

OS/2 Warp Server for e-business



Performance Tuning

OS/2 Warp Server for e-business



Performance Tuning

Note

Before using this information and the product it supports, read the general information under "Appendix B. Notices" on page 115.

First Edition

This edition applies to OS/2 Warp Server for e-business and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 1999. All rights reserved.**

US Government Users Restricted Rights – Use duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	v
Tables	vii
About This Book	ix
Who Should Use This Book	ix
Prerequisite and Related Information	ix
Chapter 1. LAN Server Design Concepts	1
What's New for Server Performance and Capacity Tuning?	1
LAN Server Design Overview	1
386 HPFS Design Overview.	5
Sideband	6
LAN Requester and NetBIOS over TCP/IP (TCPBEUI)	6
Chapter 2. Tuning LAN Server Performance Parameters	9
Capacity Tuning Considerations	10
Server IBMLAN.INI Capacity Parameters	10
Server PROTOCOL.INI Capacity Parameters	12
Additional Related Parameters	12
Performance Tuning Considerations	13
Tuning Assistant	13
Collecting Information about System Activity.	13
Server IBMLAN.INI Performance Parameters	14
Server Performance Considerations	15
Server PROTOCOL.INI Performance Parameters	16
Server CONFIG.SYS Performance Parameters.	17
OS/2 LAN Requester IBMLAN.INI Performance Parameters	18
OS/2 LAN Requester PROTOCOL.INI Performance Parameters	19
OS/2 LAN Requester CONFIG.SYS Performance Parameters	19
DOS LAN Services NETWORK.INI Performance Parameters	19
DOS LAN Services PROTOCOL.INI Performance Parameters	20
Matching Server and Requester Buffers (OS/2 LAN Requesters Only)	20
Adjusting Network Buffer Read-Ahead and Write-Behind Activity (OS/2 LAN Requesters Only).	20
General Hints and Tips	24
Memory Minimization Considerations	25
Server IBMLAN.INI Memory Parameters	25
Server PROTOCOL.INI Memory Parameters	26
Server CONFIG.SYS Memory Parameters	26
OS/2 LAN Requester IBMLAN.INI Memory Parameters.	26
OS/2 LAN Requester CONFIG.SYS Memory Parameters	27
Additional Related Parameters.	27
Chapter 3. IBMLAN.INI File Parameter Descriptions	29
Changing IBMLAN.INI Parameter Values	29
Activating Changed Parameter Values	29
Parameter Descriptions	30
Networks Section	30
Requester Section	32
Server Section - Server Function and Peer Function.	48
Server Section - 386 HPFS Option	65
Netlogon Section.	70

LSserver Section	72
Messenger Section	73
Alerter Section.	73
Netrun Section	74
Replicator Section	74
UPS Section	78
Remoteboot Section	81
DCDB Replicator Section.	82
Peer Section	84
Browser Section	85
Services Section	85
Chapter 4. NETWORK.INI File Parameter Descriptions for DOS LAN	
Services	89
Changing NETWORK.INI Parameter Values	89
Activating Changed Parameter Values	89
Parameter Descriptions	89
NETWORK.INI Messenger Parameters	92
NETWORK.INI Netpopup Parameter	92
NETWORK.INI Peer Parameters	92
Chapter 5. CONFIG.SYS File Parameter Descriptions	97
Changing CONFIG.SYS Parameter Values	97
Activating Changed Parameter Values	97
Memory Management	97
System Parameters	97
Tuning Parameters	98
Chapter 6. High Performance File Systems	101
HPFS Caching	101
Extended Attribute Support	103
386 High Performance File System Support	103
HPFS386.INI Parameter Descriptions	104
Cache386 Utility	107
Local Security Support.	109
Appendix A. File System Comparison Chart.	111
Appendix B. Notices	115
Trademarks.	116
Index	117

Figures

1.	System Architecture of the OS/2 LAN Server and OS/2 Requester	4
2.	System Architecture of the OS/2 LAN Server and DOS LAN Services . . .	5
3.	System Architecture of the OS/2 386 HPFS and OS/2 Requester	8
4.	System Architecture of the OS/2 386 HPFS and DOS LAN Services . . .	9
5.	Heuristic Relationships: OS/2 Requester Large-Record Buffering	21
6.	Heuristic Relationships: Server Large-Record Buffering	22
7.	Heuristic Relationships: Server Small-Record Buffering	23
8.	Heuristic Relationships: Requester Small-Record Buffering	24

Tables

1. IBMLAN.INI Parameters Affecting LAN Server Performance	14
2. CONFIG.SYS Parameters Affecting LAN Server Performance	17
3. Server Priority	60
4. Srvheuristics Digit Position 15 Values and Meanings	63
5. 386 HPFS Option Srvheuristics Digit Position 15 Values and Meanings . .	68
6. NETWORK.INI [network] Section Parameter Values	89
7. NETWORK.INI [messenger] Section Parameter Values	92
8. NETWORK.INI [netpopup] Section Parameter Values	92
9. NETWORK.INI [peer] Section Parameter Values	92
10. File System Comparison: JFS, 386 HPFS, HPFS, and FAT	112

About This Book

This book provides information concerning configuration parameters and performance-related topics for the LAN Server component of OS/2 Warp Server for e-business.

LAN Server runs on both single-byte character set (SBCS) and double-byte character set (DBCS) systems. SBCS is a graphic character set in which each character occupies 1 byte. DBCS is a graphic character set in which each character occupies 2 bytes. Languages (such as Japanese, Chinese, and Korean) that contain more symbols than can be represented by 256 code combinations require double-byte character sets. Entering, displaying, and printing DBCS characters require special hardware and support.

Who Should Use This Book

This book is intended as a reference for the *network administrator*, who is responsible for installing, configuring, tuning, managing, and controlling a network. The network administrator also defines resources to be shared and user access to the shared resources.

Additional users of this book include network users, IBM technical support personnel, marketing representatives, data processing professionals, and application developers.

You should also be familiar with the following:

- OS/2
- DOS
- An ASCII text editor

Prerequisite and Related Information

Quick Beginnings provides prerequisite information on the topics discussed in this book. It contains an introduction to OS/2 Warp Server for e-business, of which LAN Server is a component, and provides the steps and basic information needed to install OS/2 Warp Server.

Chapter 1. LAN Server Design Concepts

This chapter describes the major features of the file and print services of OS/2 Warp Server for e-business, called LAN Server hereafter, that affect performance. LAN Server is a component of OS/2 Warp Server for e-business. LAN Server consists primarily of server and requester services, which allow resources to be shared with other computers on a network. It has a 386 HPFS option that is available as an add-on for an additional licensing fee.

This chapter also describes the functions of the key performance parameters in the IBMLAN.INI and CONFIG.SYS files, which are the files that hold configuration information for LAN Server. Parameter descriptions and tuning guidelines are located in other chapters.

Refer to *Quick Beginnings* for a brief description of each installable function and the default components installed with each package.

What's New for Server Performance and Capacity Tuning?

- **Journalized File System (JFS)**

The Journalized File System (JFS) improves performance by allowing partitions up to 2TB and file sizes up to 2TB. For an explanation of JFS, see *Quick Beginnings*.

- **Logical Volume Manager (LVM)**

Logical Volume Manager (LVM) improves performance by allowing disk spanning, sticky drive letters, dynamic drive letters, and volume expansion with Journalized File System (JFS). For an explanation of LVM, see *Quick Beginnings*.

- **I2O Capability for LAN and for SCSI**

If your server has I2O adapters, CPU availability is increased because the instructions to support I2O have been moved to the adapter cards.

- **IBMLAN.INI Server Section**

Some IBMLAN.INI Server Section capacity parameters have been increased to improve performance.

- **maxconnections:** 16K
- **maxopens:** 64K
- **maxsearches:** 16K (386 HPFS 8K; DOS 6K)
- **maxshares:** 1500

The maximum number of ACLs for JFS resources is limited only by available disk space.

LAN Server Design Overview

LAN Server runs at application privilege level. It is well suited for multi-purpose servers running other applications, and may also serve as a desktop workstation. LAN Server uses OS/2 services to satisfy network file I/O requests, session setup, and resource sharing. It handles ring 3 requests, which include requests for files stored on a *file allocation table* (FAT) partition, print requests, and serial device requests. For better performance, recommended uses include:

- As a shared system with a local user
- As a shared system with primarily ring 3 applications

- As a dedicated print or serial device server
- As a multi-purpose server with plenty of memory to handle the anticipated amount of traffic without requiring a high-performance file server

Network file I/O requests and responses are sent by way of *server message blocks* (SMBs). The LAN Server processes SMBs using internal network buffers. The IBMLAN.INI file parameters that define the size and number of network buffers on the server are **sizreqbuf** and **numreqbuf**. The IBMLAN.INI file parameters that define the size and number of network buffers on the requester are **sizworkbuf** and **numworkbuf**. For system architecture, see Figure 1 on page 4

An SMB received from the network is copied into the adapter receive buffers by the network adapter. The NetBIOS device driver, using a global descriptor table (GDT) selector, copies the data from the adapter's receive buffers into an available server network buffer. The NetBIOS device driver can acknowledge the message or piggyback the acknowledgment on a subsequent network message. The PROTOCOL.INI file contains the configuration information for the NetBIOS device driver.

The SMB is passed through the redirector to the server. The *redirector* is a requester component that directs file system request traffic between the server, the file system, and the network. Three types of SMB protocols exist that can be used for transferring data between a requester and a server:

- *Core SMB protocol* header and data reside within one network buffer. This SMB protocol is used to transfer amounts of data that are less than or equal to the buffer size specified by the **sizworkbuf** parameter.
- *Read and Write (RAW) SMB protocol* header and data reside within the first network buffer. Subsequent data is transferred through big buffers without header information. This flow continues until all requested data has been received; no additional SMB requests are required. The RAW SMB protocol is used if the amount of data to transfer exceeds the network buffer size and if big buffers are available.
- *Multiplexed SMB protocol* header and data reside within the initial network buffer. Subsequent data is sent in network buffers without header information along with the SMB response protocol header. This flow continues until all requested data has been received; no additional SMB requests are required. Multiplexed SMB protocol is used if the amount of data to transfer exceeds the requester buffer size and if either RAW SMB protocol is not supported or big buffers are not available. The **numbigbuf** parameter in the IBMLAN.INI file on the server specifies the number of big buffers. Both RAW and Multiplexed protocols are used to transfer large amounts of data very quickly.

LAN Server is designed to optimize the movement of file I/O from the server to the requester. It supports the *Journalized File System* (JFS), the *file allocation table* (FAT) file system and the *High Performance File System* (HPFS). The JFS file system, the FAT file system, and the HPFS have a *cache* that is used to improve performance by keeping frequently used data in memory. A cache is a high-speed storage buffer that contains frequently accessed instructions and data. For more information about caches, see "Server CONFIG.SYS Performance Parameters" on page 17.

For comparative information about file systems, see "Appendix A. File System Comparison Chart" on page 111.

The design of LAN Server is optimized to handle these typical file accesses:

- *Random file access* is characterized by a request for a small amount of data that may reside anywhere in the file. In this case, the core SMB protocol is typically used by the requester. The data is cached in the file system to minimize disk seeks.
- *Sequential file access* is characterized by successive requests for data that is contiguous in the file. In this case, the core SMB protocol is used by the requester as long as the requested amount of data is less than or equal to the requester's network buffer size (**sizworkbuf**).
- *Large file transfer* is characterized by a request for an amount of data that is greater than the size of the **sizworkbuf**. In this case, the multiplexed or Read and Write (RAW) SMB protocol is used by the requester. In anticipation of the next request, the LAN Server performs read-ahead independently of the file system, using one of its big buffers if the RAW SMB protocol is used.

For the LAN Server, I/O requests to the file system are made using OS/2 APIs. Once an API call finishes, the server returns the SMB to the requester as an SMB response.

The following figures (see Figure 1 on page 4 and Figure 2 on page 5) show the components that may influence the performance of LAN Server and OS/2 requester, and the LAN Server and the DOS workstation. These figures also provide a reference for discussing individual performance-related elements.

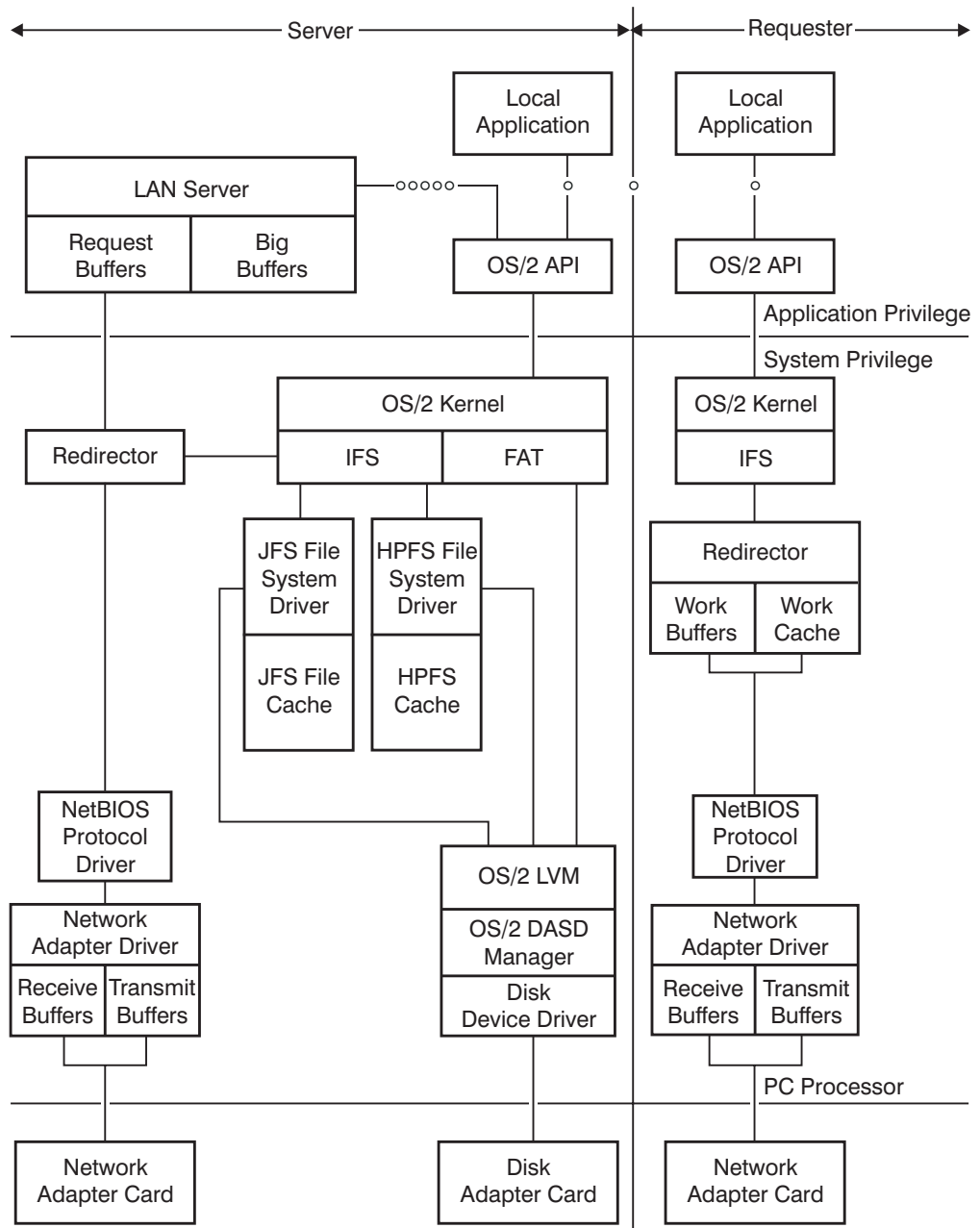


Figure 1. System Architecture of the OS/2 LAN Server and OS/2 Requester

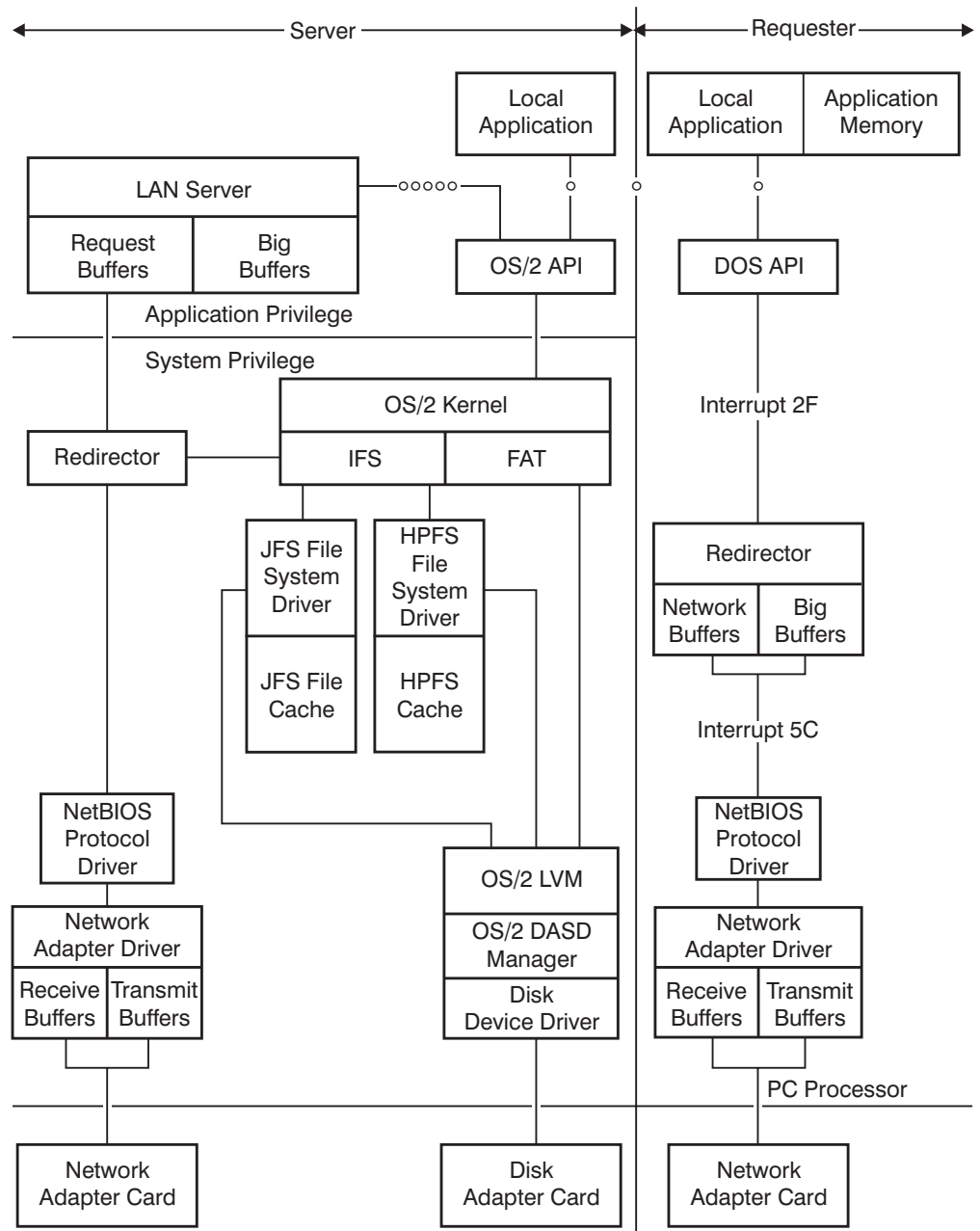


Figure 2. System Architecture of the OS/2 LAN Server and DOS LAN Services

386 HPFS Design Overview

386 HPFS option consists of an SMB processor tightly coupled to a file system (386 HPFS). These components are designed and optimized for 386-compatible and higher processors. The 386 HPFS option provides an OS/2 file system driver, running at system privilege, which requires minimal, if any, OS/2 services to satisfy network file I/O requests. Performance is enhanced by the use of Busmaster disk and network device drivers, which increase overlap of network and disk activity. The 386 HPFS option uses read-ahead and write-behind logic, which allows network file

I/O to occur at a rate approaching network bandwidth (when the data is fully cached) or disk transfer rates (when disk I/O is required).

386 HPFS provides access control information within each file. The access control information enables *Local Security*, provided with 386 HPFS, to function consistently with remote access control. Local Security is the LAN server component that allows a network administrator to secure the file system on the server workstation from local users. File I/O performance is enhanced because of a much larger cache capacity and more intelligent caching algorithms.

Server network buffers are passed to the server by the NetBIOS device driver. The 386 HPFS option is optimized for 386 HPFS I/O. It is enabled to support 486 and Pentium processors as well. Requests destined for non-HPFS resources, such as the FAT file system, character devices, and named pipes, are passed by the 386 HPFS option to the LAN Server. The requests are satisfied through OS/2 APIs.

The 386 HPFS option runs at the same privilege level as the OS/2 kernel (ring 0). The 386 HPFS option is able to accomplish this by servicing incoming requests for data or files in the HPFS partition without having to make any transitions between ring 0 and ring 3.

The following are recommended uses of the 386 HPFS option:

- As a dedicated high-performance file server
- As a remote IPL server

386 HPFS processes the three typical file accesses much like the LAN Server processes them. The 386 HPFS file I/O requests may be sent with any of the three SMB protocol types discussed previously for the LAN Server. For more information on the LAN Server, see "LAN Server Design Overview" on page 1. However, 386 HPFS uses scatter/gather logic to enhance performance if it is supported by network and disk device drivers.

Sideband

Sideband is a set of enhancements designed to improve the performance of small reads and writes to files on LAN server. These enhancements are contained in the DOS LAN Services and OS/2 LAN Requester components, in the 386 HPFS option components, and in the LAN Adapter and Protocol Support components.

If it is determined that an excessive number of frames are being lost on a single session, Sideband is disabled for that particular session and stays disabled until the session is ended. If you want to enable Sideband again, delete and restart the session by disconnecting and then reconnecting all connections to the server in question.

LAN Requester and NetBIOS over TCP/IP (TCPBEUI)

NetBIOS over TCP/IP (also known as TCPBEUI) provides a valuable functional enhancement to the LAN Server product by enabling a workstation to be geographically isolated from its domain and yet communicate with it transparently.

While the performance of NetBIOS over TCP/IP is good, the TCP/IP component must comply with architectural standards, which were defined to satisfy communication requirements over different physical networks at disparate speeds.

NetBEUI is optimized for high performance when used within a LAN segment, but does not allow for packet forwarding on routed networks. The NetBIOS interface does not rely on any specific underlying protocol to connect systems together, and as such, it is adaptable to protocols that allow packet forwarding, such as TCP/IP.

TCPBEUI is translation, not encapsulation. Once a NetBIOS connection has been established, TCPBEUI uses sockets-send commands over a TCP connection to send NetBIOS session data. TCPBEUI builds a 4-byte session header that precedes the actual user data. Therefore, a NetBIOS caching send of 128KB would have an overhead of only 4 bytes.

