appendix A Technical Information

Appendix Overview

This appendix describes the architecture of the NetWare DOS Requester and its Virtual Loadable Modules (VLMs). Also included is a discussion on how networking tasks work with Windows.

The following topics are in this appendix:

Virtual Loadable Modules (VLMs)

The NetWare DOS Requester is composed of several individual functional modules called Virtual Loadable Modules (VLMs).

A VLM is a modular executable program with a set of logically grouped features or APIs. For example, transport related functions (send packets, receive packets, etc.) fit logically into a VLM.

The two types of VLMs are these:

- Child VLMs
- Multiplexors

Child VLMs

Child VLMs handle a particular implementation of a logical grouping of functionality. For example, each of the NetWare server types has its own child VLM:

- BIND.VLM for bindery servers before NetWare v4.0.
- NDS.VLM for NDS, NetWare v4.0 servers.
- PNW.VLM for Personal NetWare (NetWare desktop) servers.

Various implementations of transport protocols also have their individual child VLMs. For example, IPXNCP.VLM handles IPX services. Other transport protocols may be defined in the future.

Multiplexors

A multiplexor is a special VLM that routes calls to the proper child VLM. Requester multiplexors can be considered parent VLMs, ensuring that requests to their child VLMs reach the apporopriate VLM module.

For example, NWP .VLM is the multiplexor that handles calls to and from the child VLMs, BIND.VLM, NDS.VLM, and PNW .VLM. Similarly , TRAN.VLM is the multiplexor that coordinates communication at the transport layer when its child VLMs (IPXNCP .VLM and other transport protocols) are concurrently loaded.

Table A-1 describes the individual components of the DOS workstation environment.

Table A-1
DOS Requester modules

Module	Description
VLM.EXE and asscociated VLMs	Provides network services for file and print redirection, as well as its own APIs for connection maintenance and other NetWare-specific support.
IPXODI	Delivers requests and replies between a workstation and the network. Also handles packet sequencing and acknowledgment for the workstation-to-server connection (for example, SPX). Takes workstation requests that the DOS Requester has determined are for the network, “packages” them with transmission information (such as their destination), and hands them to the LSL.
LSL	Puts the packaged requests from IPXODI into the proper format for transmission on the particular physical network that the workstation is running on. Also takes replies for the workstation from the network (via the LAN driver), removes the network-specific information it has added, and passes the reply to IPX.
LAN driver	Takes requests from the LSL and sends them to the network. Also receives replies from the network for the workstation and passes them to the LSL.

Table A-1 ***continued***

DOS Requester modules

Module	Description
TBMI2	Handles direct calls to IPX from switched peer-to-peer applications. Also takes replies from IPX and gives them to the correct switched application.
NETX.VLM	Provides backwards compatibility with the older NETX shell.

