.

Option 1 — Shadow RAM

With this option, you can load memory-resident programs and device drivers into shadow RAM, and you can increase DOS memory beyond 640K. However, you can't use the NEAT chipset's built-in expanded memory controller.

To select this option, start your computer's set-up program. The procedure for starting the set-up program varies from one computer to the next. Some set-up programs are loaded in from disk, while others are built-in to the computer and activated by pressing a special key when the computer is started.

Once the set-up program is started, configure the NEAT chipset as follows:

- For "640K to 1M memory relocation," select "Disabled."
- For "Expanded memory controller," select "Disabled."
- If your set-up program has a selection for "RAS timeout," select "Enabled."

After running the set-up program, re-boot your computer to put the new configuration into effect.

Option 2 — Expanded Memory

With this option, you can use the NEAT chipset's built-in expanded memory controller to run expanded memory applications. However, you can't load memory-resident programs or device drivers into shadow RAM, and you can't increase DOS memory beyond 640K.

To select this option, start your computer's set-up program. The procedure for starting the set-up program varies from one computer to the next. Some set-up programs are loaded in from disk, while others are built-in to the computer and activated by pressing a special key when the computer is started.

Once the set-up program is started, configure the NEAT chipset as follows:

- For "640K to 1M memory relocation," select "Enabled."
- For "Expanded memory controller," select "Enabled."
- For "EMS I/O port address," you can select any of the following: 208, 218, 258, 268, 2A8, 2B8, or 2E8. You must select an I/O address that is not used by any equipment in your computer. In most cases, we recommend selecting "208."
- For "EMS memory address," you can select any of the following: C000, C400, C800, CC00, D000, D400, D800, DC00, or E000. You must select the address of a 64K block of memory that is not used by any equipment in your computer. In most cases, we recommend selecting "E000."

- For “EMS size,” select “less than 1M.” (**Note** — Some set-up programs use “0.5M” to indicate “less than 1M.”)
- If your set-up program has a selection for “RAS timeout,” select “Enabled.”

After running the set-up program, re-boot your computer to put the new configuration into effect.

If You Have More Than 1024K Total RAM

This section describes the options available to you if you have *more* than 1024K RAM on the motherboard. If you have *exactly* 1024K, refer to the previous section.

With more than 1024K RAM on the motherboard, you always have shadow RAM available. The only decision you need to make is whether or not you want to have expanded memory. If you elect to have expanded memory, you need to decide how much you want to have; any RAM not used for expanded memory automatically becomes extended memory.

The following paragraphs describe how to use your computer’s set-up program to select the option you want.

Option 1 — No Expanded Memory

With this option, you can load memory-resident programs and device drivers into shadow RAM, and you can increase DOS memory beyond 640K. However, you can’t use the NEAT chipset’s built-in expanded memory controller. All RAM beyond the first 1024K is configured as extended memory.

To select this option, start your computer’s set-up program. The procedure for starting the set-up program varies from one computer to the next. Some set-up programs are loaded in from disk, while others are built-in to the computer and activated by pressing a special key when the computer is started.

Once the set-up program is started, configure the NEAT chipset as follows:

- For “640K to 1M memory relocation,” select “Disabled.”
- For “Expanded memory controller,” select “Disabled.”
- If your set-up program has a selection for “RAS timeout,” select “Enabled.”

After running the set-up program, re-boot your computer to put the new configuration into effect.

Option 2 — Expanded Memory

With this option, you can use the NEAT chipset's built-in expanded memory controller to run expanded memory applications. You can also load memory-resident programs and device drivers into shadow RAM, and you can increase DOS memory beyond 640K.

To select this option, start your computer's set-up program. The procedure for starting the set-up program varies from one computer to the next. Some set-up programs are loaded in from disk, while others are built-in to the computer and activated by pressing a special key when the computer is started.

Once the set-up program is started, configure the NEAT chipset as follows:

- For "640K to 1M memory relocation," select "Disabled."
- For "Expanded memory controller," select "Enabled."
- For "EMS I/O port address," you can select any of the following: 208, 218, 258, 268, 2A8, 2B8, or 2E8. You must select an I/O address that is not used by any equipment in your computer. In most cases, we recommend selecting "208."
- For "EMS memory address," you can select any of the following: C000, C400, C800, CC00, D000, D400, D800, DC00, or E000. You must select the address of a 64K block of memory that is not used by any equipment in your computer. In most cases, we recommend selecting "E000."
- For "EMS size," select the amount of RAM you want to use for expanded memory. The choices are "less than 1M," "1M," "2M," "3M," "4M," "5M," "6M," and "7M." (Note — Some set-up programs use "0.5M" to indicate "less than 1M.") The maximum value you can select is the total amount of RAM on the motherboard, minus 1024K. (For example, if there is 4096K RAM, the maximum EMS size you can select is 3072K, which is the same as 3M.) If you select less than the maximum possible size, the remaining RAM automatically becomes extended memory.
- If your set-up program has a selection for "RAS timeout," select "Enabled."

After running the set-up program, re-boot your computer to put the new configuration into effect.

Set-Up Program Problems

On rare occasions, there may be problems with a NEAT chipset computer when you attempt to change the configuration of the system. This happens mostly with older versions of NEAT computers. For example, there may be a

problem when you add or remove expansion boards, add or remove memory, or change the Invisible RAM parameters. The computer may “remember” the old configuration and refuse to adapt itself to the new configuration, even if you run the computer’s set-up program.

We have included a program called **CLRCMOS** that completely erases the computer’s old configuration. If you have problems changing the configuration of a NEAT chipset computer, run **CLRCMOS**. Then you can run the computer’s set-up program, and establish the desired new configuration.

Caution — After running **CLRCMOS**, you will have to set up the computer from scratch (entering memory size, diskette drive types, fixed disk drive type numbers, and so on). Don’t run **CLRCMOS** unless you know how to do this.

SUMMARY: The NEAT chipset is very flexible. It supports shadow RAM, expanded memory, and extended memory. You use your computer’s set-up program to configure the NEAT chipset, in order to select the combination of memory you want.

The AT/386 Chipset

The AT/386 chipset supports shadow RAM, and extended memory. All you have to do is verify that you have the AT/386 chipset, and at least 1024K RAM on the motherboard.

To determine the amount of RAM on the motherboard, type the command

```
SHADOW /C
```

The **SHADOW** program produces a display similar to the following:

```
Chips and Technologies chipset: AT/386 version B.  
Total RAM on motherboard: 1024K.  
Shadow RAM: Available.
```

The second line shows the amount of RAM on the motherboard; it can range from 1024K to 16384K. In the above example, there is 1024K RAM.

If the **SHADOW** display is correct, you may proceed to use Invisible RAM.

SUMMARY: The AT/386 chipset supports shadow RAM and extended memory.

SCAT Chipset Configuration

The SCAT chipset is used in 286-based computers. It supports shadow RAM, expanded memory, and extended memory. Before installing Invisible RAM, you must configure the SCAT chipset to specify how much of each type of memory you want to have. The actual configuration is done using your computer's set-up program. After the SCAT chipset is configured, you can install Invisible RAM.

Invisible RAM for the SCAT chipset can create shadow RAM, high RAM, or ROM copy in any of the 24 page frames in segments A000-FFFF. In addition, it can create paged EMS memory in the 32 page frames in segments 4000-9FFF and D000-EFFF.

The configuration you select depends on the following two factors:

- The total amount of RAM installed in your motherboard.
- The amount RAM you want to use for expanded memory.

The following sections describe how to select the appropriate configuration.

Determining Total RAM

In order to configure the SCAT chipset correctly, you need to know how much RAM you have installed on the motherboard.

To determine the amount of RAM on the motherboard, type the command

```
SHADOW /C
```

The **SHADOW** program produces a display similar to the following:

```
Chips and Technologies chipset: SCAT version A.  
Total RAM on motherboard: 4096K.  
640K to 1M memory relocation: Disabled.  
Shadow RAM: Available.  
Expanded memory controller: Enabled.  
EMS memory size: 1024K.  
Expanded memory I/O ports: Enabled.  
EMS I/O port address: 208.
```

The second line shows the amount of RAM on the motherboard; it can range

from 512K to 8192K. In the above example, there is 4096K RAM on the motherboard.

Stop—You need at least 1024K RAM to use Invisible RAM. If there is less than 1024K total RAM, you can't use Invisible RAM.

If You Have Exactly 1024K Total RAM

This section describes the options available to you if you have *exactly* 1024K RAM on the motherboard. If you have *more* than 1024K, skip ahead to the next section.

With exactly 1024K RAM on the motherboard, you can elect to have *either* shadow RAM *or* expanded memory, but not both:

- If you elect to have shadow RAM, you can load memory-resident programs and device drivers into the shadow RAM and you can increase DOS memory size above 640K, but you can't use expanded memory applications.
- If you elect to have expanded memory, you can use expanded memory applications, but you can't load memory-resident programs and device drivers outside DOS memory, and you can't increase DOS memory size above 640K.

The following paragraphs describe how to use your computer's set-up program to select the option you want.

Option 1 — Shadow RAM

With this option, you can load memory-resident programs and device drivers into shadow RAM, and you can increase DOS memory beyond 640K. However, you can't use the SCAT chipset's built-in expanded memory controller.

To select this option, start your computer's set-up program. Once the set-up program is started, configure the SCAT chipset as follows:

- For "640K to 1M memory relocation," select "Disabled."

After running the set-up program, re-boot your computer to put the new configuration into effect.

Option 2 — Expanded Memory

With this option, you can use the SCAT chipset's built-in expanded memory controller to run expanded memory applications. However, you can't load

memory-resident programs or device drivers into shadow RAM, and you can't increase DOS memory beyond 640K.

To select this option, start your computer's set-up program. Once the set-up program is started, configure the SCAT chipset as follows:

- For "640K to 1M memory relocation," select "Enabled." This gives you 384K of extended and/or expanded memory.
- Select an "Extended memory boundary" to define the top address of extended memory. Select "No boundary" if you want 384K of extended memory. Select "1M" if you want 384K of expanded memory. Select "1.25K" if you want 256K of extended memory and 128K of expanded memory.

After running the set-up program, re-boot your computer to put the new configuration into effect.

If You Have More Than 1024K Total RAM

This section describes the options available to you if you have *more* than 1024K RAM on the motherboard. If you have *exactly* 1024K, refer to the previous section.

With more than 1024K RAM on the motherboard, you always have shadow RAM available. The only decision you need to make is whether or not you want to have expanded memory. If you elect to have expanded memory, you need to decide how much you want to have; any RAM not used for expanded memory automatically becomes extended memory.

To configure the chipset, start your computer's set-up program. Once the set-up program is started, configure the SCAT chipset as follows:

- Select an "Extended memory boundary" to define the top address of extended memory. Select "No boundary" if you want all memory beyond the first megabyte to be extended memory. Select "1M" if you want all memory beyond the first megabyte to be expanded memory. Other boundary values give you a mix of extended and expanded memory.

After running the set-up program, re-boot your computer to put the new configuration into effect.

SUMMARY: The SCAT chipset is very flexible. It supports shadow RAM, expanded memory, and extended memory. You use your computer's set-up program to configure the SCAT chipset, in order to select the combination of memory you want.