The Multi-Protocol Transport Services (MPTS) shipped with LAN Server provides the capability of configuring LAN Requester (or LAN Server) workstations with both NetBEUI and NetBIOS over TCP/IP on the same network interface card. This dual protocol stack configuration will allow local sessions to continue running with NetBEUI performance while also providing Wide Area Network connectivity with NetBIOS over TCP/IP.

The following figure (see Figure 3 on page 8) shows the components that may influence the performance of 386 HPFS and OS/2 requesters. This figure also provides a reference for discussing individual performance-related elements.

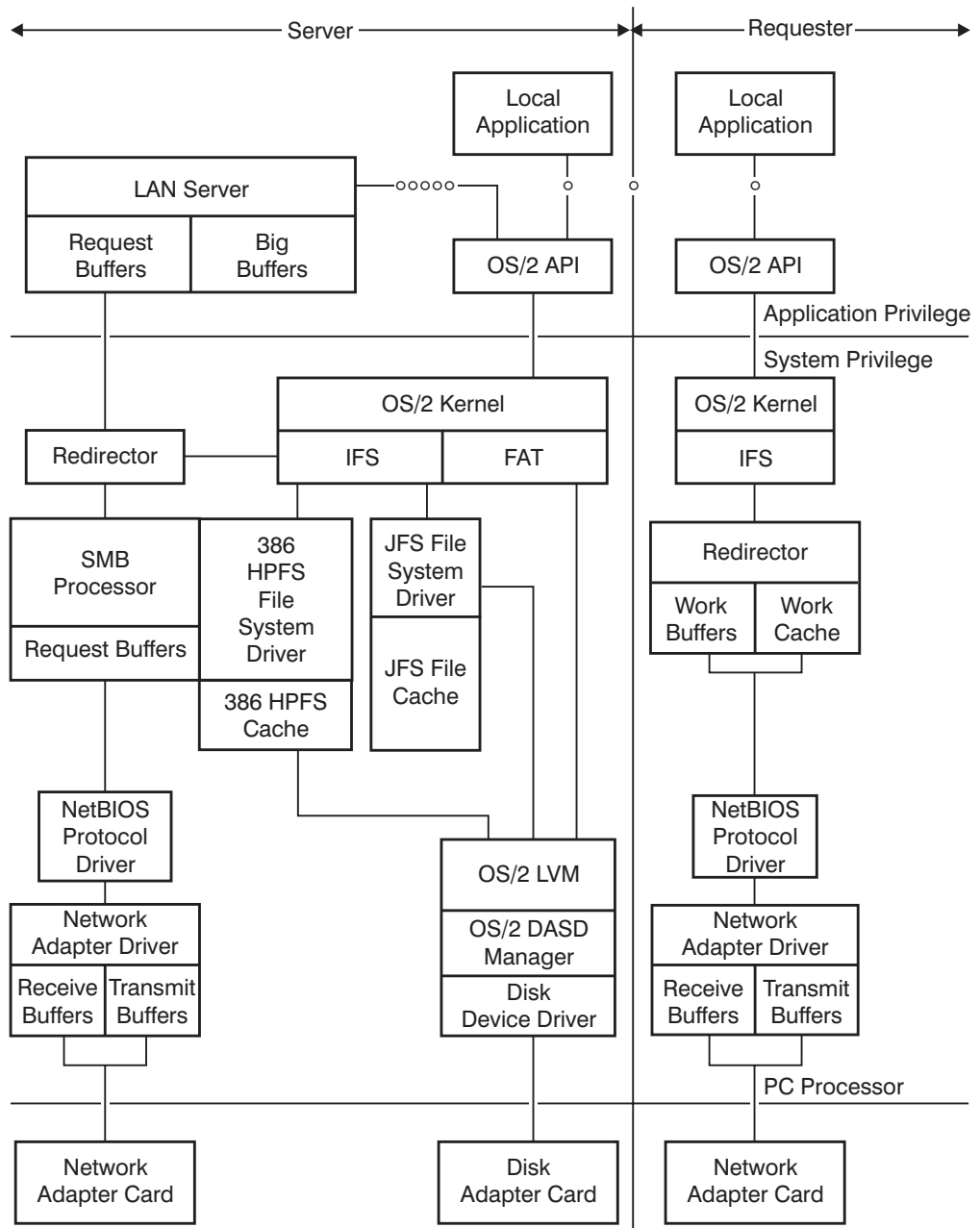


Figure 3. System Architecture of the OS/2 386 HPFS and OS/2 Requester

The following figure (see Figure 4 on page 9) shows the components that may influence the performance of the 386 HPFS and DOS requesters. This figure also provides a reference for discussing individual performance-related elements.

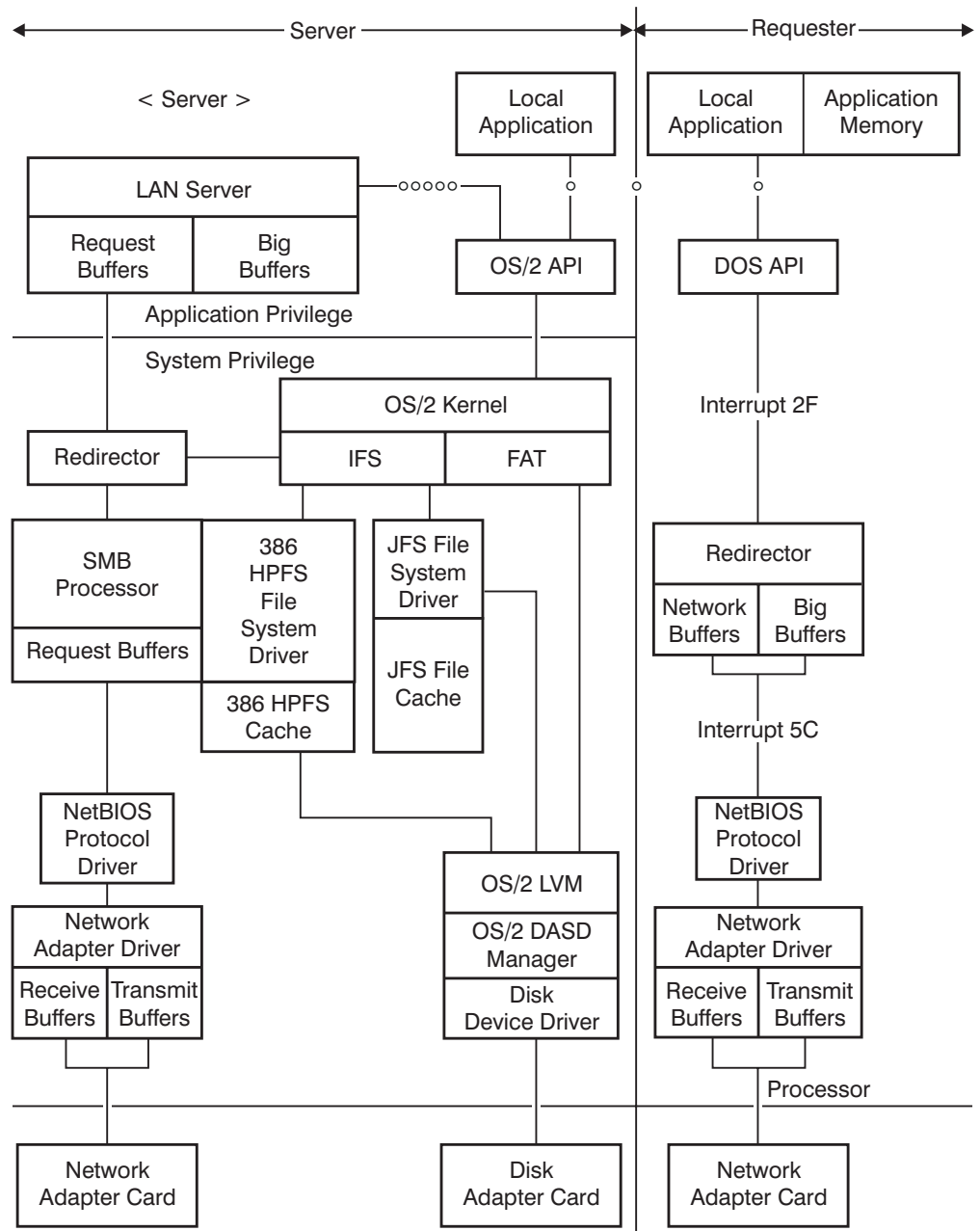


Figure 4. System Architecture of the OS/2 386 HPFS and DOS LAN Services

Chapter 2. Tuning LAN Server Performance Parameters

This chapter describes how to tune LAN Server for capacity, performance, and minimization of memory usage. The use of certain LAN Server functions to assist in the collection of information about system activity is discussed. The collected information is helpful when attempting to pinpoint the cause of a capacity or performance problem.

Note:

LAN Server provides a tool called Tuning Assistant which assists you in configuring LAN Server to your unique requirements. Its function is to configure for the users' capacity requirements and to allocate remaining memory for performance tuning.

All material in this chapter is applicable to both LAN Server and the 386 HPFS option (if it has been added) unless specifically stated otherwise. More detailed information about parameters listed in this chapter can be found in the following chapters.

Capacity tuning should usually be done prior to performance tuning.

Capacity Tuning Considerations

LAN Server defaults provide for a maximum of 32 NetBIOS sessions. If you want your server to connect to more than 32 workstations, you must increase the *x1* and *x2* variables on the corresponding **netx** statement in the IBMLAN.INI file. Corresponding changes must be made to parameters in the IBM OS/2 NetBIOS protocol section (NetBEUI_NIF) of the PROTOCOL.INI file.

Requirements for additional capacity parameter changes are determined by your particular environment.

Server IBMLAN.INI Capacity Parameters

The following IBMLAN.INI parameter list describes the parameters that most affect LAN Server capacity, offers recommendations for adjustment, and lists related parameters. The parameters are located in the Networks and Server sections of the IBMLAN.INI file. Some of these parameters are related to adding users to the network. Others are considered to be capacity parameters, but they may also affect performance because a lack of server resources forces a user to wait for a resource to become available. The default values for these parameters are set for a LAN with 200 users. For a detailed description of the IBMLAN.INI parameters, refer to "Chapter 3. IBMLAN.INI File Parameter Descriptions" on page 29.

If the server fails to start successfully after IBMLAN.INI parameter values are changed, record the error reported and use the NET ERROR command to determine the cause of the problem.

The following parameters affect LAN Server capacity:

Parameter	Description
<i>x1</i>	Number of NetBIOS sessions the server or requester allocates <ul style="list-style-type: none">• Range [Default]: 2 - 254 [93 per adapter] LAN Server [102 per adapter] 386 HPFS option• Recommendation: See <i>Quick Beginnings</i>.• Related parameters: Maximum sessions (NetBEUI_NIF)
<i>x2</i>	Number of simultaneous NetBIOS commands a server or requester can post <ul style="list-style-type: none">• Range [Default]: 16 - 255 [212 per adapter] LAN Server

	[225 per adapter] 386 HPFS option
	<ul style="list-style-type: none"> • Recommendation: Set to 2 x x1. • Related parameters: Maximum commands (NetBEUI_NIF)
x3	<p>Number of NetBIOS names the server or requester allocates</p> <ul style="list-style-type: none"> • Range [Default]: 5 - 254 [14 per adapter] • Recommendation: No recommendation • Related parameters: Maximum names (NetBEUI_NIF)
maxlocks	<p>Number of records locked by server users</p> <ul style="list-style-type: none"> • Range [Default]: 1 - 8000 [64] LAN Server or 386 HPFS option • Recommendation: 1 x maxopens • Related parameters: maxopens
maxopens	<p>Number of files, pipes, and devices simultaneously open on the server</p> <ul style="list-style-type: none"> • Range [Default]: 1 - 65535 [128] Peer [256] LAN Server [256] 386 HPFS option • Recommendation: 3 x maxusers • Related parameters: maxlocks maxsessopens
maxsearches	<p>Number of simultaneous directory searches on the server</p> <ul style="list-style-type: none"> • Range [Default]: 1 - 16384 [50] Peer [150] LAN Server [350] 386 HPFS option • Recommendation: Increase for heavily used files on the server (DLS requesters only; LAN Server only). Increase for print servers; if parameter is not sufficient, jobs disappear in a print queue without printing. • Related parameters: maxlocks maxusers srvheuristics (position 7)
maxsessopens	<p>Number of files, pipes, and devices simultaneously open for a requester.</p> <ul style="list-style-type: none"> • Range [Default]: 1 - 32768 [80] LAN Server or Peer [256] 386 HPFS option • Recommendation: Increase for many server resources simultaneously used from one requester (LAN Server server only). • Related parameters: maxopens
maxsessreqs	<p>Number of resource requests for a requester</p> <ul style="list-style-type: none"> • Range [Default]: 1 - 65535 [50] • Recommendation: Increase if users need multiple simultaneous resources.

- Related parameters: None
- maxshares** Number of server resources to share
- Range [Default]: 2 - 1500
 - [16] Peer
 - [192] LAN Server
 - [192] 386 HPFS option
 - Recommendation: Increase if many server resources are shared.
 - Related parameters: None
- maxusers** Number of simultaneous users (NetBIOS sessions) on the server
- Range [Default]: 1 - 1000
 - [5] Peer
 - [100] LAN Server
 - [101] 386 HPFS option
 - Recommendation: Set less than **maxconnections**.
 - Related parameters: **maxconnections**
- srvpipes** Number of pipes used by the server
- Range [Default]: 1 - 20 [3]
 - Recommendation: Increase if many users log on simultaneously;
maxusers / 40
 - Related parameters: **maxusers**

Server PROTOCOL.INI Capacity Parameters

The **Maximum sessions (sessions)**, **Maximum commands (ncbs)**, and **Maximum names (names)** parameter values in the IBM OS/2 NetBIOS (NETBEUI_NIF) section of the PROTOCOL.INI file must be set to a value that is greater than or equal to the corresponding *x1*, *x2*, and *x3* values on the associated **netx** statement in the IBMLAN.INI file. The values for these parameters must be set in this way for TCPBEUI_NIF also. For information on the parameters of the PROTOCOL.INI file, see *Quick Beginnings*.

Additional Related Parameters

Several files contain related parameters that do not significantly affect capacity and do not usually need to be adjusted. The files that contain related parameters are as follows:

- Server CONFIG.SYS
- OS/2 LAN Requester IBMLAN.INI
- OS/2 LAN Requester PROTOCOL.INI
- OS/2 LAN Requester CONFIG.SYS
- DOS LAN Services NETWORK.INI
- DOS LAN Services PROTOCOL.INI
- DOS LAN Services CONFIG.SYS

Performance Tuning Considerations

Performance can be viewed as either faster response time to individual workstation requests or more data throughput from the server to the network as a whole. During installation, OS/2 Warp Server for e-business Tuning Assistant, hereafter referred to as Tuning Assistant, runs automatically to configure LAN Server to your system requirements. When you make changes to your system, you can run it again. You can also collect information about system activity to analyze system performance.

Tuning Assistant

Tuning Assistant is a utility which can be used to insure:

- Certain configuration parameters are within valid boundaries for the system being tuned.
- Related parameters maintain a valid relationship.
- Configuration limits are maintained so that NetBEUI and TCPBEUI protocol drivers will not fail to load because of incorrect parameter settings.

It provides you with the following information:

- Recommended changes to certain parameters to improve performance.
- Identification of possible system needs such as additional memory or LAN adapters
- Updated configuration files (CONFIG.SYS, IBMLAN.INI, PROTOCOL.INI, HPFS386.INI OR WCLLOCAL.INI) based on the system's identified hardware or user input.

Tuning Assistant runs automatically at installation. Tuning Assistant should also be run when you do any of the following actions:

- Add or remove memory, disk space, or LAN adapters.
- Makes changes to disk partitions or volumes.
- Reconfigure the NetBIOS protocol for either NetBEUI or TCPBEUI.
- Edit parameters related to any of the above in CONFIG.SYS, IBMLAN.INI, PROTOCOL.INI, HPFS386.INI OR WCLLOCAL.INI.
- Want to experiment with the inputs to parameters in CONFIG.SYS, IBMLAN.INI, PROTOCOL.INI, HPFS386.INI OR WCLLOCAL.INI. to see the related impacts.
- Recognize that the NETBEUI OR TCPBEUI protocol drivers will not load.

Tuning Assistant may be started from the command line. Change to the \IBMLAN directory and type `nstune`. You have the option of accepting or not accepting recommended changes.

Collecting Information about System Activity

You can gather information about system activity to help identify the possible causes of a performance problem:

- Examine the server logs and statistics reports. Refer also to *Network Administrator Tasks* for more information about the logs and statistics reports. The statistics and error information can also be accessed with the NET STATISTICS or NET ERROR command from the command line on the server workstation. The reports include important data, such as the number of times big buffers, request buffers, or other resources were exhausted. After running a

typical workload on the server, examine the data to determine if the server is frequently running out of big buffers, request buffers, or other resources. If so, the number of big buffers, request buffers, or other resources should be increased unless there are memory constraints in the server workstation.

- For the 386 HPFS option, the CACHE386 utility can help you analyze server performance. To display statistics about cache usage, type CACHE386 /STATS at an OS/2 command prompt on the server. To achieve maximum benefit, use the cache read-hit ratio to maximize the hit ratio with a minimum-sized cache.

After you have collected information about the system activity, you can start to pinpoint the possible causes of a performance problem. Required actions can include:

- Minor tuning of a few parameters
- Redistributing available memory in the workstation
- Adding memory to the workstation
- Redistributing the work load among servers
- Upgrading the servers with more powerful hardware
- Adding additional servers

Server IBMLAN.INI Performance Parameters

Table 1 describes the IBMLAN.INI file parameters that affect server workstation performance, offers recommendations for adjustment, and lists related parameters. The parameters are located in the Server section of the IBMLAN.INI file. In the following table, the parameters listed first generally have the greatest effect on performance.

The **srvheuristics** parameter can have a significant effect on performance. However, in most cases changing the default settings results in minimal performance fluctuation. For more information concerning the effects of the **srvheuristics** parameter on read-ahead and write-behind activity, refer to “Adjusting Network Buffer Read-Ahead and Write-Behind Activity (OS/2 LAN Requesters Only)” on page 20 .

Table 1. IBMLAN.INI Parameters Affecting LAN Server Performance

Parameter	Range [Default]	Recommendation	Related Parameters
numbigbuf	0 - 80 [4] Peer; [6] LAN Server; [12] 386 HPFS	LAN Server: 2 for each simultaneous heavy-load user; 12 if 386 HPFS installed.	srvheuristics (positions 13, 17, and 18)
numreqbuf	5 - 2000 [10] Peer; [48] LAN Server; [250] 386 HPFS	2 for each requester actively sending requests.	sizreqbuf, maxruns
sizreqbuf	1024 - 32768 [4096]	Set equal on all servers and equal to sizworkbuf on requesters.	numreqbuf, sizworkbuf, maxruns
srvheuristics		See complete description in “Chapter 3. IBMLAN.INI File Parameter Descriptions” on page 29.	wrkheuristics, numbigbuf

Two key server performance parameters requiring additional discussion are **numbigbuf** and **numreqbuf**.

- **numbigbuf**

The **numbigbuf** parameter may need to be adjusted differently, depending on whether the 386 HPFS option is installed.

For best results with LAN Server, the server should have enough big buffers available to handle a peak request workload for large sequential-file-access operations. This includes program loading as well as file copy operations. Allocate two big buffers for each requester that might concurrently send this type of request to the server. Because each big buffer is 64KB in size, specify the minimum number required to handle the typical environment.

Checking network statistics and error logs can help you determine an optimum number of big buffers for a particular environment. To determine if the server has enough big buffers to handle the incoming requests during peak server workload periods, check network statistics and the error logs. If necessary, increase the **numbigbuf** parameter value, and repeat this process until the server has enough big buffers to handle the desired workload.

Digit positions 17 and 18 of the **srvheuristics** parameter enable you to modify the dynamic use of big buffers. These digit positions enable the server to request additional memory from the base operating system when all the big buffers are in use and more big buffers are needed. Digit position 17 of the **srvheuristics** parameter specifies the length of time that the server maintains unused dynamic memory before freeing it. Digit position 18 determines the interval of time the server waits after failing to allocate a big buffer before trying again. If the server has sufficient memory, set the **numbigbuf** parameter value to ensure optimum performance.

Big buffers set with the **numbigbuf** parameter are not used in the 386 HPFS option unless files are accessed from a FAT or JFS file system drive. Because big buffers are used for printing and for accessing FAT and JFS drives, leave the **numbigbuf** parameter value at the default value.

The 386 HPFS option may allocate up to sixty-four 64KB buffers as needed to handle network I/O. In cases where large file transfers are done and the user has exclusive access to the file, you can use big buffers for data transfer instead of using the cache. This will prevent the cache from getting flushed for one-time accesses to files.

- **numreqbuf**

For best results, the server should have enough request buffers available to handle a peak request workload. Allocate a minimum of two request buffers for each requester that has a session established with the server.

Checking network statistics and error logs can help you determine an optimum number of request buffers for a particular environment. To determine if the server has enough request buffers to handle incoming requests during peak server workload periods, check network statistics and the error logs. If more request buffers are needed, increase the **numreqbuf** parameter value, and repeat this process until the server has enough request buffers to handle the desired workload. For more information, see the **numreqbuf** parameter in “Tuning Parameters” on page 56.

Server Performance Considerations

Because the needs of a particular workstation can change in a variety of ways, the resources required to support various application programs may vary. The following situations may require adjustment of the performance-related parameters:

- Heavy random I/O environments need a large disk cache on the server.

- A large disk cache can improve performance in environments that use shared applications.
- Heavy file-transfer environments need large numbers of 64KB buffers (**numbigbuf** parameter) on the server. However, if servers with the 386 HPFS option access only 386 HPFS hard disks, the **numbigbuf** parameter has no effect; in this case, leave the **numbigbuf** parameter set at the default value.

Some additional points to consider include:

- The **srvheuristics** and **wrkheuristics** parameters can be changed to improve LAN Server performance in some special instances. However, unless there is good reason to suspect a performance problem in your environment, the **srvheuristics** and **wrkheuristics** parameter values should not be changed. The benefit is generally small compared to other changes previously recommended. For more information concerning the effects of the **srvheuristics** and **wrkheuristics** parameters on read-ahead and write-behind activity, refer to "Adjusting Network Buffer Read-Ahead and Write-Behind Activity (OS/2 LAN Requesters Only)" on page 20. The **srvheuristics** and **wrkheuristics** parameters are also described in "Chapter 3. IBMLAN.INI File Parameter Descriptions" on page 29.

Server PROTOCOL.INI Performance Parameters

The **Maximum sessions (sessions)**, **Maximum commands (ncbs)**, and **Maximum names (names)** parameter values in the IBM OS/2 NetBIOS (NETBEUI_NIF) section of the PROTOCOL.INI file must be set to a value that is greater than or equal to the *x1*, *x2*, and *x3* values on the associated **netx** statement in the IBMLAN.INI file.