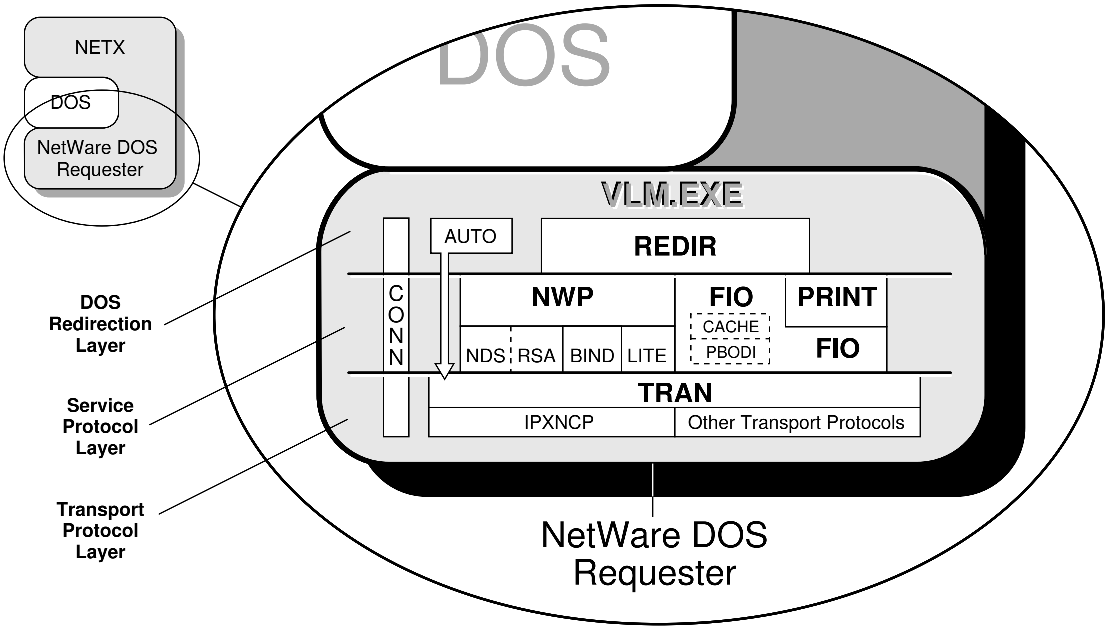
NetWare DOS Requester

The DOS Requester provides various services grouped into three layers:

- DOS Redirection Layer
- Service Protocol Layer
- Transport Protocol Layer

Figure A-1 shows how these layers and modules fit together. Individual pieces are discussed in further detail the sections that follow .

Figure A-1
NetW are DOS Requester layers and modules



DOS Redirection Layer

This layer provides DOS file ser vices through the DOS redirector; its VLM is REDIR.VLM. REDIR performs all the DOS-specific callouts.

Previously , NETX provided its own file ser vices; in NetW are v4.0, however , the DOS Requester uses the DOS redirector for most file ser vices.

Service Protocol Layer

This layer consists of the following parallel services.

NetWare Services Provided by NWP.VLM

NWP, the NetWare protocol multiplexor, handles each particular network implementation through the child VLMs:

- NDS, NetWare Directory Services (NetWare v4.0)
- BIND, bindery (NetWare prior to v4.0)
- PNW, NetWare desktop (Personal NetWare)

The NWP services also include establishing and destroying connections, logins, and logouts, and broadcast messages.

Security Services Provided by RSA.VLM

RSA (Rivest, Shamir, and Adleman, developers of this particular public key encryption system) provides system level background authentication for the workstation.

NetWare 4.0 uses this encryption system to strengthen user authentication and access control functions.

File Services Provided by FIO.VLM

FIO, the file input/output module, is a separate-but-parallel piece that implements a basic file-transfer protocol. This includes cached or non-cached read/write, burst mode-based read/write, and large internet packet read/write.

The Packet Burst protocol in FIO.VLM is based on ODI and is referred to as PBODI. This functionality allows users to choose performance over memory use and vice versa.

Print Services Provided by PRINT.VLM

The PRINT module provides printing services using the FIO module for its file writes. Print redirection is handled in the following ways, depending on entries in the NET.CFG file:

- Noncached
- Cached
- Redirected via Packet Burst protocol to the server

Printer redirection using FIO is faster.

Transport Protocol Layer

The transport layer is responsible for maintaining connections, providing packet transmissions between connections, and performing other transport-specific functions.

The IPXNCP module uses IPX as a transport mechanism.

The AUTO.VLM module reconnects a client to a server and rebuilds the client's environment—excluding file specific items—to its pre-connection-loss state.

VLM Manager

The VLM Manager (VLM.EXE) controls communication and memory issues between the individual VLMs, multiplexors, and childs VLMs.

The VLM Manager decides whether a given VLM uses expanded memory, extended memory, conventional memory, or any other memory type supported, without affecting the individual VLMs.

The user can override the VLM.EXE defaults with command line options or entries in the NET.CFG file.