PEAK Chipset Configuration

The PEAK chipset supports shadow RAM and extended memory. All you have to do is verify that you have the PEAK chipset, and at least 1024K RAM on the motherboard.

Invisible RAM for the PEAK chipset can create shadow RAM, high RAM, or ROM copy in any of the 24 page frames in segments A000-FFFF. PEAK does not support paged EMS memory.

To determine the amount of RAM on the motherboard, type the command

```
SHADOW /C
```

The **SHADOW** program produces a display similar to the following:

```
Chips and Technologies chipset: PEAK version D.  
Total RAM on motherboard: 4096K.  
Shadow RAM: Available.  
Cache: Enabled.  
Cache size: 256K.
```

The second line shows the amount of RAM on the motherboard; it can range from 1024K to 16384K. In the above example, there is 4096K RAM on the motherboard.

If the **SHADOW** display is correct, you may proceed to use Invisible RAM.

SUMMARY: The PEAK chipset supports shadow RAM and extended memory.

386 Virtual Mode

The 386 virtual mode supports shadow RAM, expanded memory, and extended memory. It works by taking part of your computer's extended memory, and converting it into shadow RAM and expanded memory.

Invisible RAM 386 is the version of Invisible RAM that supports the 386 virtual mode. Before installing Invisible RAM 386, you need to verify that you have the required hardware.

Microprocessor Type

Invisible RAM 386 requires that you have a 386, 386SX, or 486 microprocessor.

Check your computer and make sure that it has the correct type of microprocessor. Many computers display the microprocessor type when they are powered on.

Determining Extended Memory Size

To determine the amount of extended memory in the computer, type the command

```
SHADOW /E
```

The **SHADOW** program produces a display similar to the following:

```
Extended memory size: 3072K.
```

The display shows the amount of extended memory in the system. In the above example, there is 3072K (3 megabytes) of extended memory.

Stop — You need at least 1024K of extended memory to use Invisible RAM 386.

Check Extended Memory Applications

Check your system to see if any of your any applications use extended memory. The most common extended memory applications are disk cache programs (including the Invisible LAN disk cache), and RAMDISK programs (such as VDISK).

If you have any extended memory applications, take note of the total extended memory they require. You will need this information when you install Invisible RAM 386.

SUMMARY: Invisible RAM 386 takes advantage of the virtual mode of the the 386, 386SX, and 486 microprocessors. In the virtual mode, part of your computer's extended memory is converted into expanded memory and shadow RAM.

Invisible EMS

Invisible EMS is an expanded memory board manufactured by Invisible Software. It supports shadow RAM and expanded memory.

To install the Invisible EMS board, follow the instructions in the Invisible EMS User Manual which comes with the board.

The Invisible RAM program for Invisible EMS is in a file called IEMSEMM.SYS. A copy of this file is included with Invisible LAN. Another copy of this file is included with the Invisible EMS board. You should compare the dates on the two files; use whichever one is more recent.

SUMMARY: The Invisible EMS board provides expanded memory and shadow RAM.

Generic EMS

Invisible RAM for Generic EMS is designed to work in conjunction with any EMS 4.0 hardware. Invisible RAM adds shadow RAM and frontfill features to your existing EMS 4.0 hardware.

The following sections describe how to configure your existing EMS 4.0 so that it can accommodate Invisible RAM.

Install the Underlying EMS

Invisible RAM for Generic EMS requires that you have EMS 4.0 hardware and software in your computer. Your existing EMS hardware and software is called the *underlying EMS*. Invisible RAM runs "on top of" the underlying EMS.

Install your EMS hardware and software in accordance with the

manufacturer's instructions. Installing the software generally involves placing a `DEVICE=` statement in the `CONFIG.SYS` file. After installing the EMS software, you may have to reboot your computer.

After installing the underlying EMS, use the **SHADOW** program to verify that it is correctly installed. Type the command

```
SHADOW /X
```

The **SHADOW** program produces a display similar to the following:

```
Expanded Memory Manager: Installed.  
Version number: 4.0.  
Total expanded memory: 2048K.  
Free expanded memory: 1824K.  
EEMS (enhanced EMS) functions: Available.
```

The first line shows that your EMS software is installed, and the second line shows its version number. Invisible RAM requires version number 4.0 or higher.

The third line shows the total amount of expanded memory you have, and the fourth line shows the amount of free expanded memory.

The last line shows if your expanded memory software supports the enhanced EMS functions. Invisible RAM does not require the enhanced EMS functions, so it doesn't matter if they are available or not.

Check the Memory Map

With the underlying EMS installed, display the memory map by entering the command

```
SHADOW /M
```

The **SHADOW** program produces a display similar to the following:

0000-9FFF	DOS RAM
A000-B7FF	ENHANCED-EMS
B800-BFFF	VIDEO RAM
C000-C5FF	ROM
C600-C7FF	
C800-D7FF	STANDARD-EMS
D800-EFFF	ENHANCED-EMS
F000-FFFF	BIOS

Look for areas of memory labeled **ENHANCED-EMS**. Invisible RAM can convert these areas into shadow RAM. If there are no areas labeled **ENHANCED-EMS**, then Invisible RAM is probably not going to be useful for you.

Note — If there are no areas labeled **ENHANCED-EMS**, then you may not have true EMS 4.0 hardware. There has been confusion in the industry, because some manufacturers sell EMS 4.0 software with old-style EMS 3.2 hardware. To have **ENHANCED-EMS**, you need true EMS 4.0 hardware. If you believe that you have true EMS 4.0 hardware, but there is still no **ENHANCED-EMS**, check to see if your EMS software is correctly configured.

Invisible RAM for Generic EMS has two basic configurations:

- You can convert all the **ENHANCED-EMS** regions into shadow RAM, while leaving the **STANDARD-EMS** regions for use by expanded memory applications. This configuration gives you both shadow RAM and expanded memory.
- You can convert both the **ENHANCED-EMS** regions and the **STANDARD-EMS** regions into shadow RAM. This gives you an extra 64K of shadow RAM, however, the expanded memory functions are disabled. With this configuration, you have shadow RAM but no expanded memory.

The Invisible RAM configuration is selected with the memory manager parameters, described in chapter 13.

Checking for Frontfill

Frontfill is the process of increasing DOS memory beyond 640K. In most cases, DOS memory can be increased to 736K on a color system, or 704K on a monochrome system.

In order to perform frontfill, you must have **ENHANCED-EMS** beginning at

memory address A000. If the memory map does not show **ENHANCED-EMS** at memory address A000, you may be able to create it by reconfiguring your EMS software. The following sections describe two procedures that may work.

Using an “Include” Parameter

Many expanded memory managers accept a parameter that lets you tell the software where to create paged EMS memory. Usually, the parameter is called something like “include,” and is often abbreviated to the letter **I**. With the “include” parameter, you may be able to force your EMS software to create **ENHANCED-EMS** at memory address A000.

Example — Suppose that your EMS software is called EMM.SYS. You may be able to create **ENHANCED-EMS** at memory address A000 by putting the following line in the CONFIG.SYS file:

```
DEVICE=EMM.SYS INCLUDE=A000-AFFF
```

Using VGA2CGA.SYS

If you have an EGA or VGA video system, your EMS software may refuse to create **ENHANCED-EMS** at memory address A000, even if you use an “include” parameter. This is because EGA and VGA use memory locations A000-AFFF as video RAM for the high-resolution graphics modes. Your EMS software may be designed to check for the presence of EGA or VGA, and refuse to use memory locations A000 to AFFF if an EGA or VGA is installed.

You can overcome this limitation by using the device driver VGA2CGA.SYS. What this device driver does is disable the high-resolution graphics modes. In effect, it “tricks” your EMS software into thinking that you have a CGA video system, instead of the EGA or VGA. You need to install VGA2CGA.SYS immediately *before* you install your EMS software.

Example — Suppose that your EMS software is called EMM.SYS. You may be able to create **ENHANCED-EMS** at memory address A000 by installing VGA2CGA.SYS as shown:

```
DEVICE=VGA2CGA.SYS  
DEVICE=EMM.SYS
```

You may have to use both VGA2CGA.SYS and an “include” parameter, as follows:

```
DEVICE=VGA2CGA.SYS  
DEVICE=EMM.SYS INCLUDE=A000-AFFF
```

Note — After VGA2CGA.SYS is installed, the high-resolution graphics modes are disabled. When you install Invisible RAM, the high-resolution graphics modes are re-enabled.

SUMMARY: Invisible RAM for Generic EMS works in conjunction with any EMS 4.0 hardware. It converts the enhanced EMS page frames into shadow RAM. Frontfill can be provided if there are enhanced EMS pages at memory address A000.

Advanced Memory Configuration

This chapter describes all the parameters that are available for configuring the Invisible RAM memory managers.

The memory managers can be configured to match your exact requirements. There are seven different memory managers, each designed for a different type of memory hardware. Each memory manager is designed to fully exploit your hardware's capabilities.

All the parameters can be adjusted using the Invisible LAN Setup program (**SETUP30**). The parameters are stored in the Invisible LAN initialization file, and they go into effect the next time you start Invisible LAN.

The topics covered in this chapter are:

- Selecting a memory manager
- Page frames
- Memory manager parameters
- System memory configuration
- 386 control program parameters

Parameters that control the network software are described separately, in chapter 8.

Note — There are two ways to load an Invisible RAM memory manager. It can be loaded by the **NET30** command when you start Invisible LAN, or it can be loaded from the CONFIG.SYS file. These parameters only apply when Invisible RAM is loaded by the **NET30** command. If Invisible RAM is loaded from CONFIG.SYS, then the parameters must be placed into the CONFIG.SYS file; refer to the text file INVRAM.TXT for details.

Page Frames

Before discussing the various parameters, we present some basic information about page frames and the types of memory that Invisible RAM can create.

The IBM PC can address a total of 1024K of memory. Invisible RAM divides this 1024K into 64 *page frames*, each 16K in size. The first page frame occupies segments 0000-03FF, the second page frame occupies segments 0400-07FF, and so on; the last (sixty-fourth) page frame occupies segments FC00-FFFF.

Within each page frame, Invisible RAM can potentially create one of four types of memory:

- **Paged EMS memory.** A page frame set up as paged EMS memory can access any 16K of expanded memory. In effect, the page frame acts as a “window” into the expanded memory. The window can be moved under software control. Page frames set up as paged EMS memory can be used by any application that is designed to use expanded memory.
- **Shadow RAM.** A page frame set up as shadow RAM contains a fixed 16K block of memory. Page frames set up as shadow RAM can be used to load DOS device drivers and TSRs (terminate-stay-resident programs) outside of DOS memory. The programs **LSHADOW.COM** and **LSHADOW.SYS** provide access to shadow RAM. Also, Invisible LAN can load itself into shadow RAM.
- **High RAM.** A page frame set up as high RAM contains a fixed 16K block of memory. High RAM is very similar to shadow RAM, except that high RAM cannot be accessed with the **LSHADOW** programs. High RAM can only be used by specially-designed application programs.
- **ROM Copy.** A page frame set up as ROM copy is filled with RAM that contains a copy of whatever ROM was originally located in the page frame; the ROM itself is disabled. This speeds up the execution of ROM code, because RAM is faster than ROM.