If you are using the IBM 16/4 Token-Ring Adapter, the **Transmit buffer size (xmitbufsize)** parameter value should be set to 128 + **sizworkbuf** to improve data transmission efficiency. The **Transmit buffer size (xmitbufsize)** parameter is added to the IBM Token-Ring Network Adapter (IBMTOK_NIF) section of the PROTOCOL.INI file. The default number of network control blocks (NCBs) is 225. Use the NET ERROR command to determine if this number is sufficient for the server workload.

If you are using the IBM 16/4 Token-Ring Adapter, you can set the **Number of adapter transmit buffers (xmitbufs)** parameter to 2 to allow overlapped buffering of transmitted data. The **Number of adapter transmit buffers (xmitbufs)** parameter is in the IBM Token-Ring Network Adapter (IBMTOK_NIF) section of the PROTOCOL.INI file.

Frames are sent to the protocol drivers in the reverse order from which the protocol driver DEVICE statements are listed in the CONFIG.SYS file. For information on the PROTOCOL.INI file, see *Quick Beginnings*.

Performance Tuning TCP/IP MTU size and XMITBUFSIZE Parameters

If an application performs many large data transfers, you may want to increase the Maximum Transmissible Unit (MTU) size to improve performance. File transfers of 2KB or more benefit from increasing the MTU size. If most of your file transfers are less than 2KB, then the default of 1500 is recommended.

The MTU size can be changed by using the IFCONFIG command in the OS/2 TCP/IP SETUP.CMD file. Set the MTU size to a number equal to the appropriate packet size plus 40 bytes (the maximum TCP/IP header size). The packet size must be a multiple of 2048. For example:

Packet size	Header size	MTU size
2048 (2KB)	40	2088
4096 (4KB)	40	4136
8192 (8KB)	40	8232

A related parameter **Transmit buffer size (xmitbufsize)** in the Token-Ring network adapter section of the PROTOCOL.INI file must also be configured to support the new MTU size. **xmitbufsize** must be configured to support transmission of buffers that are at least the size of the MTU you have specified. By default, **xmitbufsize** is not specified in PROTOCOL.INI and its value defaults to the largest size that your adapter card and ring speed allows. If this parameter is added, it must be set equal to, or greater than, the MTU size.

To prevent data loss, the MTU size must be less than, or equal to, the maximum **xmitbufsize** value allowed for your adapter card minus 40 bytes for the header.

Refer to the .NIF file (for example, IBMTOK.NIF) for your Token-Ring adapter card in PROTOCOL.INI to determine the valid values for the MTU and **xmitbufsize** parameters.

Server CONFIG.SYS Performance Parameters

The following table describes the CONFIG.SYS file parameters that affect server workstation performance, offers recommendations for adjustment, and lists related parameters.

Table 2. CONFIG.SYS Parameters Affecting LAN Server Performance

Parameter	Description	Range [Default]	Recommendation
/C	Memory used to cache HPFS I/O for LAN Server	64KB - 2MB [64KB]	2MB. See the information following the table.
diskcache	Memory used to cache FAT file system I/O	64KB - 8MB [64KB]	64KB if no FAT file system directories and files are being shared—otherwise, as much as possible. See the information following the table.

The **/C** (HPFS) and **diskcache** (FAT file system) parameters use most of the available memory in the server after other memory requirements are satisfied. It is recommended that you leave between 0.5MB and 1MB of free memory in the server for efficient handling of peak workloads. SPM/2 2.0 can be used to determine the server memory requirements. Refer to *Quick Beginnings* for information to help you determine the minimum memory requirements.

LAN Server Guidelines

Set the **diskcache** parameter to `xxxx,128`, where `xxxx` is the amount of memory available for caching and 128 is the threshold (in sectors). Enabling the **diskcache** lazy write option will improve performance in many cases. See 99 for more

information. The HPFS cache has a maximum value of 2MB and a default threshold of 2KB. Unless your system is memory constrained, the maximum value of 2MB will provide the best performance. See “Chapter 6. High Performance File Systems” on page 101 for more information.

386 HPFS Option Guidelines

If the **Workstation determines size** option was chosen for cache size at installation, the 386 HPFS option will choose a value based on the amount of the system memory. At the OS/2 command line, type: `cache386/0` to see the `cachesize` value. If used, the Tuning Assistant will calculate a value for `cachesize` and insert the statement into the HPFS386.INI file. The 386 HPFS option cache has no threshold or capacity limitations except the amount of memory. After determining the memory requirements on the server, you can change the cache size by changing value for the **cachesize** parameter in the HPFS386.INI file. The **cachesize** parameter must be at least 256KB in size.

Heap Considerations

The *heap* contains various non-I/O file-related information. The **maxheap** parameter in the filesystem section of the HPFS386.INI in the 386 HPFS option specifies the size of the heap. It is recommended that you let the heap size remain at the default value. As the system runs, the heap grows as needed to a maximum size limited only by available physical memory in the workstation. On a very busy workstation, it is possible to run out of heap space. If this occurs, there may be a noticeable degradation in performance, and service requests may fail. Errors returned to the requester that specify the server is out of a resource, especially on DOS searches, may indicate that the server has run out of heap space. For more information about the **maxheap** parameter, refer to “Chapter 6. High Performance File Systems” on page 101.

In environments where physical memory is limited and more heap is required, it is important to know that the 386 HPFS option preallocates 64KB for the heap initially, if **maxheap** is not set. Otherwise, it allocates 1/4 of the size specified up to 512 KB. For example, if no heap size is specified, and the heap needs to grow past the initial 64KB and cannot because other applications are using the physical memory, specifying a heap size would help. If a heap limit of 2MB is then specified, the 386 HPFS option would allocate 25% of this, or 512KB, which may be enough to satisfy the heap requirements. The downside is that if you set the heap size larger than you need it, the 386 HPFS option can pre-allocate more than is needed, taking excessive physical memory.

OS/2 LAN Requester IBMLAN.INI Performance Parameters

Changes to the requester parameters are not recommended unless you suspect a performance problem on the requester.

The **wrkheuristics** parameter can have a significant effect on performance. However, in most cases minimal or negative performance changes result from changing the default settings. Refer to “Adjusting Network Buffer Read-Ahead and Write-Behind Activity (OS/2 LAN Requesters Only)” on page 20 for more information concerning the effects of the **wrkheuristics** parameter on read-ahead and write-behind activity.

The following IBMLAN.INI parameters affect OS/2 requester performance:

Parameter	Description
maxwrkcache	Size of the work cache (KB) <ul style="list-style-type: none"> • Range [Default]: 0 - 640 [64] • Recommendation: Multiple of 64; increase if requesters run multiple file-intensive applications from the server. • Related parameters:
sizworkbuf	Size of each work buffer (bytes) <ul style="list-style-type: none"> • Range [Default]: 1024 - 16384 [4096] • Recommendation: Multiple of 512; set equal on all requesters and equal to sizreqbuf on servers. • Related parameters: sizreqbuf
numworkbuf	Number of work buffers <ul style="list-style-type: none"> • Range [Default]: 3 - 50 [15] • Recommendation: Increase if multiple applications running on the requester access server resources. • Related parameters: maxthreads maxcmds wrknets
wrkheuristics	Requester fine-tuning options <ul style="list-style-type: none"> • Range [Default]: • Recommendation: See complete description in “Chapter 3. IBMLAN.INI File Parameter Descriptions” on page 29. • Related parameters: srvheuristics sizworkbuf numbigbuf printbuftime

OS/2 LAN Requester PROTOCOL.INI Performance Parameters

If you are using the IBM 16/4 Token-Ring Adapter, and you want to improve data transmission efficiency, set the **Transmit buffer size (xmitbufsize)** parameter value to **sizworkbuf + 2**, which at default value equals 128. The **Transmit buffer size (xmitbufsize)** parameter is in the IBM Token-Ring Network Adapter (IBMTOK_NIF) section of the PROTOCOL.INI file.

If you are using the IBM 16/4 Token-Ring Adapter, you can improve data transmission efficiency by setting the **Number of adapter transmit buffers (xmitbufs)** parameter to 2 to allow overlapped buffering of transmitted data. The parameter is in the IBM Token-Ring Network Adapter (IBMTOK_NIF) section of the PROTOCOL.INI file. For information on the PROTOCOL.INI file, see *Quick Beginnings*.

OS/2 LAN Requester CONFIG.SYS Performance Parameters

The parameters in the CONFIG.SYS file usually do not require performance tuning for OS/2 LAN Requesters.

DOS LAN Services NETWORK.INI Performance Parameters

Table 6 on page 89 describes the NETWORK.INI file parameters that affect DLS requester performance, offers recommendations for adjustment, and lists related parameters. The parameters listed first generally have the greatest effect on performance. The default values have been chosen to provide good performance for most users with minimal memory requirements. Change the default values only

if there is a performance problem. When one parameter value is changed, related parameter values should also be adjusted. Although some of the parameters listed do not directly affect performance, they do control the amount of memory available.

DOS LAN Services PROTOCOL.INI Performance Parameters

If you are using the IBM 16/4 Token-Ring Adapter, you can improve data transmission efficiency by setting the **adapter transmit buffer** size to 4224 and by setting the number of **transmit buffers** to 2. Using an ASCII editor, add these two parameters to the IBM Token-Ring Network Adapter (IBM\$IBMTRA.NIF) section of the PROTOCOL.INI file:

```
XMITBUFSIZE=4224  
XMITBUFS=2
```

Matching Server and Requester Buffers (OS/2 LAN Requesters Only)

Previous sections have discussed matching buffer sizes in the network server and workstation. When the buffers are not matched, the request-processing overhead in the server increases, causing a degradation of performance in some cases. In addition, the memory in the larger of the mismatched buffers is not available for use and is wasted.

There are situations in which the 4KB default value for the server and requester buffers is not optimum; for example, if the network administrator monitors the traffic on the ring and determines that most frames are less than the transmit buffer size on the workstations. In this case, buffers could be reduced, providing a reduction in memory or an increase in the number of buffers. Performance may improve because it is desirable to have two request buffers for each active workstation.

Adjusting Network Buffer Read-Ahead and Write-Behind Activity (OS/2 LAN Requesters Only)

LAN server incorporates several design features that attempt to improve LAN performance based on expected behavior of software applications used on the network. However, the behavior of all possible applications cannot be predicted and handled in an optimum manner. The **srvheuristics** and **wrkheuristics** parameters enable you to deal with unusual behavior of applications used on the network, even though the application may not be aware of the network. The following information describes some of the digit positions of the **srvheuristics** and **wrkheuristics** parameters that may help you deal with specific application activity. In addition, read "General Hints and Tips" on page 24.

LAN Server adds the ability to disable the RAW (read and write) protocol at the server. This can be useful if you have a remote server and need to avoid using RAW protocol to prevent session timeouts.

To identify some of the relationships that exist in the **srvheuristics** and **wrkheuristics** parameters in the IBMLAN.INI file, refer to the figures about heuristic relationships, "OS/2 Requester Large-Record Buffering," "LAN Server Large-Record Buffering," "LAN Server Small-Record Buffering," and "Requester Small-Record Buffering." The **srvheuristics** parameter sets some fine-tuning options on the Server workstation. The **wrkheuristics** parameter sets some fine-tuning options on an OS/2 workstation. Except where noted, the figures show all possible combinations of the parameter values, as well as the relationships they create. The figures also may help you determine which buffering scheme is active.

The figures outline the effect of several fine-tuning options set by the **srvheuristics** and **wrkheuristics** parameters. Possible values of specific digit positions are shown at the bottom or side of each box. Each possible value defines a relationship that determines which buffering scheme is active. To use any of the heuristic relationship figures:

- At the bottom of the figure, locate the buffering scheme that you want to activate. Follow the path to the top of the figure and record the parameter values required to establish the chosen buffering scheme.
- To determine which buffering scheme is active, start at the top of the figure. Follow the downward path determined by the parameter values that are set in the existing IBMLAN.INI file. The path ends at the active buffering scheme.

The following figure (see Figure 5) outlines how the **wrkheuristics** parameter value affects the requester buffering scheme for large-record reads and writes. Large records are defined as those records larger than the **sizworkbuf** parameter value.

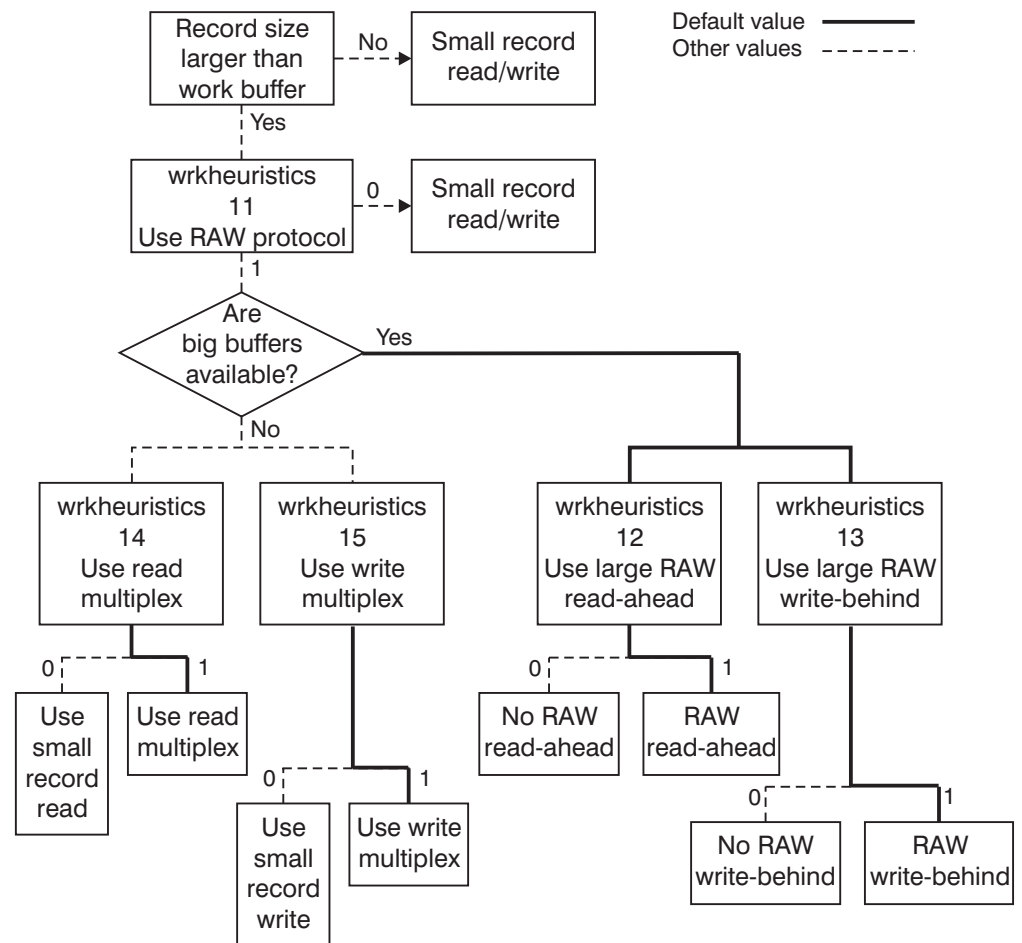


Figure 5. Heuristic Relationships: OS/2 Requester Large-Record Buffering

The following figure (Figure 6 on page 22) outlines how the **srvheuristics** parameter value affects the server buffering scheme for large-record reads and writes. Large records are defined as those records larger than the **sizreqbuf** parameter value.

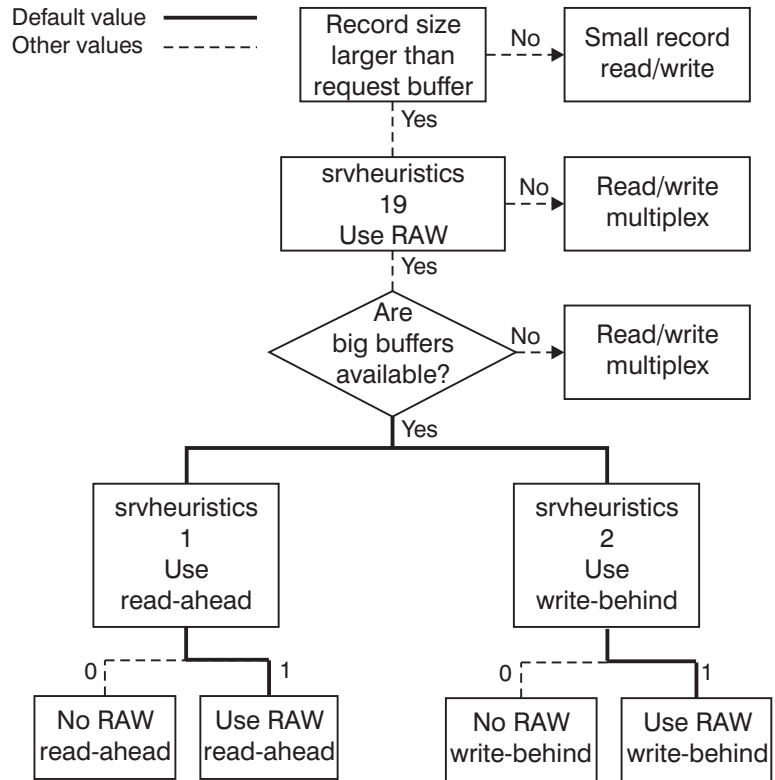


Figure 6. Heuristic Relationships: Server Large-Record Buffering

Possible values for digit positions 1 and 2 of the **srvheuristics** parameter are 0, 1, and 2. Values 0 and 1 are the only ones dealt with in the previous figure. Values 1 and 2 are functionally the same with regard to this figure, but they may have slightly different performance characteristics.

The following figure (Figure 7 on page 23) outlines how the **srvheuristics** parameter value affects the buffering scheme for small-record reads and writes. Small records are defined as those records smaller than or equal to the **sizeqbuf** parameter value.

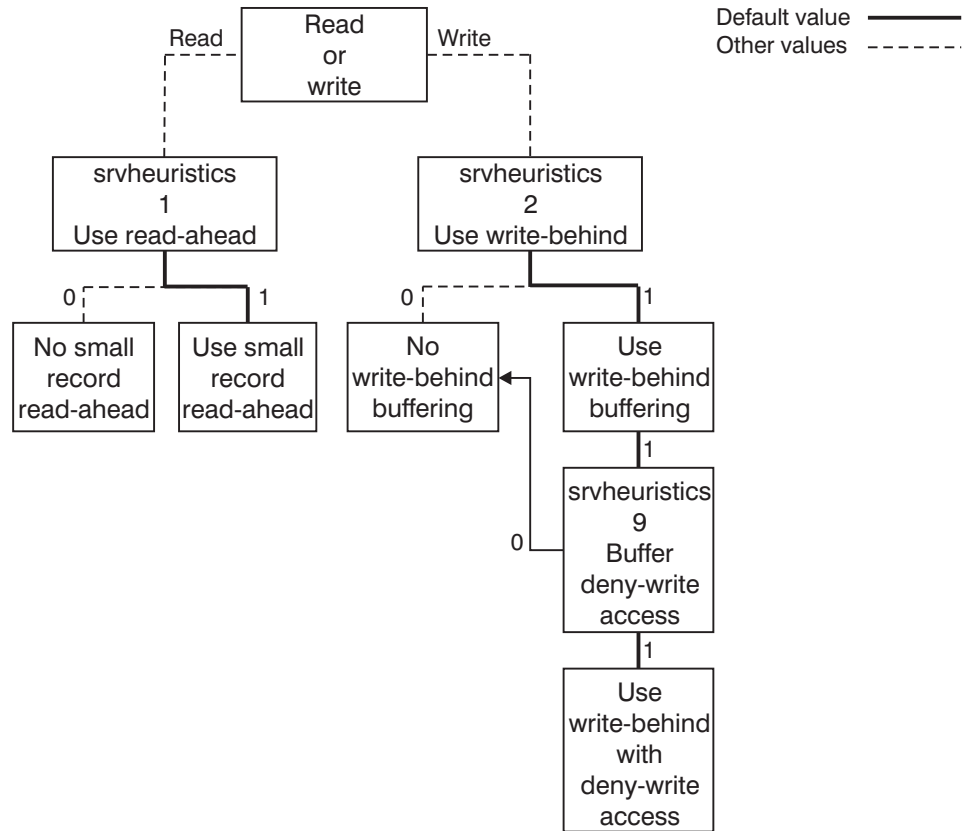


Figure 7. Heuristic Relationships: Server Small-Record Buffering

Possible values for digit positions 1 and 2 of the **srvheuristics** parameter are 0, 1, and 2. Values 0 and 1 are the only ones dealt with in Figure 7. Values 1 and 2 are functionally the same with regard to this figure, but they may have slightly different performance characteristics.

The following figure (Figure 8 on page 24) outlines how the **wrkheuristics** parameter value affects the buffering scheme for small-record reads and writes. Small records are defined as those records smaller than or equal to **sizeworkbuf**.

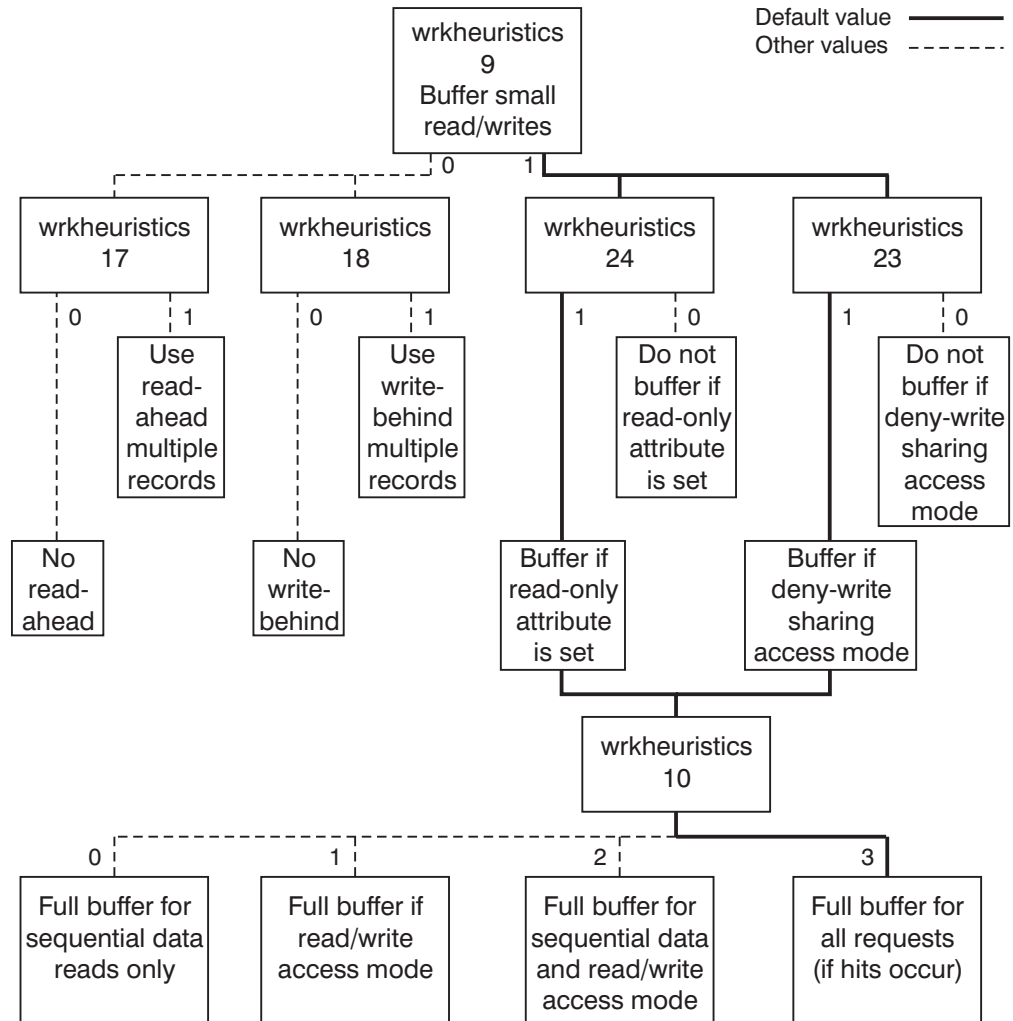


Figure 8. Heuristic Relationships: Requester Small-Record Buffering

General Hints and Tips

This section provides certain tuning recommendations that may improve performance for each specific environment. Review these recommendations and follow those that apply to your specific networks.