Syntax

VLM[*option***]**

Table A-2
VLM options

Option	Syntax	Description
Help	/?	Displays help screen
Unload	/U	Unloads VLM.EXE from memory
Configuration File	/C	Indicates alternate filename for configuration information. /C = [*path*\\]*filename*
Conventional Memory	/MC	Uses conventional memory when loading VLM.EXE
Extended Memory	/MX	Uses extended memory when loading VLM.EXE
Expanded Memory	/ME	Uses expanded memory when loading VLM.EXE
Diagnostics	/D	Displays VLM.EXE file diagnostics

Connection Manager

The Connection Table Manager (CONN.VLM) spans all the layers of the DOS Requester architecture.

CONN allows a workstation running the DOS Requester to establish a configurable number of connections with multiple NetWare servers, by using a NET.CFG parameter. See “CONNECTIONS” on page -115 for a description this parameter.

NETX.VLM

The NETX.VLM provides backward compatibility with NETX and other older versions of the shell.

NET.CFG file

NET.CFG is a specialized text file that you create with any ASCII text editor and include on a workstation boot diskette with any other necessary boot files. The NET.CFG file replaces the SHELL.CFG file.

Like the DOS CONFIG.SYS file, the NET.CFG file contains configuration values of the LAN drivers and DOS Requester that are read and interpreted when your workstation starts up. These values adjust the operating parameters of the NetWare DOS Requester, IPX, and other workstation software.

You may want to change the values of certain DOS Requester parameters to modify the workstation's reaction in certain routines and processes. For a comprehensive list of NET.CFG options see "NET.CFG File Parameters" on page 89.

Applications such as database, multitasking, or NETBIOS (involved in peer-to-peer communications or distributed processing) may require parameter values different from the default values to function properly on the network.

To find out which parameters you might need to modify, consult the setup reference for each application used on your network. You might also find that printing, file retrieval, and other network problems can be solved by adjusting workstation parameters.

To create a NET.CFG file and modify the various parameters, see "Configuring Your Workstation" on page 5.

IPXODI

Although the NetWare DOS Requester intercepts and prepares requests for network transmission, the actual delivery is made by the IPXODI.COM program (Inter-Packet Exchange Open Data-Link Interface). IPXODI attaches a header to each data packet. The header specifies necessary information for targeted network delivery, announcing where the packet came from, where it's going, and what happens after delivery.

The targeting ability of IPXODI is important, but by itself does not guarantee successful delivery of a data transmission.

IPXODI transmits data packets as datagrams (self-contained packages that move independently from source to destination) and, thus, can deliver the packets only on a best-effort basis. Delivery is guaranteed only when using the SPX protocol.

SPX

SPX (Sequenced Packet Exchange) is a protocol within IPXODI. SPX is derived from Novell's IPX using the Xerox Sequenced Packet Protocol. It enhances the IPX protocol by supervising data sent out across the network. SPX verifies and acknowledges successful packet delivery to any network destination by requesting a verification that the data was received.

Within this verification must be a value that matches the value calculated from the data before transmission. So SPX ensures not only that the data packet arrived, but that it arrived intact. It can verify one transmission, or each transmission in a series.

SPX can track data transmissions consisting of a series of separate packets. If an acknowledgment request brings no response within a specified time, SPX retransmits it. After a reasonable number of retransmissions fail to return a positive acknowledgment, SPX assumes the connection has failed and warns the operator of the failure.

LSL

The LSL (Link Support Layer) is an implementation of the Open Data-Link Interface specification. The LSL serves as an intermediary between the LAN drivers and communication protocols—like IPXODI, AFP, or TCP/IP.

The LSL allows one network board to service several communications protocol stacks. The LSL also allows several network boards to service the same protocol stack.

LAN Driver

Every transmission prepared by the DOS Requester must pass through the network board connecting the workstation to the network. The LAN driver makes the connection between the physical network board and the logical routines and programs that use it.