The biggest difference among the five versions of Invisible RAM is their capabilities and limitations in how page frames can be used. Here is a summary of the differences:

- Invisible RAM for the NEAT chipset can only use the 24 page frames that lie within segments A000-FFFF, that is, the page frames that lie outside the DOS 640K area. Moreover, only four of these page frames can be used for paged EMS memory. However, all 24 page frames can be used for shadow RAM, high RAM, and ROM copy.
- Invisible RAM for the SCAT chipset can create shadow RAM, high RAM, or ROM copy in any of the 24 page frames in segments A000-

FFFF. In addition, it can create paged EMS memory in the 32 page frames in segments 4000-9FFF and D000-EFFF.

- Invisible RAM for the AT/386 or PEAK chipset can only use the 24 page frames that lie within segments A000-FFFF, that is, the page frames that lie outside the DOS 640K area. Paged EMS memory is not supported. However, all 24 page frames can be used for shadow RAM, high RAM, and ROM copy.
- Invisible RAM 386 (for 386 virtual mode) is the most flexible. It can use all 64 page frames, and it can make any type of memory in each page frame.
- Invisible RAM for the Invisible EMS memory board can create paged EMS memory, shadow RAM, or high RAM in any available page frame (it cannot create ROM copy). However, the available page frames are limited by the design of the computer. In most cases, the available page frames occupy one of the following address ranges: A000-DFFF; or A000-EFFF; or 8000-DFFF; or 8000-EFFF.
- Invisible RAM for Generic EMS can create shadow RAM or high RAM in any available page frame (it cannot create ROM copy). Creation of paged EMS memory is not an issue for Generic EMS, since the underlying expanded memory manager provides paged EMS memory. The available page frames are determined by the expanded memory manager.

SUMMARY: Memory is divided into *page frames*, each 16K bytes in size. In each page frame, Invisible RAM can potentially create paged EMS memory, shadow RAM, high RAM, or ROM copy.

Memory Manager Parameters

The Invisible RAM parameters are entered on two screens. The Memory Manager Parameters screen controls the overall configuration of the memory manager. The System Memory Configuration screen controls the use of each individual page frame.

This section describes the Memory Manager Parameters screen, as shown in figure 13-1. The next section describes the System Memory Configuration screen.

Note — In most cases, you only have to adjust the **Memory Manager** and **Configuration** fields. The other fields automatically assume default values that are correct for most installations.

Memory Manager

The **Memory Manager** field selects one of the Invisible RAM memory managers. The optional memory managers can reduce network memory requirements and increase the memory available to DOS. Use **F5** and **F6** to select one of the choices in the following table.

<i>Parameter</i>	<i>Hardware</i>
NONE	No memory manager.
NEAT_CHIPSET	Invisible RAM, for 286- and 386SX-based computers with the NEAT chipset from Chips and Technologies.
AT/386_CHIPSET	Invisible RAM, for 386-based computers with the AT/386 chipset from Chips and Technologies.
SCAT_CHIPSET	Invisible RAM, for 286-based computers with the SCAT chipset from Chips and Technologies.
PEAK_CHIPSET	Invisible RAM, for 386- and 486-based computers with the PEAK chipset from Chips and Technologies.
386_VIRTUAL	Invisible RAM 386, for any 386-, 386SX-, and 486-based computers with at least 2MB RAM. This choice installs both a memory manager, and a 386 control program that switches the processor into virtual mode.
INVISIBLE_EMS	Invisible EMS expanded memory board. This memory board is manufactured by Invisible Software.
GENERIC_EMS	Invisible RAM Generic, which can be used with any EMS 4.0 hardware.

Configuration

The **Configuration** field selects one of three basic configurations. In effect, this field establishes the default configuration for Invisible RAM. The other parameters act to modify the default configuration selected by this field.

ADVANCED CONFIGURATION			
INVISIBLE RAM MEMORY MANAGER PARAMETERS			
Memory Manager	<input type="text" value="386_VIRTUAL"/>	Extended Emulation Size	<input type="text"/>
Configuration	<input type="text" value="EMS_AND_SHADOW"/>	Initial VGA State	<input type="text" value="ON"/>
Code Location	<input type="text" value="SHADOW"/>	Contexts (32-255)	<input type="text" value="64"/>
Frontfill Size	<input type="text"/>	Handles (32-255)	<input type="text" value="64"/>
Frame Address	<input type="text" value="AUTO"/>	Test Level	<input type="text" value="DEFAULT"/>
Frame Size	<input type="text" value="4"/>	Non-standard Page	<input type="text"/>
I/O Port	<input type="text"/>		

Press F5 or F6 to select the memory manager you want to install, if any
 HELP F1 DONE F9 CANCEL Esc SELECT Enter ↑ ↓ Tab OPTION F5=Back F6=Next A-Z

Figure 13-1. Memory Manager Parameters Screen

- **EMS_ONLY** selects an expanded memory only configuration, with no shadow RAM. Invisible RAM creates paged EMS memory in as many page frames as possible.
- **EMS_AND_SHADOW** selects a configuration that supports both expanded memory and shadow RAM. Invisible RAM locates all available page frames. Four of the page frames are set up as paged EMS memory, and the rest are set up as shadow RAM.
- **SHADOW_ONLY** selects a shadow RAM only configuration, with no expanded memory support. Invisible RAM locates all available page frames, and sets them up as shadow RAM. This configuration creates an additional 64K of shadow RAM, as compared to **EMS_AND_SHADOW**.

Note — The AT/386 and PEAK versions of Invisible RAM only support the **SHADOW_ONLY** configuration, since the AT/386 and PEAK chipsets do not have an EMS controller.

Note — The Generic EMS version of Invisible RAM does not support **EMS_ONLY**. Both **EMS_AND_SHADOW** and **SHADOW_ONLY** are supported; if you specify **SHADOW_ONLY** then the underlying expanded memory manager is disabled.

Standard EMS Area Control

There are two types of paged EMS page frames: *standard* and *enhanced*. The standard EMS page frames always form a contiguous block of memory, called the *standard EMS area*.

By default, there are four standard EMS page frames, arranged in a contiguous 64K block. Invisible RAM automatically selects an appropriate starting address. All other EMS page frames automatically become enhanced EMS page frames.

Most expanded memory applications use only the standard EMS page frames. Very few require any enhanced EMS page frames.

With the **Frame Address** and **Frame Size** fields, you can customize the location and size of the standard EMS area.

Note — These fields are valid only for Invisible RAM 386, Invisible EMS, and the SCAT chipset. In the case of the NEAT chipset, the location of the standard EMS area is set with your computer's setup program.

Frame Address

The **Frame Address** parameter specifies the starting address of the standard EMS area. Use **F5** and **F6** to select a value.

If you select **AUTO**, then Invisible RAM automatically determines the starting address of the standard EMS area.

Example — If you would like to place the 64K standard EMS area in segments E000-EFFF, select **E000**.

Note — The **Frame Address** field does not create paged EMS memory. It only specifies which paged EMS memory should be used as standard EMS page frames. If you need to create paged EMS memory, use the System Memory Configuration screen described later in this chapter. (However, if you select **EMS_AND_SHADOW** for **Configuration**, then the **Frame Address** field specifies which page frames to use as paged EMS memory; remaining page frames are used as shadow RAM.)

Frame Size

The **Frame Size** field specifies the size of the standard EMS area. Normally, the standard EMS area consists of 4 page frames (64K). However, with **Frame Size** you can adjust the size from a minimum of 2 page frames to a maximum of 12 page frames. Note that there must be a contiguous block of paged EMS memory large enough to hold the specified standard EMS area. Use **F5** and **F6** to select a value.

Note — Most expanded memory applications (including Invisible LAN) won't work if the value of **Frame Size** is different from 4.

DOS Memory Control

By default, Invisible RAM automatically makes DOS memory as large as possible. The **Frontfill Size**, **Code Location**, and **Initial VGA State** parameters let you alter this behavior, if you so desire.

Frontfill Size

The **Frontfill Size** parameter lets you control the total size of DOS memory. In most cases, Invisible RAM automatically increases the size of DOS memory to 704K on a monochrome system, or 736K on a color system. This process is called *frontfill*. With the **Frontfill Size** parameter, you can control the amount of frontfill.

Using **F5** and **F6**, you can select a **Frontfill Size** of 0, 640, 704, or 736; or you can leave the field blank.

- If you leave the field blank, then Invisible RAM automatically makes DOS memory as large as possible.
- If you select **0**, then Invisible RAM does not change the size of DOS memory.
- If you select **640**, then Invisible RAM sets the size of DOS memory to 640K.
- If you select **704**, then Invisible RAM sets the size of DOS memory to 704K.
- If you select **736**, then Invisible RAM sets the size of DOS memory to 736K.

Note — Page frames not used for frontfill become available for use as paged EMS memory, shadow RAM, or high RAM.

Note — If you have VGA or EGA video, you can still specify a **Frontfill Size** larger than 640. When Invisible RAM first loads, DOS memory size is limited to a maximum of 640K (unless you specify **OFF** for **Initial VGA State**). However, Invisible RAM remembers the value you specified for **Frontfill Size**, and DOS memory size is increased to the specified value when you run **VGAOFF** (refer to the *DOS User Manual* for details about **VGAOFF**).

Code Location

When there is shadow RAM in the system not used for frontfill, Invisible RAM can copy itself into the shadow RAM. This is controlled by the **Code Location** parameter.

Using **F5** and **F6**, you can select a **Code Location** of **SHADOW** or **DOS**.

- If you specify **SHADOW**, then Invisible RAM copies itself into shadow RAM. This frees up approximately 9K of additional DOS memory.
- If you specify **DOS**, then Invisible RAM remains in DOS memory.

SHADOW is the default value of **Code Location**.

Initial VGA State

Initial VGA State lets you specify whether high-resolution VGA or EGA graphics should be enabled or disabled. Disabling high-resolution graphics allows DOS memory size to be increased above 640K.

Using **F5** and **F6**, you can select an **Initial VGA State** of **ON** or **OFF**.

- If you specify **ON**, then VGA or EGA high-resolution graphics is enabled.
- If you specify **OFF**, then VGA or EGA high-resolution graphics is disabled.

ON is the default value of **Initial VGA State**.

Note — Regardless of the value you specify for **Initial VGA State**, you can use **VGAON** and **VGAOFF** to enable and disable high-resolution graphics as needed. Refer to the *DOS User Manual* for details.

Note—**Initial VGA State** is ignored if you don't have a VGA or EGA video system.

Internal Expanded Memory Parameters

The **Contexts**, **Handles**, and **Non-standard Page** parameters adjust certain internal parameters of the expanded memory manager. Generally, they are needed only for special applications.

Handles

The **Handles** parameter controls the number of EMS handles. *Handles* are used by expanded memory application programs to request memory from the

expanded memory manager. Generally, you need one or two handles for each expanded memory application that is running concurrently in the computer.

The default value of **Handles** is 64, which should be adequate for almost any application. However, if an application reports that there are no expanded memory handles available, increase the value of **Handles**.