- To improve performance of OS/2 .CMD and DOS .BAT files, the server opens the files with *opportunistic locks* and does not close them immediately. Opportunistic locking, or Oplock, is the buffering of network files on a local workstation. However, this causes the files to remain open longer than expected, possibly resulting in file I/O errors. In addition, when files are copied from the DOS command prompt, the times and dates may change. To disable batch-file performance optimization for OS/2 workstations, set digit position 1 of the **wrkheuristics** parameter to 0.
- If keyboard response time is a concern on a heavily used server workstation (LAN Server package only), the priority level on the server can be changed. Set digit position 6 of the **srvheuristics** parameter to 5, which allows other applications to access the processor more often. However, network response time increases as the server keyboard response time decreases.

- If the workstation is a dedicated Remote IPL server (LAN Server package only), leave digit position 6 of the **srvheuristics** parameter at the default value of 4, which sets the server priority above other applications.
- If your network includes remote bridges, you may experience session disconnects of the remote workstation. Refer to the description of digit position 15 of the **srvheuristics** parameter in “Chapter 3. IBMLAN.INI File Parameter Descriptions” on page 29 for information about increasing the NetBIOS timeout.
- If your network does not include multiple bridges, remote bridges, telecommunication lines, or other demanding devices, and your network exhibits infrequent retransmissions, you may obtain an improvement in logon performance with the following actions:
 - Reduce the **netbiosretries** parameter in the workstation’s PROTOCOL.INI file from its default to a value of 2. If this value is too low, you will experience a successful logon followed by a session termination. Should this occur, increase the value until the session no longer terminates. For information about the PROTOCOL.INI file, see *Quick Beginnings*.
 - If your network does not make use of the messaging facility, eliminate it by removing **messenger** from the Services section of the OS/2 workstation’s IBMLAN.INI file. For a DOS workstation, the Messenger service is started with the NET START RCV command. For no messaging, you can start DOS LAN Services with the NET START RDR command.

Memory Minimization Considerations

Quick Beginnings recommends minimum memory sizes for the server and requester workstations. Memory requirements are provided for the base operating system, server, and requester installed with the defaults. You must also include estimates for memory required by the applications running in the requester workstation and the server workstation, if applicable. The recommendations provided in Table 1 on page 14 increase the estimated server memory requirements, which are based on the default parameter values.

If the workstations are configured with less than the recommended memory, the following sections provide guidance in reducing the memory required by the server and requester. To ensure that significant memory swapping does not occur, it is recommended that you use the memory analysis capabilities of SPM/2 2.0.

Server IBMLAN.INI Memory Parameters

The IBMLAN.INI file parameters that affect the amount of memory required in the server are as follows:

- **numbigbuf**
 In LAN Server, the size of a big buffer is fixed at 64KB, therefore big buffers can use a large amount of memory. To check how often the server has used all the big buffers allotted, use the NET STATISTICS command. If the big buffers resource is never exhausted, reduce the **numbigbuf** parameter value on the server and check the network statistics again. Repeat this process until the server occasionally uses all the big buffer resource during peak workload times. The value that you set for the **numbigbuf** parameter involves a memory-versus-performance trade-off. You must determine the value that best satisfies the needs of your network.

In the 386 HPFS option, 64KB buffers (up to 64) are allocated as needed. Therefore, the **numbigbuf** parameter on the server should be set to 0 if the server only accesses 386 HPFS option files and does not do any printing.

- **numreqbuf**

To check how often the server has used all the request buffers allotted, use the NET STATISTICS command. If the request buffers resource is never exhausted, reduce the **numreqbuf** parameter value on the server and check the network statistics again. Repeat this process until the server occasionally uses all the request buffer resource during peak workload times. The value that you set for the **numreqbuf** parameter involves a memory-versus-performance trade-off. You must determine the value that best satisfies the needs of your network. For more information, see **numreqbuf** in the Tuning Parameters section 57.

- **sizreqbuf**

The value of the **sizreqbuf** parameter can be reduced from the 4KB default value to 2KB in some environments without causing significant performance degradation. Compare the network performance for both settings before making the final choice. Setting the **sizreqbuf** parameter to 2KB can affect the number of big buffers needed because a request to read or write with a record size greater than 2KB, instead of 4KB, will now attempt to use the big buffers. It is strongly recommended that the **sizreqbuf** parameter value on the server match the **sizworkbuf** parameter value on the requester.

Server PROTOCOL.INI Memory Parameters

The **Maximum sessions**, **Maximum commands**, and **Maximum names** parameter values must each be set to a value that is greater than or equal to the corresponding *x1*, *x2*, and *x3* values on the associated **netx** statement.

Server CONFIG.SYS Memory Parameters

The **cache /C** on HPFS.IFS line and **diskcache** parameters in the CONFIG.SYS file on the server can be adjusted depending on the LAN Server package used.

In LAN Server, reduce the value of the **diskcache** and **cache /C** on HPFS.IFS parameters. The threshold for DISKCACHE should be no larger than 1/10 the size of the cache. For example, set the **diskcache** parameter to 128,24. Reducing the cache size can severely degrade performance.

In the 386 HPFS option, use the CACHE386 utility with the /STATS option to examine the read hit and miss ratio while changing the value that sets the size of the cache in the HPFS386.INI file.

OS/2 LAN Requester IBMLAN.INI Memory Parameters

The IBMLAN.INI file parameters that affect the amount of memory required in an OS/2 requester are as follows:

- **numworkbuf**

Unless the requester workstation is running multiple OS/2 sessions with the server, the **numworkbuf** parameter can usually be reduced from the default value of 15 to 5 without severely degrading performance.

- **sizworkbuf**

The value of the **sizworkbuf** parameter can be reduced from the default value of 4KB to 2KB without severely degrading performance. Compare the performance

of your network with both settings before making the final choice. The **sizworkbuf** parameter value in the requester should match the **sizreqbuf** parameter value in the server.

- **maxwrkcache**

Usually, you should not reduce the **maxwrkcache** parameter value because it significantly affects program loading and file transfers. If the performance degradation is acceptable, the **maxwrkcache** parameter value can be reduced from the default value of 64KB to 0.

OS/2 LAN Requester CONFIG.SYS Memory Parameters

To reduce the memory requirements on an OS/2 requester workstation, set the **protectonly** parameter to YES if DOS real mode is not required. If DOS real mode is required, set the **memman** parameter to SWAP,MOVE,SWAPDOS to enable swapping of the DOS real-mode window.

Additional Related Parameters

Several files contain related parameters that do not significantly affect memory usage. The related parameters do not usually need to be adjusted. The files that contain related parameters are as follows:

- OS/2 LAN Requester PROTOCOL.INI
- DOS LAN Services NETWORK.INI
- DOS LAN Services PROTOCOL.INI
- DOS LAN Services CONFIG.SYS

Chapter 3. IBMLAN.INI File Parameter Descriptions

You can adapt LAN Server to meet the needs of individual users on the network. For example, a network with few users may require a network definition different from that of a network with many users. You can alter the LAN Server definition by changing parameter values in the IBMLAN.INI file, which is in the \IBMLAN subdirectory.

Before changing any defaults, try running your network's workstations using the default values in the IBMLAN.INI file. Default values are used for each parameter that is not specified in the IBMLAN.INI file. If the default values do not meet your network needs, adjust the parameter values.

Changing IBMLAN.INI Parameter Values

Many entries in the IBMLAN.INI file can be overridden for a logon session using the full-screen interface, the NET START command, or the NET CONFIG command. Other parameters cannot be overridden. To change these parameters or make permanent changes to any parameters, you must edit the IBMLAN.INI file. You can edit the IBMLAN.INI file with an ASCII text editor.

When you edit the IBMLAN.INI file, keep in mind the following:

- Do not change any default values in the Services section.
- While you can change values in sections other than Services, do not delete any parameters or attempt to change parameter names. Parameter entries have the form *parameter=value*. Change only the *value* portion.
- You can add comments to the IBMLAN.INI file. Comments must begin with a semicolon (;) in the first column.
- LAN Server values in lists are separated by commas. Exceptions are the Replicator section and the **auditing** parameter in the Server section, where list values are separated by semicolons.

Activating Changed Parameter Values

Most changes in the IBMLAN.INI file take effect when the service corresponding to the changed section is restarted. For example, changes to the **keepconn** parameter value in the Requester section take effect the next time the requester is started. (Restart network services using the NET START command.)

The following IBMLAN.INI file parameters are used when the workstation starts. If you change the values of any of these parameters, you must restart your workstation for the change to take effect.

- **netx** parameter in the Networks section
- **maxcmds** parameter in the Requester section
- **maxthreads** parameter in the Requester section

When LAN Server first starts, or when it needs more information, it checks the IBMLAN.INI file, which contains all the default information needed by LAN Server.

Parameter Descriptions

The following sections reflect the organization of the IBMLAN.INI file:

- Networks
- Requester
- Messenger
- Server (for servers only)
- Alerter (for servers only)
- Netrun (for servers only)
- Replicator
- Netlogon (for servers only)
- UPS (for servers only)
- Remoteboot (for servers only)
- LSserver
- Services
- DCDB Replicator
- Peer (for OS/2 workstations only)

Note:

DLS Peer parameters are in the NETWORK.INI file. The Generic Alerter service and the Timesource service do not have separate sections in the IBMLAN.INI file.

Parameters are listed with minimum, maximum, and default values. The maximum value given for each parameter represents a level above which the related service will not start successfully. Resource constraints can further reduce the maximum value for each parameter. For example, the **charcount** parameter in the Requester section has a maximum value of 65535; however, this may be an unreasonably large value, depending on your server's serial devices. If you specify a value over the maximum, the corresponding service will fail to start.

Networks Section

The Networks section of the IBMLAN.INI file defines the network adapters that the workstation can access. The Networks section contains the **netx** statements that define the network adapters. If LAN Server is run on more than one installed network adapter, there must be a **netx** statement in the Networks section of the IBMLAN.INI file for each of the installed network adapters.

The **netx** parameter is the only parameter in the Networks section of the IBMLAN.INI file. The *x* variable is a sequence number used to distinguish each **netx** statement. The format of the **netx** statement is:

netx = *drivername* \$, *a*, *drivertype*, *x1*, *x2*, *x3*

Additional variables in the **netx** statement define the name of the protocol driver and the number of the network adapter driver that the workstation uses when communicating on the network adapter defined by a particular **netx** statement. The

remaining variable values indicate the NetBIOS resources needed to start the workstation. The number of resources indicated in the **netx** statement must be available on the workstation.

If NetBIOS for TCP/IP is installed on a workstation, the **netx** parameter defines TCPBEUI\$ and LM10 values for a ring 0 server.

If a loop-back driver is installed on a workstation, the **net1** statement defines the loop-back driver and the associated variable values. Also, the loop-back driver requires that the **net1** parameter be listed first on the **srvnets** and **wrknets** statements. For more information about the format of the statements, refer to **srvnets** and **wrknets** parameters.

The following are the default **netx** statements:

- For a loop-back driver: NET1=LOOPDRV\$, 0
- For a server or requester: NET X =NETBEUI\$, 0, LM10, 32, 50, 14

The variables contained in the **netx** statements can be grouped into two categories: system variables and capacity variables. The following lists specify the variables for the **netx** statements in the Networks section of the IBMLAN.INI file.

- **System Variables**
 - *x*
 - *drivename\$*
 - *a*
 - *drivertype*
- **Capacity Variables**
 - *x1*
 - *x2*
 - *x3*

System Variables

The following list contains the system variables in the **netx** statements in the Networks section of the IBMLAN.INI file:

- *x*

This variable is a sequence number used to distinguish each **netx** statement.

Default value: 1

Minimum value: 1

Maximum value: 5
- *drivename\$*

This variable indicates the protocol driver name. The value of this variable should be netbeui\$, loopdrv\$, snabeui\$, or tcpbeui\$.
- *a*

This variable indicates the number of the network adapter driver on the **bindings** parameter statement in the specified protocol section of the PROTOCOL.INI file. The protocol section is determined by the driver name specified for the *protocol\$* variable. The *a* variable cannot have the same value in two different **netx** statements that have the same driver name specified.

Default value: 0

Minimum value: 0

Maximum value: 3

- *drivertype*

This variable indicates the protocol driver type. This value should be set to LM10 or NB30 for TCP/IP.

Capacity Variables

The following list contains the capacity variables in the **netx** statements in the Networks section of the IBMLAN.INI file:

- *x1*

This variable indicates the number of NetBIOS sessions the requester or server allocates. Changing this value increases or decreases the number of users and servers on the network defined by the **netx** statement with which this workstation can communicate. Each session can have multiple connections. Connections include logons and NET USE commands. This parameter must be set to a value that is less than or equal to the **Maximum sessions (sessions)** parameter value in the NetBEUI_NIF section of the PROTOCOL.INI file.

Default value: 93 (LAN Server); 102 (386 HPFS Option)

Minimum value: 2

Maximum value: 254

- *x2*

This variable indicates the number of simultaneous NetBIOS commands (network control blocks) a requester or server can post. For a server, changing this value can increase or decrease the number of requester requests it can process at once. If a value less than 16 is specified, the value of 16 is used. This parameter must be set to a value that is less than or equal to the **Maximum commands (ncbs)** parameter value in the NetBEUI_NIF section of the PROTOCOL.INI file.

Default value: 212 (LAN Server); 225 (386 HPFS Option)

Minimum value: 16

Maximum value: 255

- *x3*

This variable indicates the number of NetBIOS names the requester allocates. The requester uses NetBIOS names for the computer name and messaging names. To add more messaging names, increase the value of this variable. If this value is greater than 10, start OS/2 LAN Requester and the Messenger service before other NetBIOS applications. This parameter must be set to a value that is less than or equal to the **Maximum names (names)** parameter value in the NetBEUI_NIF section of the PROTOCOL.INI file.

Default value: 14

Minimum value: 5

Maximum value: 254

Requester Section

Requester parameters specify configuration values for requester memory allocation and for programs that start with the Requester service. The parameters contained in the Requester section can be grouped into three categories: system parameters, capacity parameters, and tuning parameters. The following lists specify the parameters contained in the Requester section of the IBMLAN.INI file.

- **System Parameters**

- **computername**

- domain
- othdomains
- useallmem
- wrknets
- wrkservices
- Capacity Parameters
 - maxcmds
 - maxerrorlog
 - maxthreads
 - numalerts
 - numdgrambuf
 - numservices
 - sizerror
- TuningParameters
 - charcount
 - chartime
 - charwait
 - keepconn
 - keepsearch
 - maxwrkcache
 - numworkbuf
 - printbuftime
 - sesstimeout
 - sizworkbuf
 - wrkheuristics

System Parameters

The following list contains the system parameters in the Requester section of the IBMLAN.INI file:

- **computername**

This parameter indicates the requester workstation name. The workstation name can be from 1 to 15 characters (bytes) in length and can contain any characters except embedded blanks and the following:

" / \ [] ; : | < > + = , ? *

If two or more non-bridged networks have workstations (servers or requesters) configured for multiple adapters, the **computername** parameter of each workstation residing on those networks *must* be unique in order to ensure valid connections to those workstations.

- **domain**

This parameter indicates the default domain name used when logging on from a requester. The domain name cannot be the same as any computer name on the network, including the local workstation. The domain name can be from 1 to 8 characters (bytes) in length and can contain any characters except embedded blanks and the following:

" / \ [] ; : | < > + = , ? *

The NET VIEW command lists all the servers on the domains that are listed on the **domain** and **othdomains** parameter statements. The NET VIEW command

also lists the servers on the domain that the user logged on to before issuing the command. For example, a user starts a requester that has the **domain** parameter set to DOMAIN1 and the **othdomains** parameter set to DOMAIN2, DOMAIN3. Then the user logs on to DOMAIN4 and issues the NET VIEW command. The workstation lists all the servers on DOMAIN1, DOMAIN2, DOMAIN3, and DOMAIN4.

- **othdomains**

This parameter specifies up to four additional domains in which a requester can participate. For example, you can specify DOMAIN1, DOMAIN2. The default IBMLAN.INI file includes the **othdomains** parameter without any domains listed on the right side of the equal sign (=). For information about the NET VIEW command, refer to the **domain** parameter or to the *Command Reference*

- **useallmem**

If a requester machine is configured with more than 16MB, setting **useallmem** to YES allows the system to restrict or lock memory used by the LAN adapters (REQBUFs) to above the 16MB boundary. Most machines with more than 16MB of memory should set **useallmem** to YES to provide more usable memory below the 16MB boundary. This becomes important when the total amount of memory used by REQBUFs is large (**numreqbuf*sizreqbuf**).

The default setting for **useallmem** provided by the LAN Server installation/configuration program is YES. If you have less than 16MB of memory on your machine, it does not matter whether you set **useallmem** to either YES or NO. A YES setting is ignored if less than 16MB of memory is available.

Certain adapters that use DMA or busmaster processing will not run properly if **useallmem** is set to YES, regardless of the amount of memory available on the system. For example, the IBM Token Ring Busmaster Server/A will not run correctly when **useallmem** is set to YES because the adapter uses 24-bit DMA, which does not support using memory above 16MB. For these adapters, **useallmem** must be set to NO, otherwise, problems may occur. The LAN Server Installation/Configuration program will detect known problem adapters and set **useallmem** to NO before completion.

Default value: YES

- **wrknets**

This parameter lists names of networks, separated by commas, on which the requester runs. Names of available networks are listed in the Networks section of the IBMLAN.INI file.

Default value: net1

- **wrkservices**

This parameter specifies network services to start with the Requester service. For example, the Messenger service, which sends and receives network messages, can be started with the Requester service. Any valid service defined in the Services section of the IBMLAN.INI file can be started with the Requester service. Multiple services must be separated by commas. This value is defined by the user at installation. For more information on valid services, see "Services Section" on page 85.

Capacity Parameters

The following list contains the capacity parameters in the Requester section of the IBMLAN.INI file:

- **maxcmds**

This parameter sets the maximum number of NetBIOS commands the requester can send to the workstation's network adapters simultaneously. Increase the value of this parameter if you have multiple applications on the requester simultaneously using LAN Server. Because command processing uses memory, do not specify a number higher than you need. The recommended value is 1.6 times the value specified for the **maxthreads** parameter.

If you change this value, you must restart your workstation to make the change effective.

Default value: 16

Minimum value: 5

Maximum value: 255

- **maxerrorlog**

This parameter sets the maximum size, in kilobytes, of the error log. The **maxerrorlog** parameter keeps the error log from filling up your hard disk. Reduce the **maxerrorlog** parameter value if you need disk space more than you need extensive error information.

Default value: 100

Minimum value: 2

Maximum value: Restricted by the size of the hard disk.

- **maxthreads**

This parameter sets the maximum number of threads within a requester that are available to handle simultaneous network requests. The actual number of threads is based on the value of the **maxthreads**, **maxcmds**, and **numworkbuf** parameters. If the requester has multiple applications simultaneously using LAN Server, increase the value of the three parameters. Because each thread reduces the amount of available memory, do not allocate more than you need. The **numworkbuf** and **maxcmds** parameter values should be set greater than or equal to the **maxthreads** parameter value.

If you change this value, you must restart your workstation to make the change effective.

Default value: 10

Minimum value: 10

Maximum value: 254

- **numalerts**

This parameter sets the number of program tasks that can wait for an alert condition. Increase the **numalerts** parameter value only if you are using a server-based application with documentation directing you to increase the value. Because a larger value takes more memory (approximately 280 bytes for each increment), do not allocate more **numalerts** than you need.

Default value: 12

Minimum values: 5 for servers; 3 for requesters

Maximum value: 200

- **numcharbuf**

This parameter sets the number of character pipe and device buffers. Increase this value if you are using several shared serial devices or character pipes, or if you need to transmit large amounts of data across the network to such devices. Since each buffer takes up memory, do not allocate more than you need. Actual minimum and maximum values depend on the serial device. See the documentation about your serial device for more information.

Default value: 10

Minimum value: 0

Maximum value: 15

- **sizcharbuf**

This parameter sets the size, in bytes, of special read-ahead buffers used by the requester to improve the performance of read operations on character devices (redirected EOM ports and byte mode network pipes).

Default value: 512

Minimum value: 64

Maximum value: 4096

- **numdgrambuf**

This parameter sets the number of buffers for processing datagrams. Servers use datagrams to broadcast their presence on the domain. Datagrams are also used for domain-wide broadcasts. If you are on a network with many servers or with a large amount of domain-wide broadcasts, you may want more datagram buffers to handle incoming announcements.

Default value: 14

Minimum value: 8

Maximum value: 112

- **numservices**

This parameter sets the size of the internal service table. Increase the **numservices** parameter value if your requester runs many services. Because a larger table takes up memory, do not allocate more space than you need. Available services are listed in “Services Section” on page 85. The **numservices** parameter value should be greater than or equal to the number of entries in the Services section of the IBMLAN.INI file.

Default value: 7 (requesters); 16 (servers)

Minimum value: 4

Maximum value: 256

- **sizerror**

This parameter sets the size of the requester internal error buffer. Reduce the **sizerror** parameter value if you need more available memory and your network does not have frequent network errors.

Default value: 1024

Minimum value: 256

Maximum value: 4096

Tuning Parameters

The following list contains the tuning parameters in the Requester section of the IBMLAN.INI file:

- **charcount**

This parameter sets the number of characters, in bytes, that the requester stores before sending data to a serial device queue. Increase the **charcount** parameter value if you need to reduce traffic on your network. Actual minimum and maximum values depend on the serial device. For more information, see the documentation about your serial device.

Default value: 16

Minimum value: 0

Maximum value: 65535

- **chartime**

This parameter sets the time, in milliseconds, that the requester collects data to send to a shared serial device queue. Increase the **chartime** parameter value if you need to limit network use. If the **chartime** parameter value is too low, network efficiency decreases. Actual minimum and maximum values depend on the serial device. For more information concerning valid minimum and maximum values, see your serial device documentation.