A LAN driver is linked both to the specific network board and to the IPX program. Through the LAN driver, IPX can transmit requests regardless of the type of interface board or network protocol. In effect, different drivers allow the same operating system to use many different kinds of boards. This gives an installer flexibility when choosing the most effective network topology.

TBMI (Task-Switched Buffer Manager)

The Task-Switched Buffer Manager for IPX/SPX (TBMI2) allows IPX and SPX programs to work in a multitasking environment (such as Microsoft Windows 3.1, DR DOS 6.0, or MS-DOS 5.0).

The multitasking environment in real and standard modes allows application task switching (swapping). Each application runs in a separate DOS session (DOS Prompt) in available memory.

Global memory, available to all, contains drivers and TSRs such as COMMAND.COM and, if you are running NetWare, IPXODI.COM and the DOS Requester. Local memory, available to only current DOS session or prompt, contains the application and application data.

The multitasking environment switches from one DOS application to another by moving the contents of the current DOS session from conventional memory to disk, and then loading the contents of the new DOS session into conventional memory.

Only the local memory is switched; the global memory with its drivers and TSRs stays intact and is used with the new session. This means that separate local memory segments exist, one for each DOS session, while only one global memory segment exists. You don't need to use TBMI2 if

- The application uses the NetWare DOS Requester to access IPX or SPX;

- You will not switch between sessions.

You must use TBMI2 if

- You will switch between sessions; and

- The application bypasses the NetWare DOS Requester and accesses IPX or SPX directly.

If your application requires TBMI2 and you don't use it, the session will fail when the DOS session is swapped out or task-switched.

Note

If you aren't sure your application needs TBMI2, run TBMI2; it uses only a smal amount of memory. After running the application for a period of time, enter th command TBMI2 /D, and look at the number in the "Far Call Usage field." If this number is zero, then your application is not using TBMI2; you can run y application without it.

You must update to IPXODI before you can use TBMI2.

Windows Technical Overview

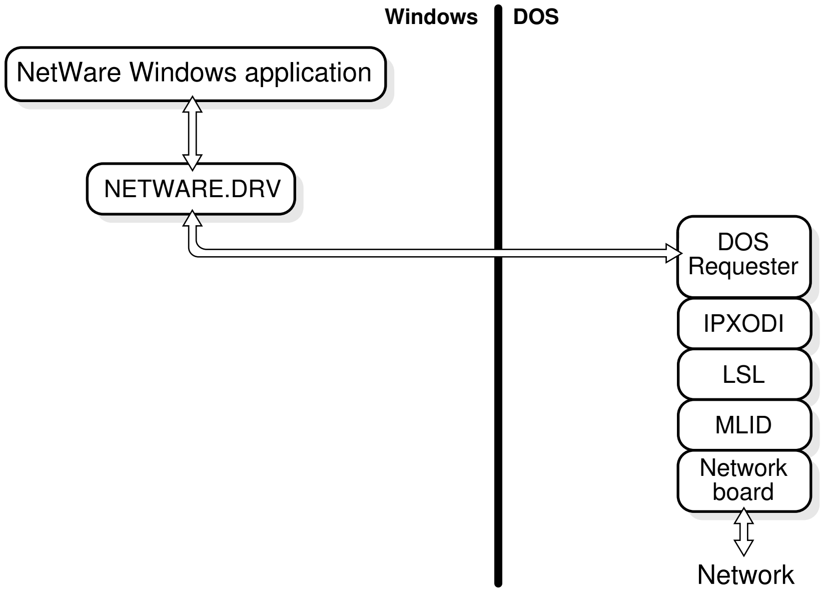
This section contains conceptual information about the NetWare Workstation for Windows components. The following topics are included.

Accessing the Network from a Windows Application

Figure A-2 illustrates which NetWare components are used to access the network from Windows. For example, when you use the Drives tools to connect a network drive, your workstation uses the components shown below.

The NetWare Windows driver, NETWARE.DRV, translates between Window and DOS.