This parameter is valid for NEAT chipset, SCAT chipset, Invisible RAM 386, and Invisible EMS.

Contexts

The **Contexts** parameter controls the number of EMS contexts. *Contexts* are used by some (but not all) expanded memory application programs.

The default value of **Contexts** is 64, which should be adequate for almost any application. However, if an application reports that there are no expanded memory contexts available, increase the value of **Contexts**.

This parameter is valid for NEAT chipset, SCAT chipset, Invisible RAM 386, and Invisible EMS.

Non-Standard Page

The **Non-standard Page** parameter lets you create enhanced EMS page frames with non-standard page frame numbers. This is used only with IBM DOS 4.0 (*not* with MS-DOS 4.0), to use the expanded memory features of DOS.

This parameter is valid for SCAT chipset, Invisible RAM 386, and Invisible EMS.

You can create page number 254, or page number 255, or both. You need to create page number 254 if you want to use VDISK or FASTOPEN in expanded memory. You need to create page number 255 if you want to use BUFFERS in expanded memory.

Note — **Non-standard Page** does not create paged EMS memory. It merely specifies that one or two of the enhanced EMS page frames should be reserved for DOS (note that **Non-standard Page** can't use a standard EMS page frame). If necessary, use the System Memory Configuration screen, described later in this chapter, to create paged EMS memory.

Note — This parameter is not very useful when Invisible RAM is loaded by the **NET30** command. To use the non-standard pages, you need to load Invisible RAM from CONFIG.SYS; refer to INVRAM.TXT for details.

Extended Memory Emulation

The **Extended Emulation Size** parameter is used only with Invisible EMS. It lets you use part of your expanded memory to emulate extended memory. This lets you run extended memory applications, such as the VDISK RAMDISK program.

Any expanded memory used to emulate extended memory is not available for use by expanded memory applications.

If you are not using extended memory emulation, enter 0 in the **Extended Emulation Size** field. The default value of **Extended Emulation Size** is 0.

Note — Don't use extended memory emulation if there is any actual extended memory in your system. Extended memory emulation is not valid if you select **SHADOW_ONLY** for **Configuration**.

Example — To use 512K of expanded memory as extended memory, enter 512 in the **Extended Emulation Size** field.

Technical note — The extended memory emulator only works with programs that access extended memory through interrupt INT 15H functions 87H and 88H. It does not work with programs that actually switch the processor into protected mode.

Hardware Configuration and Test

The **I/O Port** and **Test Level** parameters specify the configuration and testing of the memory hardware.

Test Level

When the memory manager is loaded, it automatically performs a test of the memory. Two tests are provided: a short test and a long test. Since the long test can take an annoyingly long time, the **Test Level** parameter is provided to let you select the memory test you want.

Using **F5** and **F6**, you can select from three options:

- If you select **SHORT**, the memory manager always uses the short test.
- If you select **LONG**, the memory manager always uses the long test.
- If you select **DEFAULT**, the memory manager performs the long test after a cold boot (power-on), and the short test after a warm boot (Ctrl-Alt-Del).

Note — We have found that with some versions of BIOS, **DEFAULT** always results in the long test. If you find this happens on your computer, simply use **SHORT** to get the short test.

I/O Port

The **I/O Port** parameter is used with Invisible EMS, to specify the I/O address used to access the Invisible EMS board. The value specified must match the I/O address you selected with the on-board DIP switch. Using **F5** and **F6**, you can select from 208, 218, 248, 258, 2A8, 2B8, 2E8, and 2F8. The default value is 208.

SUMMARY: The Memory Manager Parameters screen lets you control the configuration of the memory manager. In addition to selecting its overall configuration, you can also control DOS memory allocation, standard EMS page frame allocation, internal memory manager resources, and hardware configuration and test.

System Memory Configuration

The Invisible RAM parameters are entered on two screens. The Memory Manager Parameters screen controls the overall configuration of the memory manager. The System Memory Configuration screen controls the use of each individual page frame.

This section describes the System Memory Configuration screen, as shown in figure 13-2. The previous section described the Memory Manager Parameters screen.

Note — In most cases, you can leave this screen blank. Invisible RAM automatically determines a default memory configuration that is correct for most installations.

The System Memory Configuration screen lets you specify the usage of the 24 page frames that span addresses A000 to FFFF. Each page frame is 16K bytes in size. There is one field on the screen for each page frame; the field labeled **A000** is for page frame A000-A3FF; the field labeled **A400** is for page frame A400-A7FF; and so on.

In each field, you use **F5** and **F6** to select the desired usage of the page frame. There are seven options:

SET SYSTEM MEMORY CONFIGURATION			
A000		D000	
A400		D400	
A800		D800	
AC00		DC00	
B000		E000	SHADOW_RAM
B400		E400	SHADOW_RAM
B800		E800	SHADOW_RAM
BC00	DEFAULT_RAM	EC00	SHADOW_RAM
C000	ROM_COPY	F000	DEFAULT_RAM
C400	ROM_COPY	F400	DEFAULT_RAM
C800		F800	ROM_COPY
CC00		FC00	ROM_COPY

Press F5 or F6 to configure segments A000-A3FF, or blank for automatic
 HELP F1 DONE F9 CANCEL Esc SELECT Enter ↑ ↓ Tab OPTION F5=Back F6=Next A-Z

Figure 13-2. System Memory Configuration Screen

- If you leave the field blank, then Invisible RAM automatically determines whether or not the page frame is available; and if the page frame is available, Invisible RAM uses it in the default manner (either paged EMS memory or shadow RAM, depending on the value selected for **Configuration** on the Memory Manager Parameters screen).
- **EXCLUDE** tells Invisible RAM not to use the page frame, even if Invisible RAM thinks that the page frame is available.
- **DEFAULT_RAM** tells Invisible RAM to use the page frame in the default manner (either paged EMS memory or shadow RAM), even if Invisible RAM thinks that the page frame is unavailable.
- **INCLUDE** tells Invisible RAM to use the page frame for paged EMS memory. The paged EMS memory can be used by expanded memory applications, including the Invisible LAN software.
- **SHADOW_RAM** tells Invisible RAM to use the page frame for shadow RAM. The shadow RAM can be used to hold the Invisible LAN software, as well as other memory-resident programs and device drivers.
- **HIGH_RAM** tells Invisible RAM to use the page frame for unpagged high RAM. The high RAM can be used only by specially designed applications.
- **ROM_COPY** disables any ROM in the page frame, and replaces it with RAM; and the contents of the ROM is copied into the RAM. This speeds up the execution of ROM code.

The remainder of this section presents a further discussion of these options, and examples of their use.

Note — Page frames in the address range 0000 to 9FFF can also be controlled, however, this requires that you install Invisible RAM in the CONFIG.SYS file. Refer to INVRAM.TXT for details.

Note — You can display a complete memory map of your system by giving the command

SHADOW /M

This memory map can help you figure out how to customize your memory allocation. Refer to the *DOS User Manual* for a full description of the **SHADOW** program.

EXCLUDE

The **EXCLUDE** option lets you exclude one or more page frames. This means Invisible RAM does not use the specified page frames.

Normally, Invisible RAM automatically excludes all page frames that are already used by other equipment in the computer. However, Invisible RAM may not be able to detect all other equipment in the computer. With **EXCLUDE**, you can tell Invisible RAM not to use page frames that are required for other equipment.

Example — A VGA video board can support both color and monochrome video modes, regardless of what type of monitor you use. In color text mode, memory locations B800-BFFF contain video RAM, while memory locations B000-B7FF are “empty.” In monochrome text mode, memory locations B000-B7FF contain video RAM, while memory locations B800-BFFF are “empty.” Normally, when Invisible RAM is loaded, it takes over the “empty” half of the video RAM area (B000-BFFF). This causes a conflict, and crashes the system, if you switch from color to monochrome mode or vice-versa. So, if you need to switch between color and monochrome modes on a VGA, you need to exclude the entire video RAM area from B000 to BFFF. To do this, select **EXCLUDE** for page frames **B000**, **B400**, **B800**, and **BC00**.

Example — The IBM VGA graphics adapter uses memory locations CA00-CA7F. (Non-IBM graphics adapters do *not* use these locations. Also, the VGA that is built in to IBM PS/2 computers does *not* use these locations.) Invisible RAM is not able to detect this memory usage, and so it won't automatically exclude these locations. So, if you have an IBM VGA graphics adapter, you must select **EXCLUDE** for page frame **C800**.

Example — Many debugger boards (for example, Periscope) use memory locations that cannot be detected by Invisible RAM. If the debugger board uses memory locations D000-D7FF, you would select **EXCLUDE** for page frames **D000** and **D400**.

Example — This example applies only to Invisible RAM 386, Invisible EMS, and Generic EMS. If you have EGA or VGA graphics, Invisible RAM by default uses memory locations A000-AFFF as shadow RAM. This allows you to disable and enable high-resolution graphics with the **VGAOFF** and **VGAON** programs. If you never disable high-resolution graphics, you can free up an additional 64K of expanded memory by telling Invisible RAM not to use memory locations A000-AFFF. To do this, select **EXCLUDE** for page frames **A000**, **A400**, **A800**, and **AC00**.

Note — There is a special consideration in using the **EXCLUDE** with Generic EMS. With Generic EMS, there is an underlying expanded memory manager. Most memory managers have an “exclude” parameter of their own. If you need to exclude some memory, it is probably better to put the exclude parameter on the underlying expanded memory manager, if possible.

DEFAULT_RAM

The **DEFAULT_RAM** option lets you tell Invisible RAM to use one or more page frames. Invisible RAM uses the page frame in the default manner; that is, Invisible RAM uses the page frame for either paged EMS memory or shadow RAM, depending on the value of the **Configuration** parameter.

By default, Invisible RAM locates all available page frames and uses them in the default manner. So the main use for **DEFAULT_RAM** is to tell Invisible RAM to use a page frame where Invisible RAM can't tell that the page frame is available.

Example — This example applies to NEAT chipset, AT/386 chipset, SCAT chipset, PEAK chipset, and Invisible RAM 386. With monochrome video, segments B000-B7FF are used as video RAM. However, with some monochrome video boards, segments B400-B7FF are never used by the video system. You can let Invisible RAM use these segments by specifying **DEFAULT_RAM** for page frame **B400**.

Example — This example applies to NEAT chipset, AT/386 chipset, SCAT chipset, PEAK chipset, and Invisible RAM 386. With color video, segments B800-BFFF are used as video RAM. However, with some color video boards, segments BC00-BFFF are never used by the video system. You can let Invisible RAM use these segments by specifying **DEFAULT_RAM** for page frame **BC00**.

Example — This example applies to NEAT chipset, AT/386 chipset, SCAT chipset, PEAK chipset, and Invisible RAM 386. Many computers have a setup program in ROM, occupying segments F000-F7FF. The setup program is not required during normal computer operation. Therefore, you can make an additional 32K available to Invisible RAM by entering **DEFAULT_RAM** for page frames **F000** and **F400**.