Default value: 250

Minimum value: -1 (This value means the contents of the requester buffer are sent when the buffer has reached the size specified in the **charcount** parameter.)

Maximum value: 65535

- **charwait**

This parameter sets the time, in seconds, that the requester waits for a shared serial device to become available. Increase the **charwait** parameter value if you have heavy use of shared serial devices and can afford to wait. Actual minimum and maximum values depend on the serial device. For more information, see your serial device documentation.

Default value: 3600

Minimum value: 0

Maximum value: 65535

- **keepconn**

This parameter sets the amount of time, in seconds, that the requester maintains an inactive universal naming convention (UNC) connection to a shared resource. The value of this parameter specifies the amount of time that a UNC connection should remain dormant before the UNC connection is actually disconnected. The **keepconn** parameter applies only to implicit UNC connections (those resource connections called by an application without a prior NET USE). If there are no accesses to this resource during the interval specified by the **keepconn** parameter, the UNC connection is disconnected.

Default value: 600

Minimum value: 1

Maximum value: 65535

- **keepsearch**

This parameter sets the amount of time, in seconds, that the requester maintains an inactive file search request.

Default value: 600

Minimum value: 1

Maximum value: 65535

- **maxwrkcache**

This parameter sets the maximum size, in kilobytes, of the requester's large-transfer buffers. Increase the **maxwrkcache** parameter value if you use your requester for file-intensive projects, such as copying large files, and need better performance. The value of this parameter must be a multiple of 64 in order to match the size of the corresponding big buffer on the server. Big buffers on a server workstation have a set size of 64KB. The number of big buffers on the server is specified by the **numbigbuf** parameter.

Increasing the **maxwrkcache** parameter value may be effective only when a requester is running multiple applications against the server. In addition, each

increment uses an additional 64KB of the requester's memory. Therefore, the default value of this parameter should usually be used.

Default value: 64

Minimum value: 0

Maximum value: 640

- **numworkbuf**

This parameter sets the number of buffers the requester can use to store data for transmissions both to and from the server. These buffers are used in constructing the SMBs that are sent to the server. These buffers also provide data buffering between the application running in the requester and the network adapter card.

This parameter does not have a large impact on performance and therefore should rarely be changed. However, it may help requester performance to increase the value of the **numworkbuf**, **maxthreads**, and **maxcmds** parameters if there are multiple applications on the requester accessing server resources. Because each buffer reduces the amount of available memory, do not allocate more than you need. The **numworkbuf** and **maxcmds** parameter values should be set greater than or equal to the **maxthreads** parameter value.

Default value: 15

Minimum value: 3

Maximum value: 50

- **printbuftime**

This parameter sets the time limit, in seconds, for truncating a DOS print job.

Default value: 90

Minimum value: 0

Maximum value: 65535

- **sesstimeout**

This parameter specifies the time, in seconds, the requester waits before disconnecting a session from a server that is not responding to a request. If the server workload is extremely heavy, it may be necessary to increase the value of this parameter to prevent an inappropriate session termination.

Default value: 45

Minimum value: 10

Maximum value: 65535

- **sizworkbuf**

This parameter sets the size, in bytes, of requester buffers. This value should be a multiple of 512. It should be the same for every requester on the network and equal to the **sizreqbuf** value used by servers. Because the value of the **sizworkbuf** parameter should match all corresponding buffers on the network, it is recommended that the default value not be changed in most circumstances.

Default value: 4096

Minimum value: 1024

Maximum value: 16384

- **wrkheuristics**

This parameter sets a variety of requester fine-tuning options. Each digit of the **wrkheuristics** parameter has an independent meaning. When digits are missing from the right-hand side of the parameter value, the system supplies the default specified for each missing digit.

Except where noted, each digit of the **wrkheuristics** parameter is a binary digit: 0 means off (inactive); 1 means on (active). Other values are defined following the description of each digit.

	1	2	3	4
Digit position:	0123456789012345678901234567890			
Default value:	1111111213111111100010111201112211212111			

The meanings of the digits of the **wrkheuristics** parameter are as follows:

Position	Meaning								
0	Requests opportunistic locking of files. The default is 1. When digit position 0 is active, it allows a file opened in deny-none sharing mode to be locked by the server. Buffering can then be used to enhance performance. However, the server cannot lock the file if there are outstanding access requests. The server assumes that the first requester is the only active process using that file and prevents a second requester from accessing the file until buffered data is flushed (written to disk) and the file is closed. For more information, refer to the srvheuristics parameter, digit position 0 59 and digit position 15 63.								
1	Specifies performance optimization for batch (.CMD) files. Digit position 0 of the wrkheuristics parameter (opportunistic locking) must be set to 1. The default is 1. When digit position 1 is active, a batch file on the server running on the requester is kept in the requester's buffer to prevent a request across the LAN for each line of the batch file. When digit position 1 is inactive, the batch file is opened and closed with each line processed. The buffer data is flushed when the batch file is closed. The value of digit position 1 can be set to 0 if two or more requesters want to run a batch file simultaneously. However, file opens and closes across the LAN can greatly degrade performance.								
2	Specifies asynchronous unlock and asynchronous write-unlock, as follows:								
	<table> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Never</td> </tr> <tr> <td>1</td> <td>Always</td> </tr> <tr> <td>2</td> <td>Only on a LAN server virtual circuit</td> </tr> </tbody> </table>	Value	Meaning	0	Never	1	Always	2	Only on a LAN server virtual circuit
Value	Meaning								
0	Never								
1	Always								
2	Only on a LAN server virtual circuit								

The default is 1.

A *virtual circuit* is a NetBIOS session connection to another workstation over a LAN. With digit position 2, files in the requester buffer are unlocked in the buffer, and processing continues without waiting for confirmation from the server. Any errors occurring at the server are reported later. Generally, the only errors that might occur are hard media errors, such as a full disk or loss of power to the server.

Asynchronous lock and asynchronous write-unlock are not used if the file was opened with the write-through flag set. The write-through flag, when active, means that any writes to the file

must be written through the cache to the fixed disk before returning control to the calling program. If data integrity is a primary concern and the write-through flag is not used, digit position 2 can be set to 0. Values 1 and 2 are functionally equivalent in a LAN Server environment.

- 3 Specifies asynchronous close and asynchronous write-close, as follows:

Value	Meaning
0	Never
1	Always
2	Only on a LAN Server virtual circuit

The default is 1.

Digit position 3 performs a function similar to digit position 2 of the **wrkheuristics** parameter. The server sends a completion message to the requester for close or write-close requests before writing buffered file data to the hard disk. The user is warned if all the data is not written to the hard disk. Asynchronous close and asynchronous write-close are not used if the file was opened with the write-through flag set.

If data integrity is a primary concern and the write-through flag is not used, digit position 3 can be set to 0. Values 1 and 2 are functionally equivalent in a LAN Server environment.

- 4 Buffers all named pipes and serial devices. The default is 1.

Digit position 4 causes named pipe and communication device information to be buffered by the requester while the workstation is reading the information. This provides a degree of protection for the data as well as a potential performance improvement.

- 5 Specifies combined read-lock and write-unlock, as follows:

Value	Meaning
0	Never
1	Always
2	Only on a LAN Server virtual circuit

The default is 1. When digit position 5 is active, the lock and read requests are joined and sent as one command. The write and unlock requests are joined similarly. Values 1 and 2 are functionally equivalent in a LAN Server environment.

- 6 Specifies open and read. The default is 1.

When digit position 6 is active, a request to open a file also performs a read of the size specified by the **sizworkbuf** parameter from the beginning of the file to the requester's work buffer.

- 7 Reserved.

- 8 Specifies the use of the chain-send NetBIOS network control block (NCB), as follows:

Value	Meaning
0	Never
1	When a server's buffer is larger than the workstation's buffer
2	Always (to avoid copy)

The default is 2.

A chained-send enables the NetBIOS protocol driver to copy large data blocks directly from the OS/2 LAN Requester user's buffer to the network adapter card. An intermediate copy is not sent to the network buffers, from which the NetBIOS protocol driver normally copies the data to the network adapter card.

In order for the Netlogon service to replicate user and group definitions across servers in a domain, digit position 8 of the **wrkheuristics** parameter (on additional servers) and digit position 3 of the **srvheuristics** parameter (on domain controllers) must not be set to 0. If these digit positions are set to 0, user and group data may not be replicated from the domain controller, resulting in unknown user IDs and group IDs on the additional servers.

- 9 Buffers small read and write requests until the buffer is full, as follows:

Value	Meaning
0	Never
1	Always
2	Only on a LAN Server virtual circuit

The default is 1.

When digit position 9 is active and file access mode allows, requests to read or write data smaller than that specified by the **sizworkbuf** parameter are performed locally in the requester's buffer. Buffering avoids additional trips across the LAN. The buffer is flushed when the file is closed or when the buffer is needed to satisfy other requests.

Digit position 9 can enhance performance for applications that read, modify, and write back small records. Values 1 and 2 are functionally equivalent in a LAN Server environment. Digit position 9 can be set to 0 if data integrity is a primary concern.

- 10 Specifies buffer mode (assuming shared access is granted), as follows:

Value	Meaning
0	Always read buffer-size amount of data if the request is smaller than the buffer size (sizworkbuf) and data is being read sequentially.
1	Use full buffer if file is open for reading and writing.

- 2 Use full buffer if reading and writing sequentially.
- 3 Buffer all requests smaller than the buffer size (if hits occur).

The default is 3.

Shared access means the file was opened in sharing mode. These options allow selective tuning of the buffer mode if any applications handle data in a manner conflicting with buffering.

- 11 Specifies RAW read and RAW write server message block (SMB) protocols. The default is 1.

RAW read and RAW write SMB protocols transfer data across the LAN without SMB headers. These protocols transfer large files directly between the server memory and a work cache in the requester.

Digit position 11 can significantly improve performance of large file transfers across the LAN. Digit position 11 must be active for digit positions 12 and 13 to be functional.

- 12 Specifies large RAW read-ahead buffer. The default is 1.

Digit positions 12 and 13 provide independent control over the use of RAW SMB protocol for read-ahead and write-behind, respectively. Both are active by default, but they can be turned off to better suit a particular environment. Digit position 11 must be active for digit positions 12 and 13 to be functional.

- 13 Specifies large RAW write-behind buffer. The default is 1.

Digit position 11 must be active for digit positions 12 and 13 to be functional. See digit positions 11 and 12 for more information.

- 14 Specifies *read multiplexing* SMB protocols. The default is 1.

This SMB protocol is used for large read requests if the RAW SMB protocol (digit position 11) is inactive. This protocol breaks transfers into buffer-size chunks (**sizworkbuf**) and chains them together to satisfy the request.

- 15 Specifies *write multiplexing* SMB protocols. The default is 1.

This SMB protocol is used for large write requests if the RAW SMB protocol (digit position 11) is inactive. This protocol divides transfers into buffer-size chunks (**sizworkbuf**) and chains them together to satisfy the request.

- 16 Specifies using big buffers for large core (non-RAW) reads. The default is 1.

Digit position 16 is pertinent only to requests made to a *coreserver*. An example of a core server is the PC LAN Program, which does not recognize certain SMB types added to the SMB command set by LAN server. PC LAN Program is not supported for Double-Byte Character Set (DBCS) machines.

- 17 Specifies same-size small record read-ahead or read-to-sector boundary. The default is 1.

When digit position 17 is active, requests to read small data records sequentially cause read-ahead in multiples of the data record size, so a full buffer is read and sent to the requester.

For example, if the user is reading 50-byte records sequentially from a 4096-byte buffer, LAN Server will read ahead to fill the buffer to 4050 bytes. When digit position 17 is set to 0, data is read up to the next sector boundary, usually 512 bytes.

Digit position 17 is significant only if digit position 9 of the **wrkheuristics** parameter is inactive. If the requester detects small data records of the same size being read sequentially, the requester performs this small record read-ahead operation.

18 Specifies same-size small record write-behind or write-to- sector boundary. The default is 0.

When digit position 18 is active, requests to write small data records cause write-behind in multiples of the data record size, so a full buffer is written to the server.

When digit position 18 is set to 0, data is written up to the next sector boundary, usually 512 bytes.

Digit position 18 is significant only if digit position 9 of the **wrkheuristics** parameter is inactive. If the server detects small data records of the same size being written sequentially, the server performs this small record write-behind operation.

19 Reserved. This digit position must be set to 0.

20 Flushes pipes and devices on a DosBufReset or DosClose operation. The default is 0.

Digit position 20 gives the requester application more flexibility regarding which files, pipes, or devices are flushed (written to disk) following DosBufReset or DosClose operations.

Value	Meaning
0	Flush only files and devices opened by the caller. Spin until flushed. Wait for confirmation before proceeding with other tasks.
1	Flush only files and devices opened by the caller. Flush only once. Do not wait for confirmation.
2	Flush all files and all input and output of short-term pipes and devices. Spin until flushed.
3	Flush all files and all input and output of short-term pipes and devices. Flush only once.
4	Flush all files and all input and output of all pipes and devices. Spin until flushed.
5	Flush all files and all input and output of all pipes and devices. Flush only once.

21 Supports LAN Server encryption of passwords. The default is 1.

22 Controls log entries for multiple occurrences of an error. A recurring error can fill up the error log; use digit position 22 to keep down the number of log entries. If the value is other than 0,

the 1st, 4th, 8th, 16th, and 32nd occurrences of an error are logged. After that, every subsequent 32nd occurrence is logged.

If the value of digit position 22 is other than 0, the value also defines the size of an error table. The table is a record of errors that have occurred. Each table entry consists of an error ID number and the number of occurrences of that error. If an error does not match an existing entry in the table and the table is full, the logging facility deletes the table entry that has the lowest number of occurrences and records the new error information. The table size refers to the number of different errors allowed at one time in the table. The default is 0.

Set the value as follows:

Value	Meaning
0	Log all occurrences.
1	Use error table, size 1.
2	Use error table, size 2.
3	Use error table, size 3.
4	Use error table, size 4.
5	Use error table, size 5.
6	Use error table, size 6.
7	Use error table, size 7.
8	Use error table, size 8.
9	Use error table, size 9.

Typing NET ERROR at the command-line prompt will display the errors in this table that have been written to the error log.

23 Buffers all files opened with deny-write sharing mode. The default is 1.

When digit position 23 is active, the server buffers all files opened with deny-write sharing mode, regardless of the access mode the requester used to open the file. The sharing mode and access mode are two different parameters of a DosOpen command. Digit position 23 deactivates buffering on this requester if an application does not work correctly with it.

24 Buffers all files opened with the read-only (R) attribute set. The default is 1.

When digit position 24 is active, the server buffers all files with the read-only attribute set. Only read access mode will successfully open a read-only file. The sharing mode and access mode are two different parameters of a DosOpen command. Digit position 24 deactivates buffering on this requester if an application does not work correctly with it.

25 Reads ahead when opening for execution. Reading an executable file sequentially is usually, but not always, faster. The default is 1.

The value of digit position 25 should be 1 if many executable files are loaded across the LAN. The load time for some application programs can decrease by more than 50%. Experiment with your particular program to determine which value is better (0 or 1).

26 Handles the Interrupt (Ctrl+C) key, as follows:

Value	Meaning
0	Allow no interrupts.
1	Allow interrupts only on long-term operations.
2	Always allow interrupts.

The default is 2.

27 Forces correct open mode when creating files on a *core server*, which is a DOS-based LAN server. The default is 0. Digit position 27 is reserved for a DBCS user.

28 Specifies the NetBIOS NoAck mode (transferring data without waiting for an acknowledgment), as follows:

Value	Meaning
0	NoAck is disabled.
1	NoAck is set on send only.
2	NoAck is set on receive only.
3	NoAck is set on send and receive.

The default is 1.

29 Sends data along with the SMB write-block RAW requests. The default is 1.

When digit position 29 is active, the requester sends a requester buffer of data (**sizworkbuf**) to the server with its request for RAW mode protocol for large file transfers.

30 Sends a message to the screen when the requester logs an error, as follows:

Value	Meaning
0	Never
1	On write-fault errors only (no time out)
2	On write-fault and internal errors only (no time out)
3	On all errors (no time out)
4	Reserved
5	On write-fault errors only (time out)
6	On write-fault and internal errors only (time out)
7	On all errors (time out)

The default is 1.

Values other than 1 are normally used for debugging purposes only.

31 Reserved. This digit position is not used by LAN Server.

32 Establishes the behavior of the OS/2 API calls DosBufReset and DosResetBuffer and the DOS INT21 API call, COMMIT FILE, on a redirected file (not pipes or devices). When the call to the API returns, the data in the buffer has been handled as follows:

Value	Meaning
0	Changed data in the buffer was sent from the requester to the server. The server has written the data to disk.
1	Changed data in the buffer was sent from the requester to the server. The server has not yet written the data to disk.
2	DosBufReset was ignored for the files.

The default is 2.

33 Establishes the time interval for performing logon validation from the domain controller. Digit position 33 can be set as follows:

Value	Meaning
0	5 seconds
1	15 seconds
2	30 seconds
3	45 seconds
4	60 seconds
5	90 seconds
6	2 minutes
7	4 minutes
8	8 minutes
9	15 minutes

The default is 1.

34 There is a difference in date validation for older Microsoft and IBM servers. This digit position establishes interoperability with older servers. The default is 0.

Value	Meaning
0	Attempts to validate dates with preference toward PCLP date format.
1	Assumes the date is in the MSNET format.
2	Unchanged. Value 2 assumes the date is correct.

35 Many DOS applications behave incorrectly when more than two gigabytes of free space is indicated by an API to be available on a partition. This digit position allows these applications to operate correctly.

Value	Meaning
0	Return the true value for free disk space on the partition. The default is 0.
1	Always return a value for free disk space that is less than two gigabytes.

36 **timesync** performs time and date synchronization with the domain controller at logon. Digit position 36 can be set as follows:

Value	Meaning
0	no
1	yes

The default is 1 (yes).

37 **logonverification** specifies the type of verification that occurs during LAN Server logon. Digit position 37 can be set as follows:

Value	Meaning
0	Allows you to log on without your user ID and password being verified. The user ID and password are only stored.
1	Validates your user ID and password on your local workstation using the local copy of the user and definitions file (NET.ACC).
2	Validates your user ID and password on the domain logon server. The user ID and password are sent to a domain controller or backup server to be validated using the domain copy of the NET.ACC.

The default is 2 (domain).

38 **logonwarningmsgs** allows LAN Server logon warnings to be suppressed in some cases when the messages may be irritating to a user. Digit position 38 can be set as follows:

Value	Meaning
0	No warning messages are displayed.
1	All warning messages are displayed.
2	LAN Server-specific warning messages are not displayed.

The default is 1 (all).

39 Buffers all files opened in compatibility mode. Digit position 39 can be set as follows:

Value	Meaning
0	Buffer only files opened for read access in compatibility mode.
1	Buffer all files opened in compatibility mode.

The default is 1 (all files opened).

- 40** **Multiple logons** allow a user to log on multiple times in the same domain. Note that the prevention of multiple logons is true only when the user ID and the domain name are equal to or less than 8 characters each. For the longer user IDs or domain names, multiple logons are always allowed no matter how this digit is set. Digit position 40 can be set as follows:

Value	Meaning
0	Prevent multiple logons. If this digit is set to 0, the logon process will prevent a user from logging on more than once in the same domain. This is done by concatenating the user ID and the domain name and adding it as a unique NetBIOS name. A second user cannot add the same NetBIOS name.
1	Allow multiple logons. The default is 1.

- 41** **eapopup** returns errors to applications when EA setting fails due to backlevel server:

Value	Meaning
0	Always return ERROR_EAS_NOT_SUPPORTED when setting EAs to servers that do not understand EAs, i.e., Windows for Workgroups or Core servers.
1	Never return errors when setting EAs to backlevel servers. Act like the setting of extended attributes succeeded. This affects creating files or directories with extended attributes via DosOpen or DosMkDir, and setting extended attributes for an existing file or directory with DosSetFileInfo or DosSetPathInfo.

- 42** **RCVThd_Pri** Boosts the priority of the thread which receives and queues datagrams to time-critical.:

Value	Meaning
0	Do not alter the priority (default).
1	Set the priority to time-critical.

Server Section - Server Function and Peer Function

The parameters in the Server section establish values that affect the server functions. If these parameters are set too high, the server may fail to start and an error message indicating the possible configuration problems will be displayed.

Some of the parameters in the Server section of the IBMLAN.INI file work differently for the 386 HPFS option. The descriptions for all the Server section parameters are provided in this section. For descriptions of the parameters that function differently for the 386 HPFS option, see "Server Section - 386 HPFS Option" on page 65.

Note: When a server using the 386 HPFS option accesses a FAT file system partition, the server functions like a server using HPFS.

The Server section parameters that function differently on a workstation using the 386 HPFS option are:

- **maxlocks**
- **maxopens**
- **maxsearches**
- **numbigbuf**
- **srvheuristics**

The parameters contained in the Server section can be grouped into three categories: system parameters, capacity parameters, and tuning parameters. The following lists specify the parameters contained in the Server section of the IBMLAN.INI file.

- **System Parameters**
 - **accessalert**
 - **alertnames**
 - **auditing**
 - **autodisconnect**
 - **autopath**
 - **autoprofile**
 - **diskalert**
 - **erroralert**
 - **guestacct**
 - **logonalert**
 - **maxsessvcs**
 - **netioalert**
 - **othsrvnames**
 - **srvnets**
 - **srvservices**
- **Capacity Parameters**
 - **maxauditlog**
 - **maxchdevjob**
 - **maxchdevq**
 - **maxchdevs**
 - **maxlocks**
 - **maxopens**
 - **maxsearches**
 - **maxsessopens**
 - **maxsessreqs**
 - **maxshares**
 - **maxusers**
- **Tuning Parameters**
 - **alertsched**
 - **keepdossearch**
 - **numbigbuf**
 - **numfiletasks**
 - **numreqbuf**

- **sizreqbuf**
- **srvanddelta**
- **srvannounce**
- **srvheuristics**

System Parameters

The following list contains the system parameters in the Server section of the IBMLAN.INI file:

- **accessalert**

This parameter sets the number of attempted resource-access violations that can occur before the server sends an alert message. Set this parameter to 1 for immediate notification of an attempted access violation.

Default value: 5

Minimum value: 0

Maximum value: 65535

- **alertnames**

This parameter specifies the user or machine IDs that receive administrative alert messages. Do not specify group IDs. Separate IDs with commas. Alert-related broadcasts can also be limited by reducing the number of names specified by **alertnames** on the servers. This parameter does not support user or machine ID names with more than 16 characters.

- **auditing**

This parameter specifies which events to audit when the server is started. When you set the **auditing** parameter by using any of the following values, information about the events they represent is collected.

NET START SERVER **/AUDITING:YES** means an audit log is begun. The default is **AUDITING=NO** and there is no audit log. See *Command Reference* for further information regarding this command.