Figure A-2
Accessing the Network from a Windows application

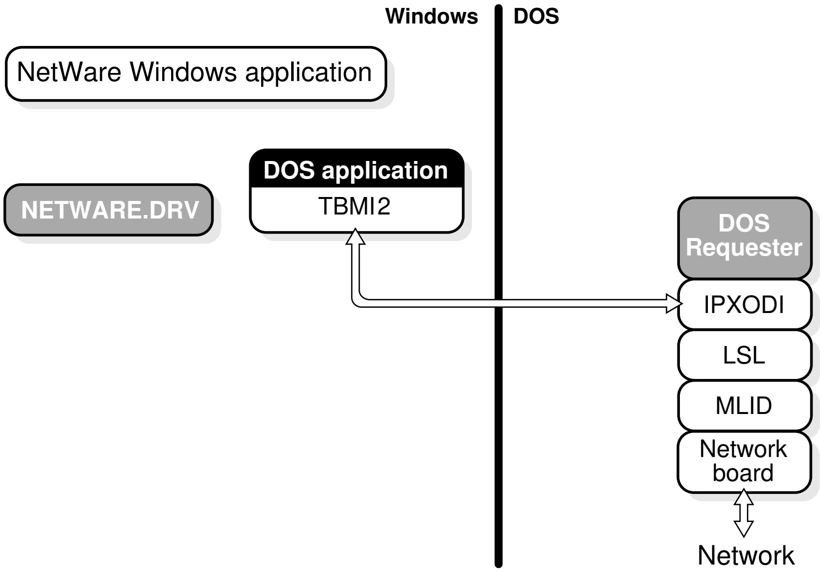


Running DOS Applications in Real or Standard Mode

Figure A-3 illustrates which NetW are components are used to run DOS applications in Real or Standard mode. For example, if you are running W indows on a 286 machine and you choose the DOS prompt icon to go to a DOS window and check your email, your workstation uses the components shown below .

NetW are's task-switching file, TBMI2, synchronizes network calls and responses for DOS sessions.

Figure A-3
Running a DOS application in Real or Standard mode

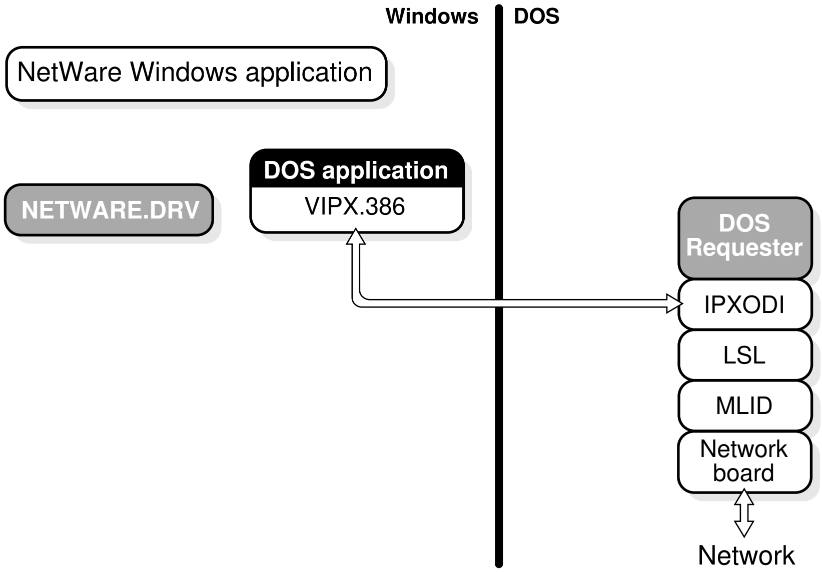


Running DOS Applications in Enhanced Mode

Figure A-4 illustrates which NetW are components are used to run DOS applications in Enhanced mode. For example, if you are running W indows in Enhanced mode and you choose the DOS prompt icon to go to a DOS window and check your email, your workstation uses the components shown below .