Example — This example applies to Invisible EMS. Some AT-type computers allow the use of segments E000-EFFF, and some don't. By default, Invisible EMS does not use segments E000-EFFF when installed in an AT-type computer. If your computer allows the use of segments E000-EFFF, you can tell Invisible EMS to use these segments by selecting **DEFAULT_RAM** for page frames **E000**, **E400**, **E800**, and **EC00**.

INCLUDE

The **INCLUDE** option lets you set up one or more page frames as paged EMS memory. This means applications designed to use expanded memory are able to use the page frames.

This parameter is only valid for SCAT chipset, Invisible RAM 386, and Invisible EMS. (For the NEAT chipset, the location of paged EMS memory is determined by your computer's setup program.)

By default, Invisible RAM sets up all unused page frames as paged EMS memory or shadow RAM. So the two main uses for **INCLUDE** are:

- To create paged EMS memory in a page frame where Invisible RAM can't tell that the page frame is available.
- To create paged EMS memory in a page frame that Invisible RAM would otherwise set up as shadow RAM.

Example — This example applies to SCAT chipset and Invisible RAM 386. With monochrome video, segments B000-B7FF are used as video RAM. However, with some monochrome video boards, segments B400-B7FF can be converted into paged EMS memory. To do this, specify **INCLUDE** for page frame **B400**.

Example — This example applies to SCAT chipset and Invisible RAM 386. With color video, segments B800-BFFF are used as video RAM. However, with some color video boards, segments BC00-BFFF can be converted into paged EMS memory. To do this, specify **INCLUDE** for page frame **BC00**.

Example — This example applies to Invisible EMS. Some AT-type computers allow the use of segments E000-EFFF, and some don't. By default, Invisible EMS does not use segments E000-EFFF when installed in an AT-type computer. If your computer allows the use of segments E000-EFFF, you can tell Invisible EMS to create paged EMS memory in these segments by selecting **INCLUDE** for page frames **E000**, **E400**, **E800**, and **EC00**.

Example — EGA and VGA graphics adapters use segments A000-AFFF for high-resolution graphics. By default, Invisible RAM sets up these segments as shadow RAM, so that you can use high-resolution graphics. If you want to use segments A000-AFFF as paged EMS memory, you can select **INCLUDE** for page frames **A000**, **A400**, **A800**, and **AC00**. Note that using A000-AFFF as paged EMS memory permanently disables the high-resolution graphics;

you can't use **VGAON**. (Refer to the *DOS User Manual* for additional information on using EGA or VGA graphics.)

SHADOW_RAM

The **SHADOW_RAM** option lets you set up one or more page frames as shadow RAM. The shadow RAM can be used to hold memory-resident programs and device drivers, thus freeing up DOS memory. The **LSHADOW** programs allow access to the shadow RAM; refer to the *DOS User Manual* for details. Shadow RAM can also be used for the Invisible LAN operating system, by specifying the appropriate parameters in the Invisible LAN Setup program.

In most cases, you can allocate required shadow RAM by selecting **EMS_AND_SHADOW** or **SHADOW_ONLY** for the **Configuration** parameter on the Memory Manager Parameters screen. So the two main uses for **SHADOW_RAM** are:

- To create shadow RAM in a page frame where Invisible RAM can't tell that the page frame is available.
- To create shadow RAM in a page frame that Invisible RAM would otherwise set up as paged EMS memory.

Example — This example applies to NEAT chipset, AT/386 chipset, SCAT chipset, PEAK chipset, and Invisible RAM 386. With some monochrome video boards, you can get an extra 16K bytes of shadow RAM by turning on shadow RAM in the region from B400 to B7FF. To do this, select **SHADOW_RAM** for page frame **B400**.

Example — This example applies to NEAT chipset, AT/386 chipset, SCAT chipset, PEAK chipset, and Invisible RAM 386. With some color video boards, you can get an extra 16K bytes of shadow RAM by turning on shadow RAM in the region from BC00 to BFFF. To do this, select **SHADOW_RAM** for page frame **BC00**.

Example — This example applies to NEAT chipset, AT/386 chipset, SCAT chipset, PEAK chipset, and Invisible RAM 386. Many computers have a setup program in ROM, occupying segments F000-F7FF. The setup program is not required during normal computer operation. Therefore, you get an extra 32K of shadow RAM by selecting **SHADOW_RAM** for page frames **F000** and **F400**.

Example — This example applies to Invisible EMS. Some AT-type computers allow the use of segments E000-EFFF, and some don't. By default, Invisible EMS does not use segments E000-EFFF when installed in an AT-type computer. If your computer allows the use of segments E000-EFFF, you get an extra 64K of shadow RAM by selecting **SHADOW_RAM** for page frames **E000**, **E400**, **E800**, and **EC00**.

Example — This example applies to Invisible RAM 386 and Invisible EMS. Suppose you select the **EMS_ONLY** configuration, so that Invisible RAM creates paged EMS memory in all available page frames. If you would like to have 32K of shadow RAM at address C800, you would specify **SHADOW_RAM** for page frames **C800** and **CC00**.

Note — The **LSHADOW** programs can use shadow RAM much more efficiently if the shadow RAM is organized in one large, contiguous block, rather than a number of smaller blocks. When allocating shadow RAM, try to allocate large blocks rather than small blocks.

HIGH_RAM

The **HIGH_RAM** option lets you set up one or more page frames as high RAM. High RAM cannot be used by expanded memory applications, it cannot be accessed with the **LSHADOW** programs, and it cannot be used by Invisible LAN. High RAM can only be used by specially-designed programs.

By default, Invisible RAM never creates high RAM. So, if you want to have high RAM, you must create it explicitly with the **HIGH_RAM** option.

Example — Suppose you want 32K of high RAM at address C800. You would specify **HIGH_RAM** for page frames **C800** and **CC00**.

ROM_COPY

The **ROM_COPY** option lets you copy ROM code into RAM. The ROM in the page frame is disabled, and replaced with RAM; and the contents of the ROM is copied into the RAM. This speeds up the execution of ROM code, since RAM is faster than ROM.

This parameter is only valid for NEAT chipset, AT/386 chipset, SCAT chipset, PEAK chipset, and Invisible RAM 386.

By default, Invisible RAM does not perform any ROM copy; so if you want ROM copy, you must explicitly request it with the **ROM_COPY** option.

Exception — If you are using the TransBIOS/NetBIOS ROM on the network card, Invisible RAM automatically detects it and performs a ROM copy for the TransBIOS/NetBIOS ROM.

Example — Many computers have the system BIOS located at addresses F800-FFFF. You can speed up the system BIOS by copying it into RAM. To do this, select **ROM_COPY** for page frames **F800** and **FC00**.

Example — Many EGA or VGA video cards have a video BIOS ROM at addresses C000-C5FF. You can speed up video operations by copying it into RAM. To do this, select **ROM_COPY** for page frames **C000** and **C400**.

SUMMARY: The System Memory Configuration screen lets you control the use of each individual page frame.

386 Control Program Parameters

Invisible RAM 386 is structured differently from the other versions of Invisible RAM. Invisible RAM 386 consists of two programs: a *386 control program*, and a *memory manager*. The memory manager is configured using the Memory Manager Parameters screen and the System Memory Configuration screen, as described in the previous sections. The 386 control program has its own separate parameters, controlled from the 386 Control Program Parameters screen shown in figure 13-3.

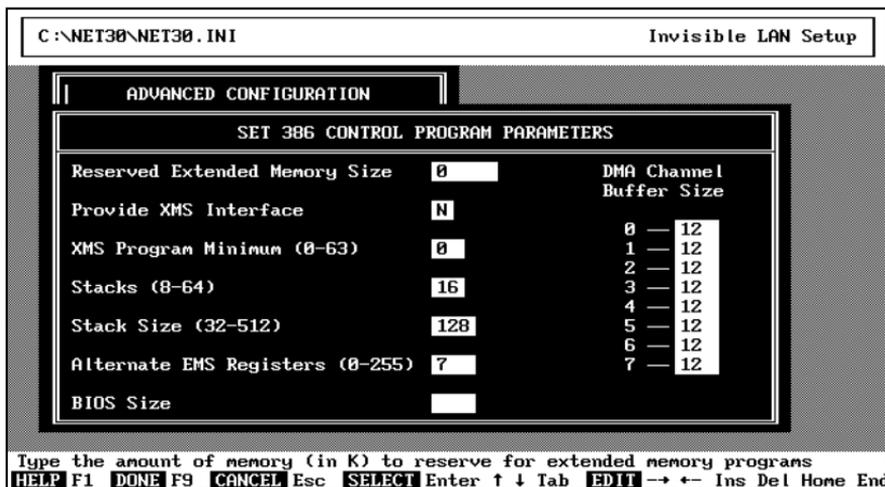


Figure 13-3. 386 Control Program Parameters Screen

The 386 control program parameters are fairly technical in nature. Except for allocating extended memory, you will probably not have to change these parameters from their default values.

Extended Memory Allocation

The **Reserved Extended Memory Size** parameter lets you reserve part of your extended memory for use by extended memory applications, such as the VDISK RAMDISK program and the Invisible LAN disk cache. The rest of your extended memory is converted to expanded memory and shadow RAM.

The reserved extended memory begins at segment address 10000, immediately above the first megabyte of memory. For example, suppose you have 3 megabytes of extended memory, occupying segments 10000 through 3FFFF. If you specify 1024 for **Reserved Extended Memory Size**, then segments 10000 through 1FFFF are reserved for extended memory applications, while segments 20000 through 3FFFF are converted into expanded memory and shadow RAM.

The default value of **Reserved Extended Memory Size** is 0.

Example — Suppose you want to use the Invisible LAN disk cache in extended memory, and you want to allocate 2048K (2 megabytes) for the cache data. You would enter 2048 for **Reserved Extended Memory Size**.

Technical note — The reserved extended memory works with programs that access extended memory through interrupt INT 15H functions 87H and 88H, or through DMA. It does not work with programs that actually switch the processor into protected mode.

High Memory Area XMS Interface

The **Provide XMS Interface** parameter allows programs to access the high memory area using the Microsoft XMS interface.

The *high memory area* is the first 64K of extended memory. This area of extended memory is special, because DOS programs can access it directly. Certain DOS applications are designed to make use of the high memory area.

Microsoft has developed an interface specification, called *XMS*, that allows programs to use the high memory area. If you enter **Y** for **Provide XMS Interface**, then Invisible RAM 386 installs an XMS interface; this allows programs to utilize the high memory area. If you enter **N**, the XMS interface is not installed.

Only one program at a time can use the high memory area. For this reason, the **XMS Program Minimum** lets you specify the minimum size, in kilobytes, that a program must have in order to use the high memory area. This allows you to exclude small programs from the high memory area, thereby ensuring that big programs can obtain access to the high memory area. This can optimize your use of memory.

The default value of **XMS Program Minimum** is 0.

Example — Suppose that you want to limit the use of the high memory area to programs that use at least 48K out of the 64K available. You would enter 48 for **XMS Program Minimum**.

Note — Microsoft distributes a device driver called HIMEM.SYS that provides access to the high memory area. You cannot use HIMEM.SYS with Invisible RAM 386. The **Provide XMS Interface** parameter is a *replacement* for the HIMEM.SYS device driver. The **XMS Program Minimum** parameter is equivalent to the “/HMAMIN=” parameter supported by HIMEM.SYS.