When you set the **/AUDITING** parameter to **YES**, the audit log records information about all of the event types represented by the following options.

When you specify **NO** with this parameter, auditing is stopped (this is the default). To record information about specific event types, you can set the **/AUDITING** parameter with one or more of the following options:

Note: It is not valid to include two options together when one implies the other (for example, **Use** and **Gooduse**).

Logon

Includes the **Netlogon** and **Sesslogon** options; records each time a user starts or stops one of the server's services. The **Logon** option is a superset of the **Netlogon** and **Sesslogon** options.

Logonlimit

Records each time a user exceeds logon hours for the user account.

Netlogon

Includes the **Badnetlogon** and **Goodnetlogon** options. Records each time a user logs on to the network. The **Netlogon** option is a superset of the **Badnetlogon** and **Goodnetlogon** options.

Badnetlogon

Records each time a user fails in an attempt to log on to the network.

Goodnetlogon

Records each time a user successfully logs on to the network.

Permissions

Records each time a user makes changes to the list of permissions for a file.

Resource

Records each time a user accesses a resource in a way that is defined in the auditing options for the resource.

Service

Records each time a user starts or stops one of the server's services.

Sesslogon

Includes the **Badsesslogon** and **Goodsesslogon** options. Records each time an attempt is made to start or end a session with the server. The **Sesslogon** option is a superset of the **Badsesslogon** and **Goodsesslogon** options.

Badsesslogon

Records each time a user fails to start a session with the server.

Goodsesslogon

Records each time a user starts a session with the server.

Use Includes the **Baduse** and **Gooduse** options. Records each time a user uses a shared resource. The **Use** option is a superset of the **Baduse** and **Gooduse** options.

Baduse

Records each time a user fails in an attempt to use a shared resource.

Gooduse

Records each time a user successfully uses a shared resource, except where the resource allows for an unlimited number of uses.

Userlist

Records each time a user makes changes to the user accounts database.

- **autodisconnect**

This parameter sets the time, in minutes, the server waits before disconnecting inactive sessions. To specify that the server does not disconnect inactive sessions automatically, set the **autodisconnect** value to -1. Also, the **cleanup** parameter defaults to YES when **autodisconnect** = -1. If you set the **autodisconnect** value to 0, unpredictable results occur.

Default value: 120

Minimum value: -1

Maximum value: 65535

- **autopath**

This parameter specifies the path to the profile that contains the OS/2 Remote IPL share information. The specified path can be an absolute path beginning with a drive letter or name of a file that exists in the \IBMLAN\PROFILES subdirectory. This parameter is not in the default IBMLAN.INI file, but it is added by the OS/2 Remote IPL installation program. The associated profile is also set up by the OS/2 Remote IPL installation program. The profile should not be edited by the user.

Default value: C:\IBMLAN\PROFILES\SRVAUTO.PRO

- **autoprofile**

This parameter determines whether the profile specified by the **autopath** parameter is run when the server starts. Specify Load to use the profile when the server starts. Specify None to not use the profile when the server starts.

Default value: Load

- **diskalert**

This parameter sets the minimum free disk space, in kilobytes, that the server allows before sending an alert message. All hard disks and partitions are monitored. If the server's disk space is a prime concern, increase the value of this parameter.

Default value: 5000

Minimum value: 0

Maximum value: 65535

- **erroralert**

This parameter specifies the number of errors that can occur before the server sends an alert message. If you need rapid notification of errors, reduce the value of this parameter.

The following events increment the error count, causing an error alert to be sent:

- An illegal server message block (SMB) sent from a client.
- A write-close error.
- Damage to the AT command schedule file.
- Running out of a predefined resource causes a resource error and increments the error count by 1. Resource errors could result from one of the following IBMLAN.INI file parameters being decreased:
 - **maxchdevjob**
 - **maxchdevq**
 - **maxchdevs**
 - **maxlocks**
 - **maxopens**
 - **maxsearches**
 - **maxsessopens**
 - **maxsessreqs**
 - **maxshares**
 - **maxusers**
 - **numbigbuf**

Default value: 5

Minimum value: 0

Maximum value: 65535

- **guestacct**

This parameter identifies a guest account that allows users to access resources in this domain without having their user IDs defined to the domain. Also, users can log on to the domain using the guest ID. The guest ID is managed (through User Profile Management and LAN Server) as any other user ID. This parameter is valid only on the domain controller.

Default value: GUEST

- **logonalert**
This parameter specifies the number of logon violations, during the interval specified by the **alertsched** parameter, before an alert is issued. This parameter is valid only on a domain controller or backup server.
Default value: 5
Minimum value: 0
Maximum value: 65535
- **maxsessvcs**
This parameter specifies the maximum number of virtual circuits the server can accept from a requester. This parameter must be set to 1.
- **netioalert**
This parameter sets the number of network input and output errors allowed, during the interval specified by the **alertsched** parameter, before the server sends an alert message.
Default value: 5
Minimum value: 0
Maximum value: 65535
- **othsrvnames**
This parameter specifies a list of up to 7 machine IDs by which the server is also known on the network. The primary computername should not be included in the list. Machine IDs should be separated with commas.
- **srvnets**
This parameter lists names of the networks on which the server runs. Names of available networks are listed in the Networks section of the IBMLAN.INI file.
Default value: net1
- **srvservices**
This parameter specifies which network services to start with the Server service. Its format is a list of services, separated by commas, and is defined by the user at installation. For more information and the names of the services that can be started with the **srvservices** parameter, see “Services Section” on page 85.

Capacity Parameters

The following list contains the capacity parameters in the Server section of the IBMLAN.INI file:

- **maxauditlog**
This parameter sets the maximum size, in kilobytes, of the audit log file. When the log file size reaches this value, no more entries are saved in the file. Saving entries resumes as soon as the log file is cleared. If you do not need extensive audit information, reduce the value of this parameter.
Default value: 100
Minimum value: 0
Maximum value: 65535
- **maxchdevjob**
This parameter specifies the maximum number of requests the server can accept for all serial device queues combined. If your shared serial devices are heavily used, increase the value of this parameter.
Default value: 48

Minimum value: 0
Maximum value: 65535

- **maxchdevq**

This parameter specifies the maximum number of shared serial device queues. If you want to share many serial device queues, increase the value of this parameter.

Default value: 2
Minimum value: 0
Maximum value: 65535

- **maxchdevs**

This parameter specifies the maximum number of shared serial devices. Set the value of this parameter to the number of serial devices the server shares with the network.

Default value: 16
Minimum value: 0
Maximum value: 16

- **maxlocks**

386 HPFS Users

Refer to “Server Section - 386 HPFS Option” on page 65 for the **maxlocks** parameter description as a reference for tuning a server with the 386 HPFS option. These parameters are still required on a server with the 386 HPFS option.

This parameter specifies the maximum number of file locks on the server. This is the maximum number of byte ranges (records) that may be locked by users on the server. Increase the value of this parameter if there is a large number of heavily used files. This parameter applies only to lock requests issued by DOS requesters.

Note: The maximum number of file locks applies only to networks that have DOS PC LAN Program requesters, which do not have DBCS hardware.

Default value: 64
Minimum value: 1
Maximum value: 8000

- **maxopens**

386 HPFS Users

Refer to “Server Section - 386 HPFS Option” on page 65 for the **maxopens** parameter description as a reference for tuning a server with the 386 HPFS option. These parameters are still required on a server with the 386 HPFS option.

This parameter specifies the maximum number of files, pipes, and devices the server can have open at one time. For example, the value of this parameter must be greater than or equal to 5 for a user opening five files. The value of this

parameter must also be greater than or equal to 5 for five users opening the same file. If many users access the server simultaneously, increase the value of this parameter.

Default value: 128 (Peer); 256 (LAN Server)

Minimum value: 1

Maximum value: 65535

- **maxsearches**

386 HPFS Users

Refer to “Server Section - 386 HPFS Option” on page 65 for the **maxsearches** parameter description as a reference for tuning a server with the 386 HPFS option. These parameters are still required on a server with the 386 HPFS option.

This parameter specifies the maximum number of directory searches the server can do simultaneously. These searches are executed when a user does a wildcard search of a directory; for example, DIR Y:\TEXTFILE.*. If the server's files are heavily used, increase the value of this parameter. See digit position 7 of the **srvheuristics** parameter 60 for more information about searches. The following equation can be used to calculate a value for the **maxsearches** parameter. Do not use the calculated value if it is less than the default value.

Default value: 50 (Peer); 150 LAN Server

Minimum value: 1

Maximum value: 16384

- **maxsessopens**

This parameter specifies the maximum number of files, pipes, and devices one requester can have open on the server. If many of the server resources are used simultaneously, increase the value of this parameter. The default for Peer or LAN Server functions is 80.

Note: The server uses some of the value specified with the **maxsessopens** parameter for internal processing, so the entire value specified with this parameter is not available to the user.

Default value: 80 (Peer, LAN Server)

Minimum value: 1

Maximum value: 32768 (must be less than **maxopens**)

- **maxsessreqs**

This parameter specifies the maximum number of resource requests one requester can have pending on the server. If users need to perform multiple tasks simultaneously on the server, increase the value of this parameter.

Default value: 50

Minimum value: 1

Maximum value: 65535

- **maxshares**

This parameter specifies the maximum number of resources the server can share with the network. For example, if one user is using five resources on the server, the value of this parameter must be at least 5; but if five users are using the same server resource, the value of this parameter need only be set to 1. If the server shares many resources, increase the value of this parameter. Certain older clients cannot enumerate large numbers of shares; thus, they may not be able to see the maximum number of shares.

Default value: 16 (Peer); 192 LAN Server; 192 (386 HPFS option)

Minimum value: 2

Maximum value: 1500

Note:

The number of shared resources displayed by the NET CONFIG SRV command will be different from the number specified with the **maxshares** parameter. This is because the number of shared resources displayed by the NET CONFIG SRV command also includes default system shares (ibmlan\$, admin\$, and so on), and one share for each partition on the server (a\$, b\$, and so on).

- **maxusers**

This parameter sets the maximum number of users who can use the server simultaneously. This equals the number of users who might issue a NET USE command to the server. A user who issues up to five NET USE commands counts as one user. Five users, each issuing a NET USE command to the same resource, count as five users. This value is the number of NetBIOS sessions on the server.

Default value: 5 (Peer); 100; 100 (386 HPFS option)

Minimum value: 1

Maximum value: 1000

(250 maximum for each adapter **netx** specified in the IBMLAN.INI file)

Tuning Parameters

The following list contains the tuning parameters in the Server section of the IBMLAN.INI file:

- **alertsched**

This parameter sets the interval, in minutes, at which the server checks for alert conditions and sends any needed alert messages. Because checking for alert conditions uses memory and computing power, reduce the number only if you need more frequent checking for alert conditions.

Default value: 5

Minimum value: 0

Maximum value: 65535

- **keepdossearch**

This parameter sets the amount of time, in seconds, that the server maintains an inactive search from a DOS client or from the DOS session of an OS/2 client. Decreasing the timeout maybe be necessary when requesters are running DOS applicaitons to allow the server to recycle search structures rather than allocate new ones.

Default value: 600

Minimum value: 1
Maximum value: 65535

- **numbigbuf**

386 HPFS Users

Refer to “Server Section - 386 HPFS Option” on page 65 for the **numbigbuf** parameter description as a reference for tuning a server with the 386 HPFS option. These parameters are still required on a server with the 386 HPFS option.

This parameter specifies the number of 64KB buffers the server uses for moving large files or large amounts of data. These buffers are used while loading programs and copying files from the server to the requester. Increase the value of this parameter if the server moves large files slowly. Because each big buffer takes up 64KB, do not allocate more big buffers than the server can support.

Ideally, you should have two big buffers for each simultaneous heavy-load user. For example, if you have ten users and three of them are copying large files at the same time, you should have six big buffers.

The **numbigbuf** parameter is related to digit positions 13, 17, and 18 of the **srvheuristics** parameter. The server is initialized with three big buffers. Digit position 13 specifies how many of the big buffers can be used for read-ahead. Additional big buffers are dynamically allocated as needed, up to the limit specified by the **numbigbuf** parameter. Digit position 17 of the **srvheuristics** parameter specifies the amount of time that dynamically allocated big buffers stay in memory. Digit position 18 of the **srvheuristics** parameter determines the amount of time that the server waits after failing to allocate additional big buffers before trying again. Dynamic allocation of additional big buffers (greater than **numbigbuf**) can occur only for write requests to the server, *not* for read requests.

Default value: 4 (Peer); 12; 12 (386 HPFS option)

Minimum value: 0

Maximum value: 80

- **numfiletasks**

This parameter specifies the number of concurrent processes that handle file and print requests from requesters. The file tasks (processes) are multithreaded, which allows multiple read/write requests to be processed concurrently.

There is a maximum of 48 threads per file task, which is usually sufficient. However, additional file tasks could be needed if about 45 requesters attempt to simultaneously access the same range of a file. The server logs a message in the error log if it exhausts its threads for a process. Specify the value of this parameter as 1, unless a message is logged in the error log indicating that the server exhausted its threads for a process.

Default value: 1

Minimum value: 1

Maximum value: 8

- **numreqbuf**

This parameter specifies the number of buffers (with a size determined by **sizreqbuf**) the server uses to take requests from requesters. This parameter should be one of the first parameters checked if there is a performance problem in the LAN server environment.

Determine the number of concurrent requesters served by the server. The server should have enough request buffers available to handle a peak requester workload. You should allocate two request buffers for each requester connected to the server, with two additional buffers for each named pipe being used. Allocating more than two buffers per requester will probably not increase performance and will take up memory on the workstation.

Default value: 10 (Peer); 200; 250 (386 HPFS option)

Minimum value: 5

Maximum value: 2000

- **sizreqbuf**

This parameter sets the size, in bytes, of the buffers the server uses to take requests from requesters. The value set for this parameter should be the same for every server on the network. The value of the **sizworkbuf** parameter on each requester should be set to the same value as the **sizreqbuf** parameter on the server.

Default: 4096

Minimum value: 1024

Maximum value: 32768

- **srvandelta**

This parameter specifies the time, in milliseconds, the server uses to vary its announce rate. Varying the announce rate distributes network traffic.

Default value: 3000

Minimum value: 0

Maximum value: 65535

- **srvannounce**

This parameter specifies the rate, in seconds, at which the server announces its presence on the network.

Default value: 180

Minimum value: 0

Maximum value: 65535

- **srvhidden**

This parameter specifies whether the server announces its presence on the network. When the **srvhidden** value is set to YES, it overrides **srvheuristic** 20.

Default value: NO

- **srvheuristics**

This parameter sets a variety of server fine-tuning options. Each digit of the **srvheuristics** parameter has an independent meaning. Except where noted, each digit of the **srvheuristics** parameter is a binary digit: 0 means off (inactive); 1 means on (active). Other values are defined following the description of each digit.

386 HPFS Option Users

If servers with the 386 HPFS option access only 386 HPFS disks, refer to the **srvheuristics** parameter description on page 66. The following description applies only to LAN Server and to servers with the 386 HPFS option added that access FAT file-system disks.

		1	2
Digit position:	0	1234567890	1234567890
Default value:	1	11110141111131100	13311

The meanings of the digits of the **srvheuristics** parameter are as follows:

Position	Meaning								
0	<p>Specifies opportunistic locking when opening files. Opportunistic locking allows the server to assume that the first requester of the file is the only active process using that file. The server buffers reads and writes of the file while preventing a second requester from accessing the file until the buffered data is flushed and the file is closed. The buffering can occur even if the user opened the file in deny-none sharing mode. The default is 1.</p> <p>For opportunistic locking to occur, both digit position 0 of the srvheuristics parameter on the server and digit position 0 of the wrkheuristics parameter on the requester must be active. See digit position 0 of the wrkheuristics parameter 39 and digit position 15 of the srvheuristics parameter 63 for more information.</p>								
1	<p>Specifies <i>read-ahead</i> when the requester is performing sequential access. Read-ahead means read additional data to attempt to determine what the requester may need. The default is 1.</p> <p>Digit position 1 pertains to reading ahead to the server's buffers (big buffers and requester buffers) from the file system and cache.</p> <table><thead><tr><th>Value</th><th>Meaning</th></tr></thead><tbody><tr><td>0</td><td>Do not use read-ahead.</td></tr><tr><td>1</td><td>Use single read-ahead thread.</td></tr><tr><td>2</td><td>Use asynchronous read-ahead thread.</td></tr></tbody></table>	Value	Meaning	0	Do not use read-ahead.	1	Use single read-ahead thread.	2	Use asynchronous read-ahead thread.
Value	Meaning								
0	Do not use read-ahead.								
1	Use single read-ahead thread.								
2	Use asynchronous read-ahead thread.								
2	<p>Specifies <i>write-behind</i>. Write-behind means tell the requester that a write is completed before actually performing the write. If the write generates an error, the error appears on a subsequent write. Files opened for write-through do not use write-behind. The default is 1.</p> <p>Digit position 2 pertains to writing behind from the server's buffers (big buffers and requester buffers) to the file system and cache.</p>								
3	<p>Specifies the use of the NetBIOS network control block (NCB) chain-send. The default is 1.</p> <p>A chain-send enables the NetBIOS protocol driver to copy large data blocks directly from server memory to the network adapter</p>								

card. An intermediate copy to the network buffers, from which the NetBIOS protocol driver normally copies the data to the network adapter card, is bypassed.

In order for the Netlogon service to replicate user and group definitions across servers in a domain, digit position 8 of the **wrkheuristics** parameter (on additional servers) and digit position 3 of the **srvheuristics** parameter (on domain controllers) must not be set to 0. If these digit positions are set to 0, user and group data may not be replicated from the domain controller, resulting in unknown user IDs and group IDs on the additional servers.

- 4 Checks all incoming SMBs for correct format. This is useful with mixed versions and brands of network software on the LAN. The default is 0.

To prevent wasted processor cycles in an OS/2 LAN server environment, do not change the default value.

- 5 Supports file control block (FCB) opens (collapses all FCB opens for a file to a single open). This is only useful for DOS applications on the network. The default is 1.

- 6 Sets the priority for the server. The possible priority values range from 0 for the highest priority to 9 for the lowest. See Table 3 . The default is 4.

Server priority determines how often other applications can access the processor. For example, changing the priority from 4 to 5 causes applications on the server to respond more quickly but slows response to requests from the network.

Table 3. Server Priority

Priority	Class of Priority	Level of Class
0	3 (Fixed High)	31
1	3	23
2	3	15
3	3	7
4	3	0
5	2 (Regular)	31
6	2	23
7	2	15
8	2	7
9	2	0

- 7 Automatically allocates additional memory (up to **maxsearches**) for directory searches if the allocated memory is not sufficient. If DOS requesters are on the network, set this position to 1. The default is 1.

Digit position 7 pertains to directory searches (DosFindFirst). Memory is allocated dynamically, instead of being committed when it is not needed.

- 8 Writes records to the audit log only when the *scavenger* wakes up on the interval set by digit position 10. The scavenger is a

high-priority server thread that monitors the network for errors, writes to the error log and audit log, disconnects inactive sessions, and sends alerts (see digit position 10 of the **srvheuristics** parameter).

When digit position 8 is set to 0, any write to the audit log wakes the scavenger. Digit position 10 controls the wake-up interval of the scavenger. The default is 1.

9 Does full buffering (as controlled by digit positions 1 and 2 of the **srvheuristics** parameter) when a file is opened with deny-write sharing mode. When digit position 9 is set to 0, deny-write access has no buffering for any requester using this server. For more information, see digit position 23 of the **wrkheuristics** parameter on page 44. The default is 1.

If an application breaks while buffering deny-write opened files, use digit position 9 to disable buffering for all requesters.

10 Sets the interval for the scavenger to wake up. The scavenger is a thread of the server process that performs the following tasks:

- Automatic disconnection of sessions (see the **autodisconnect**
- Sending administrative alerts (see the **alertsched** parameter on page 56)
- Writing to the audit log (see the **auditing** parameter on page 50 and digit position 8 of the **srvheuristics** parameter)

Set this entry as follows:

Value	Meaning
0	5 seconds
1	10 seconds
2	15 seconds
3	20 seconds
4	25 seconds
5	30 seconds
6	35 seconds
7	40 seconds
8	45 seconds
9	50 seconds

The default is 1. Digit position 8 can cause the scavenger to wake up at other times.

11 Allows compatibility-mode opens of certain types of files by translating them to sharing mode opens with deny-none. This is useful for sharing executable and other types of files.

Digit position 11 controls how strictly the server enforces compatibility opens for read-only. In the strictest sense of compatibility opening, if any file is opened with a sharing mode set, or if another session has that file open in compatibility mode, a compatibility-mode open of that file fails.

The settings of digit position 11 relax the strictness of compatibility opens. The first level allows different DOS LAN Services workstations to run the same programs. The second level extends to batch files. The third level translates compatibility-mode opens into deny-none sharing mode while maintaining access authority (read-only, write-only, or read-write). Not all applications support this mode of operation.

Values for digit position 11 of the **srvheuristics** parameter include:

Value	Meaning
0	Always use compatibility-mode opens.
1	Use deny-none sharing mode if read-only access to .EXE or .COM files is requested. Use compatibility mode for a .BAT file or if write access to .EXE or .COM files is requested.
2	Use deny-none sharing mode if read-only access to .EXE or .COM files is requested. Use deny-write sharing mode if read-only access to .BAT files is requested. Use compatibility mode if write access to .EXE, .COM, or .BAT files is requested.
3	Use deny-none sharing mode on all compatibility-mode opens. The default is 3.

12 Allows DOS LAN Services workstations to use a second NetBIOS session when sending printer requests. If digit position 12 is not set to 1, a second NetBIOS session ends any previous sessions set up for that DOS LAN Services workstation. If these sessions are used, ensure that there are enough NetBIOS sessions available on the server. Setting this parameter to 1 only allows the usage of additional sessions, it does not configure for the use of those sessions. The default is 1.

13 Sets the number of 64KB buffers (big buffers) used for read-ahead. Possible values are 0 to 9, where 0 means read-ahead is disabled. If set to a value larger than the **numbigbuf** parameter value, this digit position is reset to the **numbigbuf** parameter value minus 1.

Each 64KB buffer is divided into sixteen 4KB read-ahead buffers. If you are processing many files with small reads simultaneously, allocate more than one big buffer. Increasing the value of this parameter requires additional NetBIOS commands. The value of this digit position usually should not be set above 4 or 5. The default is 1.

Using 64KB (big buffers) for read-ahead involves a trade-off between large file transfers and small-record read and write operations. Provided there are two 64KB buffers remaining in the server for each requester doing concurrent large file transfers, you can use the remaining 64KB buffers for read-ahead without a penalty.

14 Converts incoming path specifications into the most basic format that LAN Server understands. This conversion includes changing

lowercase characters to uppercase characters and changing the slashes used in path names to backslashes (/ to \). The default is 0.

15

This digit position has two functions:

- It sets the time the server waits before transmitting an error message indicating accessed denied due to a previous opportunistic lock (see digit position 0 of the **srvheuristics** Set a longer time when the network is subject to long delays. The default value is 0. Digit position 15 values can be set as shown in Table 4.