NetW are's task-switching file for Enhanced mode, VIPX.386, synchronizes network calls and responses for DOS sessions.

Figure A-4
Running a DOS application in Enhanced mode

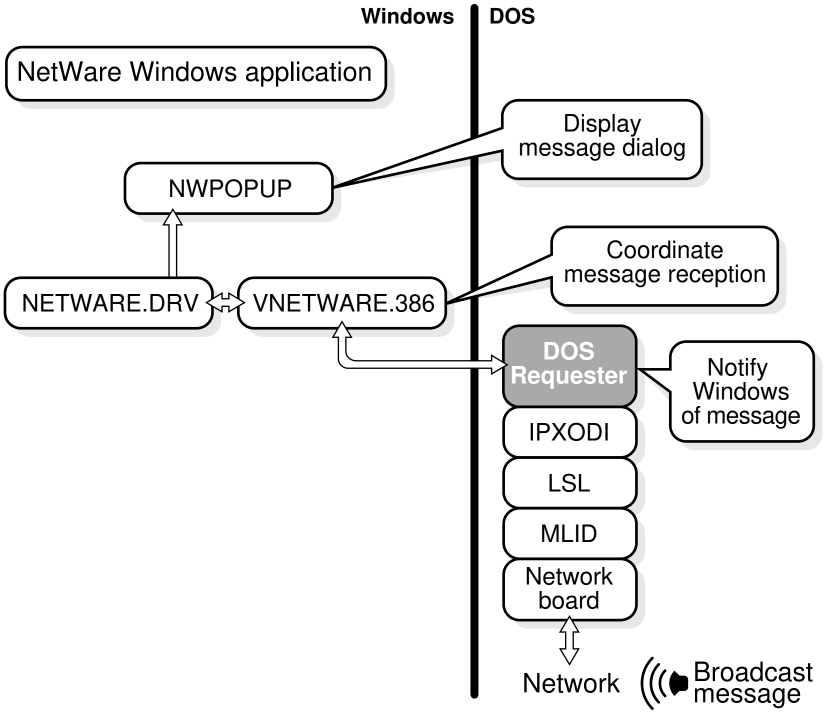


Receiving Broadcast Messages

Figure A-5 illustrates which NetW are components are used when your workstation receives a broadcast message. For example, if the secretar y sends a message informing you that paychecks have arrived, your workstation uses the components shown below .

VNETW ARE.386 works with NETW ARE.DRV and NWPOPUP .EXE to coordinate and display message dialogs.

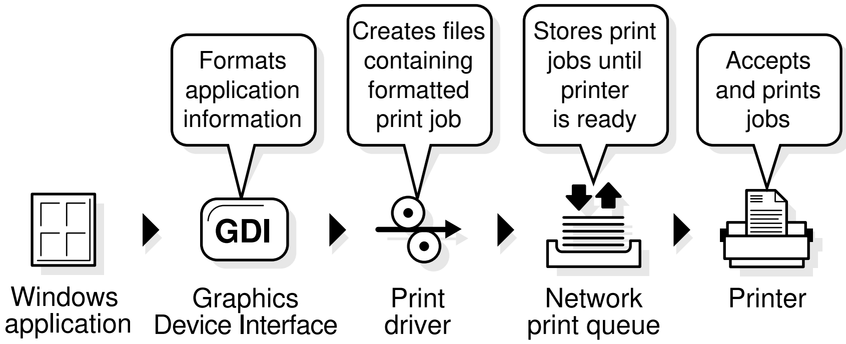
Figure A-5
Receiving a broadcast message



Printing to Network Queues

Figure A-6 illustrates W indows and NetW are components that are used when you print from a W indows application. For example, if you print a text file from the W indows Notebook application, your workstation uses the components shown below .

Figure A-6
Printing path


Formats application information
Creates files containing formatted print job
Stores print jobs until printer is ready
Accepts and prints jobs
GDI
Windows application
Graphics Device Interface
Print driver
Network print queue
Printer