Note — When you include the XMS interface, Invisible RAM 386 automatically allocates 64K of extended memory. It is not necessary to use **Reserved Extended Memory Size** to allocate extended memory for use as the high memory area.

Note — The **Reserved Extended Memory Size** parameter is equivalent to the “/INT15=” parameter supported by some versions of HIMEM.SYS.

EMS Alternate Register Sets

The **Alternate EMS Registers** parameter selects the number of alternate register sets available to expanded memory applications.

Expanded memory works by mapping each 16K page frame into any 16K page of expanded memory (refer to chapter 11 for details). A *register set* is simply a complete set of mappings for all the page frames in the system. That is, a register set contains information that specifies which expanded memory page is mapped into each page frame.

There is always at least one register set. If there is more than one register set, then it is possible to quickly switch *all* the page mappings simply by switching from one register set to another. This is very useful in multitasking systems such as Desqview, because it allows each task to define its own expanded memory page mappings in a separate register set. Then you can switch quickly from one task to another simply by switching register sets.

The **Alternate EMS Registers** parameter specifies the number of register sets to allocate *in addition* to the standard register set. The default value is 7.

Example — To allocate 32 alternate register sets (for a total of 33 register sets), enter 32 for **Alternate EMS Registers**.

Interrupt Stack Frame Allocation

DOS versions 3.2 and later include a `STACKS` command that can be placed in the `CONFIG.SYS` file. The `STACKS` command allocates stack frames that are used to process hardware interrupts. Whenever a hardware interrupt occurs, DOS automatically switches from the application program's stack to one of the interrupt stack frames. The reason for this is that many application programs do not provide enough stack space to handle hardware interrupts.

The DOS `STACKS` command includes two parameters that let you specify the number of interrupt stack frames, and the size (in bytes) of each stack frame.

Invisible RAM 386 provides its own interrupt stack frames. When using Invisible RAM 386, you use its own stack frames instead of the stack frames provided by DOS. To do this, you specify the command `STACKS=0,0` in order to tell DOS not to allocate any interrupt stack frames.

The **Stacks** and **Stack Size** parameters replace the two parameters in the DOS `STACKS` command. **Stacks** specifies the number of interrupt stack frames to allocate. **Stack Size** specifies the size in bytes of each interrupt stack frame. The default values are 16 for **Stacks**, and 128 for **Stack Size**.

Example — To allocate 32 stack frames, each 256 bytes in size, enter 32 for **Stacks** and 256 for **Stack Size**.

Note — DOS versions 3.1 and earlier do not have a `STACKS` command, and do not allocate any interrupt stack frames. When using DOS version 3.1 or earlier, you would not place a `STACKS=0,0` command in the `CONFIG.SYS` file. Invisible RAM 386 always allocates its own interrupt stack frames, regardless of which version of DOS you use.

BIOS ROM configuration

The *system BIOS* is a ROM located on the computer's motherboard. It occupies the highest segment addresses in the computer, from `FFFF` on down. For example, a 64K BIOS ROM would occupy addresses `F000-FFFF`.

Many BIOS ROMs contain a setup program which is not necessary during normal computer operation. With the **BIOS Size** parameter, you can deactivate any unnecessary portion of the computer's BIOS, thereby freeing up more space for shadow RAM. Use **F5** and **F6** to select the desired BIOS size. If you leave the field blank, then no portion of the BIOS ROM is deactivated. The default is to leave the field blank.

Example — Suppose you have a 64K BIOS ROM, but the lower half (from segment `F000` to `F7FF`) is a setup program. In this case, you only need to retain 32K of the BIOS ROM. You would select 32 for **BIOS Size**.

DMA Data Buffer Allocation

DMA (*Direct Memory Access*) is a method for transferring data between the computer's memory and a peripheral device such as a disk drive, tape drive, or network.

The computer has several built-in *DMA channels* which are used to perform the data transfer. The number and capabilities of the DMA channels vary depending on the type of computer you have:

- On computers with the Industry Standard Architecture (ISA), the standard PC AT design, there are seven DMA channels. Channels 0 to 3 perform 8-bit data transfers, and are able to transfer up to 64K bytes in a single operation. Channels 5 to 7 perform 16-bit data transfers, and are able to transfer up to 128K bytes in a single operation. There is no channel 4.
- On computers with the Micro Channel Architecture (MCA), there are eight DMA channels, numbered 0 to 7. All channels can perform either 8-bit or 16-bit data transfers. When performing 8-bit transfers, they are able to transfer up to 64K bytes in a single operation. When performing 16-bit transfers, they are able to transfer up to 128K bytes in a single operation.

With Invisible RAM 386 installed, the DMA channels cannot transfer data directly between an application's memory and a peripheral device. Instead, Invisible RAM 386 uses a separate *data buffer*:

- When transferring data from an application's memory to a peripheral device, Invisible RAM 386 first copies the data from the application's memory to the data buffer, and then uses DMA to transfer the data from the buffer to the peripheral device.
- When transferring data from a peripheral device to an application's memory, Invisible RAM 386 first uses DMA to transfer the data from the peripheral device to the data buffer, and then copies the data from the buffer to the application's memory.

There is a separate data buffer for each DMA channel. The buffers are located in extended memory, so they do not take up any DOS memory. Each buffer must be big enough to hold the largest DMA transfer that is ever performed on the corresponding DMA channel. For example, the buffer for DMA channel 2 must be at least as large as the largest DMA transfer that is ever performed on DMA channel 2.

With the **DMA Channel Buffer Size** parameters, you can specify the size of each DMA channel's data buffer. There are eight data entry fields, so you can specify the buffer size for each channel. You can specify a buffer size ranging from 0K to 128K bytes.

The minimum, maximum, and default buffer sizes are determined as follows:

- For all channels, the minimum buffer size is 0K, with the following two exceptions: (1) For channel 2, the minimum buffer size is 9K. (2) If the hard disk uses DMA, the minimum buffer size for the hard disk's DMA channel is 64K.
- For channels 0 to 3 on ISA computers, the maximum buffer size is 64K.
- For channel 4 on ISA computers, the maximum buffer size is 0K.
- For channels 5 to 7 on ISA computers, and for channels 0 to 7 on MCA computers, the maximum buffer size is 128K.
- For all channels, the default buffer size is 12K, with the following two exceptions: (1) On ISA computers, the default buffer size for channel 4 is 0K. (2) If the hard disk uses DMA, the default buffer size for the hard disk's DMA channel is 64K.

With the **DMA Channel Buffer Size** parameters, if you specify a buffer size less than the permitted minimum, Invisible RAM 386 automatically uses the minimum. Similarly, if you specify a buffer size larger than the permitted maximum, Invisible RAM 386 automatically uses the maximum.

If you enter the value 12, Invisible RAM always uses the default value.

Example — You have a tape drive that uses DMA channel 3. The tape drive transfers data 48K bytes at a time. To specify a buffer size of 48K bytes for DMA channel 3, enter 48 for **DMA Channel Buffer Size 3**.

Example — You can free up memory by specifying a buffer size of 0K for unused DMA channels. This frees up 12K bytes per channel, which becomes available for use as expanded memory. For example, if you have no peripheral devices that use DMA channels 0, 5, 6, or 7, you can enter 0 for **DMA Channel Buffer Size 0**, **DMA Channel Buffer Size 5**, **DMA Channel Buffer Size 6**, and **DMA Channel Buffer Size 7**.

SUMMARY: The 386 Control Program Parameters screen lets you specify parameters for the 386 Control Program, which is a part of Invisible RAM 386. The parameters let you reserve extended memory, provide an XMS interface, allocate hardware interrupt stack frames and alternate EMS register sets, and configure the DMA buffers.

Part 4

Appendices and Index

Invisible LAN CONFIG.SYS Reference

Install the first Invisible LAN device driver (must be the first device driver installed in CONFIG.SYS):

```
DEVICE=CACHE30.SYS
```

Load device driver into shadow RAM:

```
DEVICE=LSHADOW.SYS [/A] filename [parameters]
```

<i>/A</i>	Alter BIOS memory variable
<i>filename</i>	Name of device driver to load
<i>parameters</i>	Parameters for device driver being loaded

Install the second Invisible LAN device driver (must be the last device driver installed in CONFIG.SYS):

```
DEVICE=N30DEV.SYS
```

Install device driver to disable high-resolution EGA or VGA graphics:

```
DEVICE=VGA2CGA.SYS
```

Index

3

- 386 control program
 - see* Invisible RAM 386
- 386 control program parameters, 13-18
 - BIOS size, 13-21
 - DMA buffer size, 13-22
 - EMS register sets, 13-20
 - reserved extended memory, 13-19
 - stack size, 13-21
 - stacks, 13-21
 - XMS interface, 13-19
- 386 virtual mode
 - see* Invisible RAM
 - see* Invisible RAM 386

A

- A386SRM.SYS device driver
 - see also* Invisible RAM
- access method (for printer)
 - see also* print spooler parameters
- ANSI.SYS DOS device driver, 2-9
- applications
 - see* TransBIOS parameters
- associativity
 - see* cache parameters
- AT/386 chipset, 1-7, 11-2, 11-6
 - configuring, 12-6
 - see also* Invisible RAM
- ATTRIB DOS command, 10-12
- audio
 - see* speaker

automatic print stream truncation
 see print stream truncation
automatic remapping, 8-5
 see also redirector parameters

B

background tasks
 see cache parameters
beeper
 see speaker
bell
 see speaker
BIOS, 11-4, 13-21
BIOS patch
 see cache parameters
blackout zone
 see Ultra Server
boot diskette image file
 see image file
buffer
 see disk buffer
 see video buffer
buffer size
 see print spooler parameters
 see Ultra Server parameters
buffers
 see server parameters
BUFFERS DOS command, 2-9, 2-11, 7-6, 13-9
bulletin board, 1-4, 1-13
 private message, 1-13
 public message, 1-13
 reply chain, 1-13
 sending to group, 1-13
 store-and-forward mail, 1-13

C

C386EMM.SYS device driver
 see also Invisible RAM

- see also* Invisible RAM 386
- cache, 1-4, 1-7 to 1-8
 - configuration, 8-23
 - data location, 2-6, 5-3, 10-4
 - data size, 2-7, 10-4
 - installing, 8-2
- CACHE command, 1-9
- cache parameters, 8-23
 - associativity, 8-25
 - background tasks, 8-26
 - BIOS patch, 8-26
 - code location, 8-25
 - data location, 6-5 to 6-6, 8-24, 10-4
 - data size, 8-25, 10-4
 - delay write time, 8-26
 - seek horizon, 8-26
 - transfer size, 8-26
 - write policy, 8-25
- CACHE.SYS device driver, 7-5
- CACHE/30
 - see* cache
- CACHE30.SYS device driver, 2-9, 7-5, A-1
- caching options
 - see* Ultra Server parameters
- channels
 - see* Ultra Server parameters
- CLRCMOS command, 12-6
- code location
 - see* cache parameters
 - see* memory manager parameters
 - see* TransBIOS parameters
- commands
 - see* NetBIOS parameters
- CONFIG.SYS, 2-9, 6-5, 7-5 to 7-8
- configuration (memory)
 - see* memory manager parameters
- configuration (network software)
 - advanced configuration, 8-1
- configuring
 - for large networks, 9-6
 - for maximum speed, 9-1
 - for minimum memory usage, 9-3
- conflicts (memory)
 - see* memory manager parameters
- conventional memory, 11-1, 11-3