Table 4. *Srvheuristics* Digit Position 15 Values and Meanings

Value	Oplock Timeout	NetBIOS Timeout
0 (default)	35 seconds	34 seconds
1	70 seconds	69 seconds
2	140 seconds	127 seconds
3	210 seconds	127 seconds
4	280 seconds	127 seconds
5	350 seconds	127 seconds
6	420 seconds	127 seconds
7	490 seconds	127 seconds
8	560 seconds	127 seconds
9	640 seconds	No NetBIOS timeout occurs

If a second requester requests opening an Oplocked file, the server notifies the first requester to flush buffers and close the file. If the first requester does not respond within the time defined by digit position 15, the server sends an Access denied message to the second requester.

The server can lock a file opened in deny-none sharing mode (as long as there are no other requests to access the file), so that buffering can be used to enhance performance. The server provides exclusive use of the file to the first requester, preventing the second requester from accessing the file until buffer data is flushed (written to disk) and the file is closed.

- The NetBIOS timeout is the length of time the server waits for an acknowledgment response from a requester being sent a NetBIOS message. If a response is not received prior to this timeout, the server disconnects the session to that requester. The default value of 34 seconds may not be long enough if the requester and server are separated by a congested bridge or slow telecommunications lines.

16

Validates the input/output controls (IOCTLs) across the network. When digit position 16 is set to 1, the server accepts only generic device IOCTLs (categories 01H, 05H, and 0BH). The default is 1.

Note: When digit position 16 is set to 0, the server can receive incorrect IOCTL pointers because of differences in device drivers between vendors. This can shut down the server.

Set digit position 16 to 0 when using certain device drivers, such as custom-built drivers.

- 17** Determines how long the server maintains unused, dynamic big buffers before freeing the memory. This digit can range from 0 through 9. The default is 3 (1 minute).

Value	Meaning
0	0 seconds
1	1 second
2	10 seconds
3	1 minute
4	5 minutes
5	10 minutes
6	20 minutes
7	40 minutes
8	1 hour
9	Maintain big buffers indefinitely.

- 18** Determines how long the server waits after failing to allocate a big buffer before trying again. The server processes the request it received using the request buffers if no big buffers are currently available. This digit can range from 0 to 5. The default is 3 (1 minute).

Value	Meaning
0	0 seconds
1	1 second
2	10 seconds
3	1 minute
4	5 minutes
5	10 minutes

- 19** Specifies RAW read and RAW write server message block (SMB) protocols. The default is 1.

RAW read and RAW write SMB protocols transfer data across the LAN without SMB headers. These protocols transfer large files directly between the server memory and a work cache in the requester.

Value	Meaning
0	Neither Read Block RAW nor Write Block RAW is supported.
1	Both Read Block RAW and Write Block RAW are supported.

- 20** Server responds to announcement requests.
Default is 1.

Value	Meaning
--------------	----------------

0	Servers ignore requests to announce themselves.
1	Servers respond to announcement requests.

Server Section - 386 HPFS Option

Some of the parameters in the Server section of the IBMLAN.INI file function differently for a 386 HPFS option. The descriptions of the parameters that have the same function in both LAN Server and a 386 HPFS option are located in “Server Section - Server Function and Peer Function” on page 48. The following section provides 386 HPFS option descriptions of the parameters that function differently for a 386 HPFS option.

The following parameter descriptions assume that a 386 HPFS option is used on a 386-based workstation. When a server with a 386 HPFS option accesses FAT file system partitions, the server with the 386 HPFS option functions like LAN Server. Defaults are provided for a server with a 386 HPFS option accessing FAT file systems.

The Server section parameters that function differently on a workstation using a 386 HPFS option can be grouped into two categories: capacity parameters and tuning parameters.

- **Capacity Parameters**
 - **maxlocks**
 - **maxopens**
 - **maxsearches**
- **Tuning Parameters**
 - **numbigbuf**
 - **srvheuristics**

Capacity Parameters

The following list contains the capacity parameters specific to 386 HPFS in the Server section of the 386 HPFS version of the IBMLAN.INI files:

- **maxlocks**
This parameter specifies the maximum number of locks the server can have on non-386 HPFS files. The number of locks permitted on 386 HPFS files is bounded by the amount of heap space the 386 HPFS has available. Each 386 HPFS file lock requires at least 30 bytes of heap space.
- **maxopens**
A server with a 386 HPFS option ignores this parameter and allocates handles dynamically. The maximum number of opens permitted on 386 HPFS files are as follows:

Opens for files
64K

For FAT file system accesses, this default is 256.

- **maxsearches**
Servers with 386 HPFS ignore this parameter. The maximum number of searches on 386 HPFS files is equal to the maximum number of opens for finds

plus the maximum number of opens for searches as defined in **maxopens**. For FAT file system accesses, this default is now 350.

Opens for finds
8192

Opens for searches
6144

Note: The first open file instance takes approximately 300 bytes from the heap. Each additional instance of the file opened takes approximately 60 bytes. For large numbers of open files, the lack of available physical memory may reduce the maximum number of open files.

The space for the finds and searches comes out of the same 8KB table. So, if you have allocated 6KB for searches, and the system uses all of that 6KB, then there will only be 2KB left for finds.

Tuning Parameters

The following list contains the 386 HPFS-specific tuning parameters in the Server section of the IBMLAN.INI file:

- **numbigbuf**

This parameter specifies the number of 64KB buffers LAN Server uses for moving large files or large amounts of data.

As long as users of the 386 HPFS server access only 386 HPFS disks, the 386 HPFS server does not use big buffers for large data transfers. However, some big buffer space is allocated if the **numbigbuf** parameter is not set to 0. Therefore, if users of the 386 HPFS server only perform file I/O on 386 HPFS disks, the **numbigbuf** parameter can be set to 0.

Printing operations are handled through LAN Server function that is installed with LAN Server and 386 HPFS. Therefore, 386 HPFS print servers with an attached printer may need a few big buffers to handle print operations.

If the 386 HPFS server accesses any FAT 9 or JFS file system disks, the **numbigbuf** parameter should be set as described in "Server Section - Server Function and Peer Function" on page 48.

Default value: 12

Minimum value: 0

Maximum value: 80

- **srvheuristics**

The following description applies to 386 HPFS servers that access only 386 HPFS disks. If the 386 HPFS server accesses FAT file system disks, refer to the **srvheuristics** parameter description on page 58.

	1	2
Digit position:	012345678901234567890	
Default value:	111101411113110013311	

This parameter sets a variety of server fine-tuning options. Each digit of the **srvheuristics** parameter has an independent meaning. Except where noted, each digit of the **srvheuristics** parameter is a binary digit: 0 means off (inactive); 1 means on (active). Other values are defined following the description of each digit. The meanings of the digits of the **srvheuristics** parameter are as follows:

Position	Meaning
----------	---------

0	Specifies opportunistic locking when opening files. Opportunistic
---	---

locking allows the server to assume that the first requester of the file is the only active process using that file. The server buffers the reads and writes of the file while preventing a second requester from accessing the file until the buffered data is flushed and the file is closed. The buffering can occur even if the user opened the file in deny-none sharing mode. The default is 1.

For opportunistic locking to occur, both digit position 0 of the **srvheuristics** parameter on the server and digit position 0 of the **wrkheuristics** parameter on the requester must be active. For more information, see digit position 0 of the **wrkheuristics** parameter on page 39 and digit position 15 of the **srvheuristics**.

- 1 Does not apply to 386 HPFS servers using HPFS partitions. See **srvheuristics** for LAN Server for meanings if using FAT partitions for file sharing with 386 HPFS. The default is 1.
- 2 Does not apply to 386 HPFS servers using HPFS partitions. See **srvheuristics** for LAN Server for meanings if using FAT partitions for file sharing with 386 HPFS. The default is 1.
- 3 Specifies the use of NetBIOS network control block (NCB) chain-send. The default is 1.
- 4 Does not apply to 386 HPFS servers using HPFS partitions. See **srvheuristics** for LAN Server for meanings if using FAT partitions for file sharing with 386 HPFS. The default is 0.
- 5 Supports file control block (FCB) opens (collapses all FCB opens for a file to a single open). This is only useful for DOS applications on the network. The default is 1.
- 6 Does not apply to 386 HPFS servers using HPFS partitions. See **srvheuristics** for LAN Server for meanings if using FAT partitions for file sharing with 386 HPFS. The default is 4.
- 7 Does not apply to 386 HPFS servers using HPFS partitions. See **srvheuristics** for LAN Server for meanings if using FAT partitions for file sharing with 386 HPFS. The default is 1.
- 8 Does not apply to 386 HPFS servers using HPFS partitions. See **srvheuristics** for LAN Server for meanings if using FAT partitions for file sharing with 386 HPFS. The default is 1.
- 9 Does not apply to 386 HPFS servers using HPFS partitions. See **srvheuristics** for LAN Server for meanings if using FAT partitions for file sharing with 386 HPFS. The default is 1.
- 10 Does not apply to 386 HPFS servers using HPFS partitions. See **srvheuristics** for LAN Server for meanings if using FAT partitions for file sharing with 386 HPFS. The default is 1.
- 11 Allows compatibility-mode opens of certain types of files by translating them to sharing mode opens with deny-none. This is useful for sharing executable and other types of files.

Digit position 11 controls how strictly the server enforces compatibility-mode opens for read-only. In the strictest sense of compatibility-mode opens, if any file is opened with a sharing mode set, or if another session has that file open in compatibility mode, a compatibility-mode open of that file fails.

The settings of digit position 11 relax the strictness of compatibility-mode opens. The 386 HPFS package handles this function differently than LAN Server package. There are effectively only two settings for this digit position. Refer to the value meanings described below. Not all applications support this mode of operation. The default is 3.

Values for digit position 11 of the **srvheuristics** parameter include:

Values	Meaning
0	Always use compatibility-mode opens.
1, 2	Workstation requests for read-only access will have their sharing mode translated to deny-write. Any requests for write access will be translated to compatibility mode, thereby preventing shared usage of the file.
3	Use deny-none sharing mode on all compatibility-mode opens.
12	Allows DOS LAN Services workstations to use a second NetBIOS session when sending printer requests. If digit position 12 is not set to 1, a second NetBIOS session ends any previous sessions set up for that DOS LAN Services workstation. If these sessions are used, ensure that there are enough NetBIOS sessions available on the server. Setting this parameter to 1 only allows the usage of additional sessions; it does not configure for the usage of those sessions. The default is 1.
13	Does not apply to 386 HPFS servers using HPFS partitions. See srvheuristics for LAN Server for meanings if using FAT partitions for file sharing with 386 HPFS. The default is 1.
14	Does not apply to 386 HPFS servers using HPFS partitions. See srvheuristics for LAN Server for meanings if using FAT partitions for file sharing with 386 HPFS. The default is 0.
15	This digit position has two functions: <ul style="list-style-type: none"> – It sets the time the server waits before transmitting an error message indicating access denied due to a previous opportunistic lock (see digit position 0 of the srvheuristics parameter on page 66). When the network is subject to long delays, set a longer time. The default value is 0. Digit position 15 values can be set as follows.

Table 5. 386 HPFS Option *Srvheuristics* Digit Position 15 Values and Meanings

Value	Oplock Timeout	NetBIOS Timeout
0	35 seconds	34 seconds
1	70 seconds	69 seconds
2	140 seconds	127 seconds
3	210 seconds	127 seconds
4	280 seconds	127 seconds
5	350 seconds	127 seconds
6	420 seconds	127 seconds

Table 5. 386 HPFS Option *Srvheuristics Digit Position 15 Values and Meanings* (continued)

Value	Oplock Timeout	NetBIOS Timeout
7	490 seconds	127 seconds
8	560 seconds	127 seconds
9	640 seconds	No NetBIOS timeout occurs

If a second requester requests opening an Oplocked file, the server notifies the first requester to flush buffers and close the file. If the first requester does not respond within the time specified, the server sends an Access denied message to the second requester.

The server can lock a file opened in deny-none sharing mode (as long as there are no other requests to access the file), so that buffering can be used to enhance performance. The server provides exclusive use of the file to the first requester, preventing the second requester from accessing the file until buffer data is flushed (written to disk) and the file is closed.

- The NetBIOS timeout is the length of time the server waits for an acknowledgment response from a requester being sent a NetBIOS message. If a response is not received prior to this timeout, the server disconnects the session to that requester. The default value of 34 seconds may not be long enough if the requester and server are separated by a congested bridge or slow telecommunications lines.

- 16** Does not apply to 386 HPFS servers using HPFS partitions. See *srvheuristics* for LAN Server for meanings if using FAT partitions for file sharing with 386 HPFS. The default is 1.
- 17** Does not apply to 386 HPFS servers using HPFS partitions. See *srvheuristics* for LAN Server for meanings if using FAT partitions for file sharing with 386 HPFS. The default is 3.
- 18** Does not apply to 386 HPFS servers using HPFS partitions. See *srvheuristics* for LAN Server for meanings if using FAT partitions for file sharing with 386 HPFS. The default is 3.
- 19** Specifies RAW read and RAW write server message block (SMB) protocols. The default is 1.

RAW read and RAW write SMB protocols transfer data across the LAN without SMB headers. These protocols transfer large files directly between the server memory and a work cache in the requester.

Digit	Meaning
0	Neither Read Block RAW nor Write Block RAW is supported.
1	Both Read Block RAW and Write Block RAW are supported.

- 20** Server responds to announcement requests. The default is 1.

Digit	Meaning
0	Servers ignore requests to announce themselves.

Netlogon Section

The Netlogon service copies the master user and group definitions file (NET.ACC) on the domain controller to servers. Netlogon is available only on servers.

Passwords can become unsynchronized when using the Netlogon service if you:

- Reinstall a default or backup NET.ACC file onto an additional server
- Reinstall a default or backup NET.ACC file onto the domain controller
- Inadvertently change an additional server's password on the domain controller without the new password being replicated to the additional server prior to the next time Netlogon uses it

To resynchronize the password for the additional server see *Network Administrator Tasks*

- **System Parameters**
 - **scripts**
 - **update**
- **Tuning Parameters**
 - **pulse**
 - **randomize**
 - **scanpause**
 - **scanperiod**
 - **scantime**
 - **update**

System Parameters

The following list contains the system parameters in the Netlogon section of the IBMLAN.INI file:

- **scripts**

This parameter specifies the path where the Netlogon service searches for the logon scripts. This parameter applies only to primary and backup domain controllers and must be present with a valid path name. The path name can be absolute or relative to the IBMLAN subdirectory.

Default value: C:\IBMLAN\REPL\IMPORT\SCRIPTS
- **update**

This parameter specifies whether to synchronize the Netlogon service with the primary domain controller at the next pulse. This parameter applies only to member and backup servers.

Default value: Yes

Tuning Parameters

The following list contains the tuning parameters in the Netlogon section of the IBMLAN.INI file:

- **pulse**

This parameter specifies the time interval, in seconds, between update notices. This parameter is valid only if the server's role is primary (domain controller). The server's role is set at installation or with the NET ACCOUNTS command.

Default value: 60
Minimum value: 60
Maximum value: 3600

- **randomize**

This parameter specifies the time period, in seconds, during which the member or backup servers should send a request at random to get changes after receiving a change notice. This prevents overloading the primary domain controller with multiple update requests. This parameter applies only to primary domain controllers.

This parameter is not included in your default IBMLAN.INI file. You can add it anywhere in the Netlogon section.

Default value: 30
Minimum value: 5
Maximum value: 120

- **scanpause**

This parameter specifies the amount of time, in seconds, that the domain controller waits before polling each user. This parameter can be used to slow down the pace of polling each user, thereby limiting the amount of datagram traffic on the network at any one time. Refer to the **scantime** parameter for more information about the domain controller polling.

Default value: 0
Minimum value: 0
Maximum value: 15

- **scanperiod**

This parameter specifies the time interval, in minutes, at which the domain controller polls all the users. The minimum value is 0; when set to 0, the domain controller does not poll users. This parameter is not needed often because NET WHO checks the session list. The **scanperiod** value is set to 5 automatically if the domain controller poll response is 1 through 4. If this parameter is set to 1440, the domain controller polls the users at the time specified by the **scantime** parameter. Refer to the **scantime** parameter for more information about the domain controller polling.

Default value: 15
Minimum value: 0 (the domain controller does not poll the users)
Maximum value: 1440 (24 hours)

- **scantime**

This parameter specifies the time at which polling occurs each day if the **scanperiod** parameter is set to 1440.

The Netlogon service on the domain controller uses a polling technique to determine whether users are still logged on to the domain. The polling ensures that the output of the NET WHO command is as accurate as possible. In a domain with many users logged on, the polling causes a large amount of datagram traffic on the network. The **scantime**, **scanperiod**, and **scanpause** parameters enable you to control how often the polling occurs.

Default value: 0:00 (midnight)
Minimum value: 0:00
Maximum value: 23:59

LSserver Section

The LSserver service provides logical server support, which includes sharing startup aliases, sharing the IBMLAN tree, and starting DOS LAN Services support. The parameters contained in the LSserver section can be grouped into two categories: system parameters and tuning parameters. The following lists specify the parameters contained in the LSserver section of the IBMLAN.INI file.

- **System Parameter**
 - **cleanup**
- **Tuning Parameter**
 - **srvpipes**

System Parameter

The following is the system parameter in the LSserver section of the IBMLAN.INI file:

- **cleanup**

This parameter specifies whether to clean up (disconnect) shares for *dynamic aliases*. A dynamic alias is any alias that is shared with the **As required by user** option in the interface. Typing NET USE *x aliasname* at a requester DOS or OS/2 command prompt also creates a share for a dynamic alias. *x* is a local device name and *aliasname* is the alias name for the shared resource.

If the **autodisconnect** parameter value is set to **-1**, the **cleanup** parameter defaults to **yes**. The active **cleanup** parameter disconnects shares for dynamic aliases that no longer have users. When the **autodisconnect** parameter is set to **-1**, idle sessions are never ended by the server; therefore, if the **cleanup** parameter is set to **N**, the number of shares outstanding for dynamic aliases at the server may increase to a point where the value of the **maxshares** parameter would need to be increased.

Setting the **autodisconnect** parameter to **0** is not advisable, as it leads to numerous error messages.

For Configuration, Installation, and Distribution (CID) users, CID installation will always set **cleanup** to **Yes** if the **autodisconnect** parameter is set to **-1** regardless of what the user specifies. This prevents the buildup of shares for aliases that are no longer being used.

If the **autodisconnect** parameter is not set to **-1**, you can set the **cleanup** parameter to suit your requirements. If the **autodisconnect** parameter is not **-1** and the **cleanup** parameter is set to **Y**, cleanup of a dynamic alias share will occur when there are no users using the alias. The server will also end an idle session and the dynamic aliases shared for that session when the **autodisconnect** timeout occurs (unless an alias is being used in another session). If the **cleanup** parameter is set to **N**, you can remove dynamic alias shares manually. Restarting the server also removes dynamic alias shares.

If the server is started and stopped frequently, users must disconnect and reconnect to dynamic aliases. To avoid this extra activity, use the **At server startup** option in the GUI to share critical alias resources. For information about using the GUI to share aliases, see *Network Administrator Tasks*.

Default value: **Y**

Tuning Parameter

The following is the tuning parameter in the LSserver section of the IBMLAN.INI file:

- **srvpipes**

This parameter sets the maximum number of pipes that the server uses. If many users log on simultaneously, increase the value of this parameter. The following equation can be used to calculate a value for the **srvpipes** parameter. Do not use the calculated value if it is less than the default value.

srvpipes = **maxusers** / 12 (up to the maximum value)

Default value: 3

Minimum value: 1

Maximum value: 20

Messenger Section

The parameters in the Messenger section specify the buffer size and file name of the log file used to receive messages. The parameters contained in the Messenger section can be grouped into two categories: system parameters and capacity parameters. The following lists specify the parameters contained in the Messenger section of the IBMLAN.INI file.

Note: Ensure that the Messenger service is stopped before you delete the message log file.

- **System Parameter**
 - **logfile**
- **Capacity Parameter**
 - **sizmessbuff**

System Parameter

The following is the system parameter in the Messenger section of the IBMLAN.INI file:

- **logfile**

This parameter specifies a file name within the \IBMLAN\LOGS subdirectory for the message log.

Default value: MESSAGES.LOG

Capacity Parameter

The following is the capacity parameter in the Messenger section of the IBMLAN.INI file:

- **sizmessbuf**

This parameter sets the buffer size, in bytes, for receiving network messages. If you receive numerous or long messages, increase the **sizmessbuf** parameter value. Because larger buffers require more memory, do not allocate larger buffers than you need.

Default value: 4096

Minimum value: 512

Maximum value: 62000

Alerter Section

The Alerter section of the IBMLAN.INI file contains the following capacity parameter:

- **sizalertbuf**

This parameter sets the size, in bytes, of the buffer used for administrative alerts.

Default value: 3072
Minimum value: 512
Maximum value: 16384

Netrun Section

The Netrun section controls the service that allows other workstations to run a program in this server's memory. The parameters contained in the Netrun section can be grouped into two categories: system parameters and capacity parameters. The following lists specify the parameters contained in the Netrun section of the IBMLAN.INI file.

- **System Parameter**
 - **runpath**
- **Capacity Parameter**
 - **maxruns**

System Parameters

The following is the system parameter in the Netrun section of the IBMLAN.INI file:

- **runpath**

This parameter sets the server path in which Netrun requests can operate. To run a program remotely on this server, the .EXE program file must exist in the path defined by the **runpath** parameter. Only programs in this run path can be run remotely. Path syntax is the same as for the OS/2 PATH command.

Default value: C:\ (root directory)

Capacity Parameter

The following is the capacity parameter in the Netrun section of the IBMLAN.INI file:

- **maxruns**

This parameter specifies the maximum number of netrun requests that can run simultaneously in the server's memory. This parameter value affects the **numreqbuf** parameter in the Server section. The **numreqbuf** parameter value must be at least five times greater than the **maxruns** parameter value.

Default value: 3
Minimum value: 1
Maximum value: 10

Replicator Section

The Replicator service lets you maintain identical sets of files and directories on different workstations. Servers can replicate data to other servers or requesters. The server sending the data is called an *exporter*. The requester or server receiving the data is called an *importer*. You can have any number of exporters and importers in your domain; however, two exporters should not replicate the same set of files. A server can be both an exporter and an importer, while a requester can only be an importer. For more information concerning the Replicator service, refer to *Network Administrator Tasks*.

In the IBMLAN.INI Replicator section, you set up an *export path* on the exporter and an *import path* on each importer. The Replicator service monitors the export directory. When you add, delete, or change a directory or file in the export path, the

Replicator service sends an update notice to its importers. The importer then obtains a copy of the changed files and directories. Therefore, each import path is an up-to-date copy of the export path.

Each parameter listed in the Replicator section must have a corresponding value, except the **password** parameter. The following lists specify which parameters apply to importers and which apply to exporters.