CP386.SYS device driver
see also Invisible RAM
see also Invisible RAM 386

D

data location
see cache parameters
see easy configuration
see TransBIOS parameters

data size
see cache parameters
see easy configuration
see TransBIOS parameters

data terminal ready
see flow control

default RAM
see memory manager parameters

DEVICE DOS command, 7-5, 12-13

device driver, 7-5

direct memory access
see DMA

directory entries
see Ultra Server parameters

directory path, 8-21

disk buffer, 7-6

disk cache
see cache

diskless workstation, 1-8, 10-15

DMA (direct memory access), 13-22
see also 386 control program parameters

DMA channel
see TransBIOS parameters

DOS, 1-1

DOS 4.0
expanded memory support, 13-9

DOS 5.0
using high memory area, 6-5

DOS memory
see conventional memory

DOS=HIGH command, 6-5

DOS=UMB command, 6-2

drive letter, 7-6, 8-21

- drive mappings, 1-2
 - displaying (command), 3-3
 - initial, 8-21
 - DTR
 - see* flow control
-

E

- easy cache configuration
 - data location, 10-4
 - data size, 10-4
- easy configuration
 - data location, 2-6
 - data size, 2-7
 - network hardware type, 2-5
 - network printer definition, 2-6
 - program selection, 2-5 to 2-6
 - software version, 2-4
 - user name, 2-4
 - using expanded memory for network software, 2-7
 - using shadow RAM for network software, 2-7
 - Windows support, 2-6, 5-2, 10-3
- EGA
 - see* enhanced graphics adapter
- electronic mail
 - see* mail
- EMS
 - see* expanded memory
- EMS register sets
 - see* 386 control program parameters
- enhanced EMS, 11-5, 12-14
- enhanced graphics adapter, 11-4, 12-15, 13-7 to 13-8, 13-14 to 13-15, 13-17
 - see also* high-resolution graphics
- EVERYONE
 - see also* group
- exclude
 - see* memory manager parameters
- expanded memory, 1-6, 11-1, 11-4, 13-2, 13-4, 13-15
 - using for cache, 2-6, 8-24, 10-4
 - using for mail, 8-30
 - using for network software, 2-7, 6-3
 - using for redirector, 8-19
 - using for server, 8-11

- using for TransBIOS, 8-5
- using with Invisible LAN For Windows, 10-5
- with Invisible EMS, 13-4
- with Invisible RAM 386, 13-4
- with NEAT chipset, 12-2 to 12-3, 12-5, 13-4
- with SCAT chipset, 12-8, 13-4
- expanded memory manager, 11-5
 - see also* Generic EMS
 - see also* Invisible RAM
- extended emulation
 - see* memory manager parameters
- extended memory, 11-4, 12-11
 - converting to conventional memory, 11-4, 12-10, 13-19
 - converting to expanded memory, 11-4, 12-10, 13-19
 - converting to shadow RAM, 11-4, 12-10, 13-19
 - using for cache, 2-6, 6-5 to 6-6, 8-24, 10-4
 - using for Ultra Server, 8-33
 - with Invisible RAM 386, 13-19
 - with NEAT chipset, 12-4 to 12-5

F

- FASTOPEN DOS command, 13-9
- FCBs
 - see* redirector parameters
- FCBS DOS command, 2-9, 7-7
- file buffer
 - see* disk buffer
- file control block, 7-7
- file handle, 7-7
- file locks
 - see* record locks
- file server
 - see* server
- files
 - see* redirector parameters
 - see* server parameters
 - see* Ultra Server parameters
- FILES DOS command, 2-9, 2-11, 7-7, 8-19
- filesystem
 - see* SHARE parameters
- flow control (for serial printer)
 - see also* print spooler parameters

form feed (for printer)
see also print spooler parameters
frame address
see memory manager parameters
frame size
see memory manager parameters
frontfill (memory), 1-6, 11-1, 12-14, 13-7
frontfill size
see memory manager parameters

G

GEMSSRM.SYS device driver
see also Invisible RAM
general parameters, 8-2
 network hardware type, 8-2
 program selection, 8-2 to 8-3
 software version, 8-2
 user name, 8-2
 Windows support, 5-2, 8-3, 10-3
Generic EMS, 11-3, 11-6
 configuring, 12-12, 12-14
 underlying expanded memory manager, 12-12, 13-14
 see also Invisible RAM
graphics
 see high-resolution graphics
group
 use in bulletin board, 1-13

H

handle
 see file handle
handles
 see memory manager parameters
hardware type
 see easy configuration
 see general parameters
help, 3-4, 7-2

- high memory area (XMS), 13-19
 - using for DOS 5.0, 6-5
- high RAM, 11-6, 13-2, 13-17
 - see also* memory manager parameters
- high-resolution graphics, 11-3, 13-8, 13-14 to 13-15
 - conflict with frontfill, 12-15
 - see also* memory manager parameters
- HIMEM.SYS, 6-5
 - configuring for cache, 6-6
- hot key, 1-9
- hyper-extended memory
 - using for Ultra Server, 8-33

I/O port

- see* memory manager parameters
- see* TransBIOS parameters

IEMSEMM.SYS device driver, 12-12

- see also* Invisible RAM

include

- see* memory manager parameters

initialization file, 2-4, 4-2, 7-1

- creating, 2-4, 4-2, 7-2 to 7-3
- modifying, 4-1, 7-3
- specifying, 4-2, 7-2

installation

- see* setup program

installing Invisible LAN For Windows

- advanced installation, 10-1
- components of Invisible LAN, 1-4, 1-11, 10-1
- configuring Invisible LAN, 10-3
- configuring Windows, 10-6
- copying Windows to file server, 10-12
- disk cache configuration, 10-4
- diskless workstation, 10-15
- enabling Windows support, 10-3
- Invisible LAN icon installation, 10-8, 10-10
- Invisible LAN wallpaper installation, 10-16
- network driver installation, 10-8, 10-10
- printer drivers, 10-16
- quick-start, 5-1
- quick-start Invisible LAN icon installation, 5-3

- quick-start network driver installation, 5-3
- quick-start network installation, 5-2
- quick-start Windows installation, 5-2
- serial network printers, 10-14
- server configuration, 10-6
- setting up Windows on workstation, 10-13
- starting the software, 5-4
- TransBIOS ROM, 10-15
- using expanded memory, 10-5
- using Invisible RAM, 10-5
- using shadow RAM, 10-5
- Windows installation (shared), 10-11
- Windows installation (stand-alone), 10-7
- WINSTART.BAT, 10-15
- Invisible EMS, 1-7, 11-2, 11-6
 - configuring, 12-12, 13-11, 13-15 to 13-16
 - see also* Invisible RAM
- Invisible LAN icon
 - installing, 5-3, 10-8, 10-10
- Invisible RAM, 1-4, 1-6 to 1-7, 11-1
 - expanded memory, 11-1
 - for 386 virtual mode, 11-2, 13-3, 13-6, 13-9
 - for AT/386 chipset, 11-2, 13-3, 13-5
 - for Generic EMS, 11-3, 13-3, 13-5
 - for Invisible EMS, 11-2, 13-3, 13-6, 13-9 to 13-11
 - for NEAT chipset, 11-2, 13-2, 13-9
 - for PEAK chipset, 11-3, 13-3, 13-5
 - for SCAT chipset, 11-3, 13-2, 13-6, 13-9
 - installing in setup program, 13-3
 - using with Invisible LAN For Windows, 10-5
 - see also* 386 control program parameters
 - see also* expanded memory
 - see also* frontfill (memory)
 - see also* memory manager parameters
 - see also* shadow RAM
- Invisible RAM 386, 11-2, 11-4, 11-6, 13-3, 13-6, 13-9, 13-18
 - 386 control program, 13-18
 - 386 virtual mode, 12-10
- Invisible RAM parameters
 - see* 386 control program parameters
 - see* memory manager parameters
- IRQ level
 - see* TransBIOS parameters
- ISNET.DRV, 10-9 to 10-10, 10-12
- ISNET.HLP, 10-9 to 10-10, 10-12

L

- LASTDRIVE DOS command, 2-9, 2-11, 7-6
 - local disk, 1-3
 - local printer, 1-3
 - locks
 - see* record locks
 - LSHADOW command, 11-6, 13-2
 - LSHADOW.SYS device driver, 11-6, 13-2, A-1
-

M

- mail, 1-4, 1-9
 - configuration, 8-29
 - installing, 2-5, 8-3
- mail parameters, 8-29
 - using expanded memory, 8-30
 - using shadow RAM, 8-30
- main menu
 - see* setup program
- manager, 1-4, 1-10
 - automatic print stream truncation, 1-10
 - drive mappings, 1-10
 - group definitions and security, 1-10
 - main menu, 1-10
 - miscellaneous security, 1-11
 - poll network, select machine, 1-10
 - print queue contents, 1-10
 - print spooler status, 1-10
 - printer mappings, 1-10
 - server usage statistics, 1-11
 - shortname definitions and security, 1-10
 - user definitions and security, 1-10
- MAP command, 1-9, 3-4
 - connecting to network disk without shortname, 3-2
 - connecting to network printer, 3-2
 - disconnecting from network disk, 3-3
 - disconnecting from network printer, 3-3
 - displaying network resources, 3-3
- memory (general information), 11-3
- memory address
 - see* TransBIOS parameters

- memory manager
 - see* Invisible RAM
- memory manager parameters, 13-3, 13-11
 - code location, 13-8
 - configuration, 13-4
 - contexts, 13-9
 - default RAM, 13-14
 - exclude, 13-13
 - extended emulation, 13-10
 - frame address, 13-6
 - frame size, 13-6
 - frontfill size, 13-7
 - handles, 13-8
 - high RAM, 13-17
 - I/O port, 13-11
 - include, 13-15
 - initial VGA state, 13-8
 - memory manager, 13-4
 - non-standard page, 13-9
 - ROM copy, 13-17
 - shadow RAM, 13-16
 - test level, 13-10
 - see also* Invisible RAM
- memory manager selection
 - see* general parameters
- memory relocation
 - see* NEAT chipset
- menu system
 - see* setup program
- MODE DOS command, 8-16
 - configuring the serial port, 8-16
 - redirecting parallel printer output, 8-16

N

- N30DEV.SYS device driver, 2-9, 7-5, A-1
- names
 - see* NetBIOS parameters
 - see* TransBIOS parameters
- NEAT chipset, 1-7, 11-2, 11-6
 - configuring, 12-1
 - expanded memory controller, 11-5, 12-3 to 12-5
 - memory relocation, 12-3 to 12-5

- RAS timeout, 12-3 to 12-5
- resetting configuration, 12-5
 - see also* Invisible RAM
- NEATEMM.SYS device driver
 - see also* Invisible RAM
- NET30 command, 3-1, 4-2, 7-2
- NET30.INI
 - see* initialization file
- NET30APP.EXE, 5-3, 10-8 to 10-10, 10-12
- NetBIOS, 1-4, 1-14, 8-8
 - ROM, 8-9, 13-17
- NetBIOS parameters, 8-8
 - commands, 8-10
 - names, 8-9
 - sessions, 8-9
- NETCLOCK command, 1-9
- network disk, 1-3
 - connecting, 3-2
 - disconnecting, 3-3
- network hardware type
 - see* easy configuration
 - see* general parameters
- network operating system, 1-1
- network printer, 1-3
 - connecting, 3-2
 - disconnecting, 3-3
 - see also* printer number
- network printer #1, 2-6, 8-15
 - see also* miscellaneous security
- network printer #2, 2-6, 8-15
 - see also* miscellaneous security
- network printer definition
 - see* easy configuration
 - see* print spooler parameters
- network protocol, 1-14, 8-8
- non-standard page
 - see* memory manager parameters
- NW command, 1-9

O

- OEMSETUP.INF, 10-9, 10-13

- optimizing
 - for large networks, 9-6
 - for maximum speed, 9-1
 - for minimum memory usage, 9-3
-

P

- packet size
 - see* TransBIOS parameters
- page (memory)
 - see* expanded memory
- page frame, 11-5 to 11-6, 13-2, 13-11, 13-20
- paged EMS memory
 - see* expanded memory
- parameters, 7-1, 8-1
 - 386 control program, 13-18
 - cache, 8-23
 - initial drive mappings, 8-21
 - initial printer mappings, 8-22
 - mail, 8-29
 - memory manager, 13-3, 13-11
 - NetBIOS, 8-8
 - print spooler, 8-15
 - redirector, 8-18
 - server, 8-10
 - SHARE, 8-27
 - TransBIOS, 8-3
 - Ultra Server, 8-30
- password
 - see also* redirector parameters
- PATH DOS command, 2-11
- PEAK chipset, 1-7, 11-3, 11-6
 - configuring, 12-10
- polls
 - see* TransBIOS parameters
- PQ command, 1-9
- print device
 - see* device name
- print queue, 1-5 to 1-6
 - see also* print spooler
- print spooler, 1-4 to 1-6
 - configuration, 8-15
 - see also* flow control (for serial printer)

- see also* form feed (for printer)
 - see also* print queue
- print spooler parameters, 8-15
 - access method, 8-16
 - break page, 8-16
 - buffer size, 8-17
 - direct print, 8-17
 - initial flow control state, 8-16
 - initial form feed state, 8-16
 - network printer definition, 2-6, 8-15
 - print priority, 8-17
 - spool directory, 8-17
 - spool files, 8-17
- print stream truncation
 - see also* redirector parameters
- printer mappings, 1-2
 - displaying (command), 3-3
 - initial, 8-22
- printer number, 8-22
- private files
 - see* server parameters
 - see* Ultra Server parameters
- program selection
 - see* easy configuration
 - see* general parameters
- protected mode, 11-4
- protocol
 - see* network protocol
- PSETUP command, 1-9

R

- record locks
 - see* server parameters
 - see* SHARE parameters
 - see* Ultra Server parameters
- redirector, 1-3, 1-5, 7-6
 - configuration, 8-18
 - installing, 8-3
- redirector parameters, 8-18
 - automatic remapping, 8-21
 - buffer size, 8-20
 - buffers, 8-20

- FCBs, 8-19
- files, 8-19
- initial audio state, 8-20
- initial print stream truncation, 8-20
- password, 8-20
- print buffer size, 8-20
- sessions, 8-19
- using expanded memory, 8-19
- using shadow RAM, 8-18
- wait for server, 8-21
- remote program load image
 - see* server parameters
- remote program load server, 1-4, 1-8
 - configuration, 8-12
 - see also* image file
- reserved extended memory
 - see* 386 control program parameters
 - see* easy memory configuration
- resident program
 - see* memory-resident program
- responses
 - see* TransBIOS parameters
- retry limit
 - see* TransBIOS parameters
- RIPLMEM, 10-15
- ROM, 11-4
- ROM copy, 13-2, 13-17
 - see also* memory manager parameters
- RPL
 - see* remote program load server

S

- SCAT chipset, 1-7, 11-3, 11-6
 - configuring, 12-7
 - expanded memory controller, 11-5
 - extended memory boundary, 12-9
 - memory relocation, 12-8 to 12-9
- segment (memory), 11-3
- server, 1-4 to 1-5
 - configuration, 8-10
 - installing, 2-6, 8-2

- server parameters, 8-10
 - audit trail, 8-13
 - audit trail size, 8-13
 - block ctrl-alt-del, 8-13
 - buffers, 8-11
 - files, 8-12
 - local superuser, 8-13
 - private files, 8-12, 10-6
 - record locks, 8-12
 - remote program load image, 8-12
 - sessions, 8-11
 - UPS discharge time, 8-14
 - UPS I/O port, 8-13
 - UPS monitor, 8-13
 - UPS power fail, 8-13
 - UPS recharge time, 8-14
 - using expanded memory, 8-11
 - using shadow RAM, 8-11
 - write pipeline, 8-14
 - see also* print spooler parameters
 - see also* Ultra Server parameters
- setup program, 1-5, 2-1
 - 386 control program parameters, 13-18
 - advanced configuration, 7-3, 8-1
 - cache parameters, 8-24, 10-4
 - copying files, 2-2
 - easy cache configuration, 10-4
 - easy configuration, 2-4, 5-2, 10-3
 - general parameters, 5-2, 8-2, 10-3
 - initial drive mappings, 8-21
 - initial printer mappings, 8-22
 - loading initialization file, 2-4, 4-1 to 4-3, 7-3
 - mail parameters, 8-29
 - memory manager parameters, 13-3
 - modifying CONFIG.SYS, 2-9, 7-5
 - NetBIOS parameters, 8-8
 - print spooler parameters, 8-15
 - redirector parameters, 8-18
 - saving initialization file, 2-8, 4-1, 4-3, 7-4
 - server parameters, 8-10
 - SHARE parameters, 8-27
 - starting, 2-1, 4-1, 7-2
 - system memory configuration, 13-11
 - TransBIOS parameters, 8-3
 - Ultra Server parameters, 8-30

- SETUP30 command
 - see* setup program
- SHADOW command
 - chipset information, 12-2, 12-6 to 12-7, 12-10
 - expanded memory information, 12-13
 - extended memory information, 12-11
 - memory map, 12-13, 13-13
- shadow RAM, 1-6, 11-1, 11-6, 13-2, 13-4, 13-16
 - loading device drivers, 11-6
 - loading memory-resident programs, 11-6
 - using for cache, 8-25
 - using for mail, 8-30
 - using for memory manager, 13-8
 - using for network software, 2-7, 6-1
 - using for redirector, 8-18
 - using for server, 8-11
 - using for SHARE, 8-27
 - using for TransBIOS, 8-5
 - using with Invisible LAN For Windows, 10-5
 - with AT/386 chipset, 13-4
 - with Generic EMS, 13-4
 - with Invisible EMS, 13-4
 - with Invisible RAM 386, 13-4
 - with NEAT chipset, 12-2 to 12-5, 13-4
 - with PEAK chipset, 13-4
 - with SCAT chipset, 12-8, 13-4
 - see also* memory manager parameters
- shadow RAM information
 - see* SHADOW command
- SHARE command, 2-2
 - configuration, 8-27
 - installing, 8-2
- SHARE parameters, 8-27
 - filespace, 8-28
 - record locks, 8-28
 - using shadow RAM, 8-27
- shortname
 - use in initial drive mappings, 8-21
- software version
 - see* easy configuration
 - see* general parameters
- speaker
 - see also* redirector parameters
- spool directory
 - see* print spooler parameters

spool files
 see print spooler parameters
spooler
 see print spooler
stack size
 see 386 control program parameters
stacks
 see 386 control program parameters
STACKS DOS command, 2-9, 2-11, 7-8, 13-21
standard EMS, 11-5, 12-14, 13-6
starting Invisible LAN, 3-1, 4-2, 7-2
SYSTEM.INI, 10-10, 10-14

T

test level
 see memory manager parameters
track buffer
 see cache
TransBIOS, 1-4, 1-14, 8-3, 8-8
 ROM, 8-4, 10-15, 13-17
TransBIOS parameters, 8-3
 applications, 8-6
 code location, 8-5
 data location, 8-5
 data size, 8-5
 DMA channel, 8-7
 I/O port, 8-7
 IRQ level, 8-7
 memory address, 8-7
 names, 8-6
 packet size, 8-7
 polls, 8-6
 responses, 8-6
 retry limit, 8-5
 sessions, 8-6
transfer size
 see cache parameters
TRUNCATE command, 1-9

U

- Ultra Server, 1-4, 1-14
 - blackout zone, 8-36
 - configuration, 8-30
 - installing, 2-5, 8-2
- Ultra Server parameters, 8-30
 - buffer size, 8-32
 - channels, 8-32
 - directory entries, 8-32
 - extended memory, 8-34
 - files, 8-31
 - hyper-extended size, 8-35
 - hyper-extended start, 8-35
 - private files, 8-32
 - record locks, 8-32
 - see also* server parameters
- underlying expanded memory manager
 - see* Generic EMS
- UPS discharge time
 - see* server parameters
- UPS I/O port
 - see* server parameters
- UPS monitor
 - see* server parameters
- UPS power fail
 - see* server parameters
- UPS recharge time
 - see* server parameters
- user name
 - see* easy configuration
 - see* general parameters

V

- VDISK DOS device driver, 13-19
- VDISK.SYS DOS device driver, 13-9 to 13-10
- version
 - see* easy configuration
 - see* general parameters
- VGA
 - see* video graphics array

VGA2CGA.SYS device driver, 12-15, A-1
VGAOFF command, 13-14
VGAON command, 13-14
video
 see enhanced graphics adapter
 see high-resolution graphics
 see video graphics array
video buffer, 11-3
video graphics array, 11-4, 12-15, 13-7 to 13-8, 13-13 to 13-15, 13-17
 see also high-resolution graphics
video RAM
 see video buffer

W

wait option
 see also redirector parameters
WALL30.BMP, 10-12, 10-16
wallpaper, 10-16
WIN.INI, 10-10
window (memory)
 see expanded memory
Windows Control Panel
 printer drivers, 10-16
 wallpaper, 10-16
Windows Network Manager, 1-4, 1-11
 about box, 1-12
 bulletin board, 1-12
 drive mappings, 1-11
 help, 1-12
 logon, 1-11
 mail, 1-12
 polling, 1-11
 preferences, 1-11
 print queue contents, 1-12
 print spooler status, 1-12
 printer mappings, 1-11
 security, 1-12
 server usage statistics, 1-12
Windows printer drivers, 10-16
WINSTART.BAT, 10-15
WINSTRT command, 10-15

write pipeline
 see server parameters
write policy
 see cache parameters
write-back
 see cache parameters
write-through
 see cache parameters

X

XMS
 using for cache, 2-6, 6-6, 8-24, 10-4
XMS interface
 see 386 control program parameters
XON/XOFF
 see flow control