- **Importer Parameters**
 - **replicate importlist**
 - **importpath logon**
 - **password tryuser**
- **Exporter Parameters**
 - **replicate exportlist**
 - **exportpath guardtime**
 - **interval pulse random**

The parameters contained in the Replicator section can be grouped into two categories: system parameters and tuning parameters. The following lists specify the parameters contained in the Replicator section of the IBMLAN.INI file.

- **System Parameters**
 - **replicate**
 - **exportlist**
 - **exportpath**
 - **importlist**
 - **importpath**
 - **logon**
 - **password**
 - **tryuser**
- **Tuning Parameters**
 - **guardtime**
 - **interval**
 - **pulse**
 - **random**

System Parameters

The following list contains the system parameters in the Replicator section of the IBMLAN.INI file:

- **replicate**

This parameter specifies whether the workstation is an importer, an exporter, or both. Values are: Import, Export, and Both. If a server is an exporter, this value must be either Export or Both. If a server is an importer, this value must be either Import or Both. On requesters, only Import is valid.

Default value: Import
- **exportlist**

This parameter lists workstations and domains to receive announcements of changes to the export directory. Each name is separated by a semicolon and no spaces. For example:

```
exportlist = server1;server2;domain3
```

This parameter is not included in your default IBMLAN.INI file. You can add it anywhere in the Replicator section.

- When no **exportlist** parameter is specified, update notices are sent to importers in this exporter's domain.
- If a list is specified for this parameter, update notices are sent to each workstation and domain on the list.
- If you include a domain name, the exporter sends updates to workstations in that domain. In order to be able to receive the updates, those workstations must be set up as importers and specify the exporter or exporter's domain.

- **exportpath**

This parameter specifies the absolute path to the export directories. All files to be replicated must be in this path.

This parameter is not included in your default IBMLAN.INI file. When this parameter is missing, the default takes effect. You can add the parameter anywhere in the Replicator section. The default path is created at installation.

There is no restriction on the path name or location. If it does not already exist, you must create the path and set up specific subdirectories in the export path.

Default value: C:\IBMLAN\REPL\EXPORT

Note: When setting up a single server as both an exporter and an importer, use different export and import paths and ensure that neither path is a subdirectory of the other. Otherwise, files replicated to other importers cannot be distinguished from files replicated from other exporters.

- **importlist**

This parameter lists servers and domains that can send a directory to this importer. Each name is separated by a semicolon and no spaces; for example:

importlist = *server1;server2;domain3*

This parameter is not included in your default IBMLAN.INI file. You can add it anywhere in the Replicator section.

- When no **importlist** parameter is specified, the Replicator service checks for updates from all servers set up as exporters in the importer's domain.
- If a list is specified for this parameter, update notices from any specified exporters and domains are sent to the importer.
- If you include a domain name, the importer receives updates from servers in that domain that are set up as exporters for the importer or the importer's domain.

- **importpath**

This parameter specifies the absolute path to the import directories. All files are replicated to a subdirectory of this path. This can also be a remote path.

There is no restriction on the path name or location. If it does not already exist, you must create the path and set up specific subdirectories in the import path.

The default path is created at installation.

Default value: C:\IBMLAN\REPL\IMPORT

Note: When setting up a single server as both an exporter and an importer, use different export and import paths and ensure that neither path is a subdirectory of the other. Otherwise, files replicated to other importers cannot be distinguished from those replicated from other exporters.

- **logon**

This parameter specifies the user ID that the Replicator service uses to connect to the exporter when no one is logged on at the importer. This user ID is used with the password specified by the **password** parameter. If the **logon** parameter (the default) is not included, the Replicator service tries logging on using the machine ID of the importer as a user ID.

Ensure the user ID (or machine ID if the **logon** parameter is not included) is set up as a user through User Profile Management and has read and attributes permissions to the replicated directories through LAN Server (using either the full-screen interface or the NET ACCESS command).

This parameter is not included in your default IBMLAN.INI file. You can add it anywhere in the Replicator section.

- **password**

This parameter specifies the password that the Replicator service uses to connect to the exporter when no one is logged on to the importer. This password is used with the user ID specified by the **logon** parameter. The **password** parameter cannot be used unless the **logon** parameter is also specified. If the default setup for the **logon** parameter is used (that is, the **logon** parameter is not included and the Replicator service tries logging on with the machine ID), then a password cannot be specified. If a password is desired for the workstation, both the **logon** and **password** parameters must be used.

- **tryuser**

This parameter specifies whether the importer automatically tries to connect to the exporter when a user is logged on to the workstation.

If the **tryuser** parameter value is Yes, the importer tries connecting to the exporter using the user ID and password of the user logged on at the importer. The connection succeeds if the user has read and attributes permissions to the directories to be replicated. If the connection attempt fails, the importer cannot connect to the exporter until the user logs off. If the **tryuser** parameter value is No, the importer does not connect to the exporter until the user logs off.

Default value: Yes

Tuning Parameters

The following list contains the tuning parameters in the Replicator section of the IBMLAN.INI file:

- **guardtime**

This parameter specifies how long, in minutes, the export path must be stable before importers can connect to it. The **guardtime** parameter prevents importers from connecting to an export path while you are still changing it. For example, if the **guardtime** parameter value is set to 10, importers cannot copy new material from the exporter until at least 10 minutes after you finish changing an export path.

This parameter is valid only if the **integrity** parameter in the REPL.INI file is set to Tree. For more information concerning the REPL.INI file, refer to the *Network Administrator Tasks*.

Default value: 2

Minimum value: 0

Maximum value: **interval** / 2

- **interval**

This parameter specifies how often, in minutes, subdirectories and files in the export path are checked for changes.

Default value: 5

Minimum value: 1
Maximum value: 60

- **pulse**

This parameter specifies how often the exporter sends extra updates to importers, besides those sent when a change to the export path occurs. These *pulse updates* are identical to the most recent change update and ensure that all servers receive an update after the export path changes.

To reduce the number of broadcast frames, increase the value of **pulse**. The risk of changing pulse is that it controls when new users and password changes are replicated from the domain controller to additional servers.

For example, if the **interval** value is 5 and the **pulse** value is 3, a new update is generated every 5 minutes, and, redundant pulse updates are sent every 3 minutes.

Default value: 3
Minimum value: 1
Maximum value: 10

- **random**

This parameter specifies the interval, in seconds, over which to distribute importer connections. The exporter sends this value to importers, which wait an interval of seconds between 0 and this value before connecting.

The **random** parameter controls network traffic of importers connecting to the exporter. If an exporter replicates files to several importers, network performance could be degraded if all importers try connecting to the exporter at once. You can avoid this problem by requiring the importers to wait a random amount of time before connecting.

For example, suppose you specify a **random** parameter value of 90. This value tells each importer to randomly select a number between 0 and 90 and wait that many seconds before connecting to the exporter.

Default value: 60
Minimum value: 1
Maximum value: 120

UPS Section

The UPS section of the IBMLAN.INI file defines the Uninterruptible Power Supply support. The UPS section cannot be the last section listed in the IBMLAN.INI file. The parameters contained in the UPS section can be grouped into two categories: system parameters and tuning parameters. The following lists specify the parameters contained in the UPS section of the IBMLAN.INI file.

- **System Parameters**

- **cmdfile**
- **devicename**
- **lowbattery**
- **signals**
- **voltlevels**

- **Tuning Parameters**

- **batterymsg**
- **batterytime**
- **cmdtimer**

- **messdelay**
- **messtime**
- **recharge**

System Parameters

The following list contains the system parameters in the UPS section of the IBMLAN.INI file.

- **cmdfile**

This parameter specifies the command file to run before the network services are shut down. If the complete path name is not specified, the command file is assumed to be in the \IBMLAN subdirectory.

Default value: "" (no command file should be run)

- **devicename**

This parameter specifies the name of the device to open. UPS_DEV must be a valid OS/2 device name.

Default value: UPS_DEV

- **lowbattery**

This parameter specifies the action taken when the UPS signals that the battery is low, but AC power is present. Possible values are:

Value	Meaning
Warning	The UPS service generates a warning to alert recipients on the interval defined by the batterymsg parameter. The warning informs the user that the UPS battery is low at the specified server.
Shutdown	The server is shut down when the UPS signals that the battery is low.
Disable	The UPS service ignores the signal from the UPS. This value enables the user to discontinue the low battery alerts without stopping the UPS service.

Default value: Warning

- **signals**

This parameter specifies which signals are available from the UPS. The value of this parameter is a 3-digit number, where each digit indicates whether a specific signal is available. A value of 0 for a digit indicates that the signal is not available. A value of 1 for a digit indicates that the UPS can generate the signal.

Value	Signal
1	The UPS can signal when a power failure is detected.
2	The UPS can signal when there are 2 minutes or more until shutdown due to a low battery. If the low-battery signal on the UPS occurs when less than 2 minutes are remaining, this digit should be set to 0.
3	The UPS can accept a signal from the workstation that tells the UPS to shut down.

Note:

The UPS must support a signal either when a power failure is detected or when the battery will run down in 2 or more minutes. If the first 2 digits of the **signals** parameter are set to 0, the UPS service will not function.

Default value: 100

- **voltlevels**

This parameter specifies the voltage levels for the signals defined by the **signals** parameter. The value of the **voltlevels** parameter is a 3-digit number, where each digit indicates the voltage level for a specific signal. A value of 0 for a digit indicates that a negative voltage is used to report the associated signal. A value of 1 for a digit indicates that a positive voltage is used to report the associated signal. For more information about each specific signal, refer to the description of the **signals** parameter.

Default value: 100

Tuning Parameters

The following list contains the tuning parameters in the UPS section of the IBMLAN.INI file:

- **batterymsg**

This parameter specifies the number of seconds that the workstation waits between low battery alerts. The value of this parameter is used only if a low battery signal is received and the AC power is good.

Default value: 600 (10 minutes)

Minimum value: 30 (30 seconds)

Maximum value: 3600 (1 hour)

- **batterytime**

This parameter specifies the number of seconds that the workstation can run on battery power before the UPS service initiates shutdown. The value of this parameter is used only if the low battery signal is not available. Because shutdown should take about 2 minutes, if the battery time is less than or equal to two minutes, the value of this parameter should be set to 0.

Default value: 60

Minimum value: 0

Maximum value: 28800

- **cmdtimer**

This parameter specifies the number of seconds that the UPS service will give the .CMD file to complete. The range is 0 to 600 seconds. The default is 30. If the .CMD file does not finish execution within the specified number of seconds, an error is logged.

Default value: 30

Minimum value: 0

Maximum value: 600

- **messdelay**

This parameter specifies the number of seconds between the initial power failure and the first message sent to the users. If power is restored within the time specified by this parameter, no messages are sent to the users, but the event is logged.

Default value: 5

Minimum value: 0

Maximum value: 120

- **messtime**

This parameter specifies the number of seconds between messages sent to the users. The messages inform the users of the power failure.

Default value: 120

Minimum value: 30

Maximum value: 300

- **recharge**

This parameter specifies the number of minutes of recharge time required by the UPS to gain 1 minute of battery run time.

Default value: 100

Minimum value: 5

Maximum value: 250

Remoteboot Section

The Remoteboot section of the IBMLAN.INI file defines the Remote IPL service. The parameters contained in the Remoteboot section can be grouped into two categories: system parameters and tuning parameters. The following lists specify the parameters contained in the Remoteboot section of the IBMLAN.INI file.

- **System Parameters**

- **rpln**

- **rpldir**

- **configfile**

- **Tuning Parameter**

- **maxthreads**

System Parameters

The following list contains the system parameters in the Remoteboot section of the IBMLAN.INI file:

- **rpln = dll1 dll2 dll3 adapternum**

This parameter specifies the dynamic link libraries (DLLs) used to support various network adapter cards. There is an **rpln** statement for each network adapter card that the Remote IPL service uses. Each **rpln** statement provides the names of the three DLLs to use for the associated network adapter card and the number of the adapter card. The file names specified must be in the user's LIBPATH statement in the CONFIG.SYS file.

The *n* variable is a sequence number used to distinguish each **rpln** statement. The *n* variable can be from 1 to 4.

The *adapternum* variable specifies the number of the associated network adapter card. This variable is optional and defaults to 0 if it is not specified. The network adapter driver associated with the network adapter card specified by this variable must be bound to the IBM IEEE 802.2 protocol driver, or the Remote IPL service will fail.

Default value: RPLNET1.DLL RPLNET2.DLL RPLOEM.DLL

- **rpldir**

This parameter specifies the directory containing the remote IPL files. The directory specified must contain the RPL.MAP file. If a full path name is not specified, the path is assumed to be a subdirectory of the \IBMLAN subdirectory.

Default value: C:\IBMLAN\RPL

- **configfile**

This parameter specifies the name of the DOS boot block configuration file. If a full path name is not specified, the file is assumed to be in the RPLDIR directory.

Default value: DOSBBTR.CNF

Tuning Parameter

The following is the tuning parameter in the Remoteboot section of the IBMLAN.INI file.

- **maxthreads**

This parameter specifies the number of threads that the Remote IPL service starts in order to perform asynchronous reading of the configuration files.

Default value: 10

Minimum value: 0 (no asynchronous I/O)

Maximum value: Maximum number of threads permitted by the system.

DCDB Replicator Section

The DCDB Replicator service propagates *domain control database* (*DCDB*) updates from the domain controller to other servers in the domain that are running with *ROLE=BACKUP*. To change a server's role, see *Quick Beginnings*. The domain control database resides on the domain controller and contains files that describe the domain's users, its resources, and the users' access to those resources. This service runs on the domain controller, and exports the \DCDB directory to the servers. Corresponding DCDB Replicator services run on each server that receives the updated DCDB files. The domain controller is the only machine in the domain that is an exporter, while all other servers running the service are importers. The exportation is thus confined within the domain.

Running the DCDB Replicator service does not conflict with running the Replicator service in the same workstation.

- **System Parameters**

- **logon**
- **password**
- **tryuser**

- **Tuning Parameters**

- **guardtime**
- **interval**
- **pulse**
- **random**

System Parameters

The following list contains the system parameters in the DCDB Replicator section of the IBMLAN.INI file:

- **logon**

This parameter specifies the user ID that the DCDB Replicator service uses to connect to the domain controller when no one is logged on at the server. This user ID is used with the password specified by the **password** parameter. If the

logon parameter (the default) is not included, the DCDB Replicator service tries logging on using the machine ID of the server as a user ID.

Ensure that the user ID (or machine ID if the **logon** parameter is not included) is set up as a user through User Profile Management and has read and attributes permissions to the DCDB through LAN Server (using either the GUI or the NET ACCESS command).

This parameter is not included in your default IBMLAN.INI file. You can add it anywhere in the DCDB Replicator section.

- **password**

This parameter specifies the password that the DCDB Replicator service uses to connect to the domain controller when no one is logged on to the server. This password is used with the user ID specified by the **logon** parameter. The **password** parameter cannot be used unless the **logon** parameter is also specified. If the default setup for the **logon** parameter is used (that is, the **logon** parameter is not included and the DCDB Replicator service tries logging on with the machine ID), then a password cannot be specified. If a password is desired for the workstation, both the **logon** and **password** parameters must be used.

- **tryuser**

This parameter specifies whether the server tries automatically connecting to the domain controller when a user is logged on to the workstation.

If the **tryuser** parameter value is Yes, the server tries connecting to the domain controller using the user ID and password of the user logged on at the server. The connection succeeds if the user has read and attributes permissions to the directories to be replicated. If the connection attempt fails, the server cannot connect to the domain controller until the user logs off. If the **tryuser** parameter value is No, the server does not connect to the domain controller until the user logs off.

Default value: Yes

Tuning Parameters

The following list contains the tuning parameters in the DCDB Replicator section of the IBMLAN.INI file:

- **guardtime**

This parameter specifies how long, in minutes, the export path must be stable before servers can connect to it. The **guardtime** parameter prevents servers from connecting to an export path while you are still changing it. For example, if the **guardtime** parameter value is set to 10, servers cannot copy new material from the domain controller until at least 10 minutes after you finish changing an export path.

This parameter is valid only if the **integrity** parameter in the REPL.INI file is set to Tree. For more information concerning the REPL.INI file, refer to the *Network Administrator Tasks*.

Default value: 2

Minimum value: 0

Maximum value: **interval** / 2

- **interval**

This parameter specifies how often, in minutes, the \DCDB directory in the export path is checked for changes.

Default value: 5

Minimum value: 1

Maximum value: 60

- **pulse**

This parameter specifies how often, in minutes, the domain controller sends extra updates to servers, besides those sent when a change to the export path occurs. These *pulse updates* are identical to the most recent change update and ensure that all servers receive an update after the export path changes.

For example, if the **interval** value is 5 and the **pulse** value is 3, a new update is generated every 5 minutes, and redundant pulse updates are sent every 3 minutes.

Default value: 3

Minimum value: 1

Maximum value: 10

- **random**

Specifies the interval, in seconds, over which to distribute server connections.

The domain controller sends this value to servers, which wait an interval of seconds between 0 and this value before connecting.

The **random** parameter controls network traffic of servers connecting to the domain controller. If a domain controller replicates the DCDB to several servers, network performance could be affected if all servers try connecting to the domain controller at once. You can avoid this problem by requiring the servers to wait a random amount of time before connecting.

For example, suppose you specify a **random** parameter value of 90. This value tells each server to randomly select a number between 0 and 90 and wait that many seconds before connecting to the domain controller.

Default value: 60

Minimum value: 1

Maximum value: 120

Peer Section

The Peer Service allows requesters to share resources. During installation, the Peer Service can be defined to start when the requester starts.

The values for the Peer Service parameters can be changed in the IBMLAN.INI file. An option issued from the command line can override the values in the IBMLAN.INI file.

For more information about the Peer Service, see *Quick Beginnings* and *Network Administrator Tasks*.

System Parameters

Use the following parameters with the Peer Service to override the parameters in the IBMLAN.INI file:

- **autodisconnect**

For a description of this parameter, see the **autodisconnect** parameter 51.

This parameter should always be set to -1 for no automatic disconnect.

Default value: -1

- **security**

This parameter indicates the security mode the peer server is using. Valid values are **user** and **share**.

User level security controls access to resources shared by the peer server through identifying users by a user ID and a password.

Share level security allows the peer server owner to specify both the password and permissions associated with a shared resource.

Default value: User

- **username**

This parameter indicates the user ID of the peer server owner. This parameter is relevant only if the peer server is using **user** level security. The user ID listed in this parameter will be granted a session with the peer server even if another user has a connection. This ensures that the owner can always connect to the peer server. This parameter is ignored if the **security** parameter is set to **share**.

Note: If a resource is shared with /USERS:1, this limit is enforced. That is, the owner cannot connect to the peer server if another user is already connected.

Default value: " "

For additional information about the Peer Service, see the following commands in *Command Reference*:

- NET CONFIG
- NET CONTINUE
- NET PAUSE
- NET SESSION
- NET SHARE
- NET START
- NET STATISTICS
- NET STOP
- NET USE

Browser Section

The Browser Service allows a LAN Server machine in a domain to be visible to Windows clients in the Network Neighborhood folder. The Browser Service consists system and tuning parameters.

System Parameter

- **maintainserverlist**

Specifies whether browser can function as a master browser or a backup browser.

Default : Yes

Tuning Parameter

- **quiet**

Stops error log from filling up.

Services Section

The Services section lists the path to each LAN Server network service. Any path used in this section that does not start with a drive or backslash (\) is assumed to be relative to the IBMLAN directory.

Note:

Do not edit the default values in the Services section.

System Parameters

The following list contains the system parameters in the Services section of the IBMLAN.INI file:

- **alerter**
This parameter specifies the path and file name of the program that sends administrative alert messages. This parameter is valid only for servers.
Default value: SERVICES\ALERTER.EXE
- **DCDBREPL**
Stops the Domain Control Database (DCDB) Replicator service, which copies the DCDB on the domain controller to designated backup servers. This parameter is valid only for servers.
- **genalert**
This parameter specifies the path and file name of the Generic Alerter service.
Default value: SERVICES\GENALERT.EXE
- **lsserver**
This parameter specifies the path and file name of the LSServer service.
Default value: SERVICES\LSSERVER.EXE
- **messenger**
This parameter specifies the path and file name of the message program. The Messenger service allows your workstation to receive the messages sent across the network by other users and servers.
Default value: SERVICES\MSRVINIT.EXE
- **netlogon**
This parameter specifies the path and file name of the Netlogon program. The Netlogon service copies the master user-and-group definition file located on the domain controller to network servers. This parameter is valid only for servers.
Default value: SERVICES\NETLOGON.EXE
- **netrun**
This parameter specifies the path and file name of the NET RUN utility. The Netrun service allows users at other workstations to run programs in this server's memory. This parameter is valid only for servers.
Default value: SERVICES\RUNSERVER.EXE
- **peer**
This parameter specifies the path and file name of the peer server initialization program. The Peer service allows a requester to share resources with one user at a time.
Default value: SERVICES\NETPSINI.EXE
- **remoteboot**
This parameter specifies the path and file name of the DOS and OS/2 Remote IPL programs. The Remote IPL service allows DOS and OS/2 requesters to be started by a server. This parameter is valid only for servers.
Default value: SERVICES\RPLSERVER.EXE
- **replicator**

This parameter specifies the path and file name of the replication program. The Replicator service copies files from a master location to one or more workstations requiring a copy of the data.

Default value: SERVICES\REPLICAT.EXE

- **requester**

This parameter specifies the path and file name of the requester initialization program. The Requester service connects your workstation to the network server as a requester, allowing access to shared network resources.

Default value: SERVICES\WKSTA.EXE

- **server**

This parameter specifies the path and file name of the server initialization program. The Server service allows the workstation to share its resources with users on the network. This parameter is valid only for servers.

Default value: SERVICES\NETSVINI.EXE

- **timesource**

This parameter specifies the path and file name of the Timesource service.

Default value: SERVICES\TIMESRC.EXE

- **ups**

This parameter specifies the path and file name of the Uninterruptible Power Supply initialization program. The UPS service provides protection against loss of data during power failures. Upon power interruption, users with active sessions to the affected server are notified of the impending shutdown, and an orderly server shutdown is performed.

Default value: SERVICES\UPS.EXE

Chapter 4. NETWORK.INI File Parameter Descriptions for DOS LAN Services

This chapter describes the parameters in the NETWORK.INI file, which is on the DOS requesters. This chapter has default, minimum, and maximum values for the NETWORK.INI file parameters.

Changing NETWORK.INI Parameter Values

The NETWORK.INI file can be edited with an ASCII text editor. Before you edit the NETWORK.INI file, it is important to understand its structure.

Activating Changed Parameter Values

If you change the value of a parameter in the NETWORK.INI file, you must shut down and restart the workstation for the change to take effect.

If the network is not started, type NET START without parameters to start DOS LAN Services, using the NETWORK.INI file as the parameter default file. However, if the network has been started, typing NET START displays the current parameter settings.

If the network has been started, first type NET STOP to stop DOS LAN Services. Next, type NET START without parameters to start DOS LAN Services, using the NETWORK.INI file as the parameter default file.

Parameter Descriptions

The parameter defaults have been chosen to enhance performance for most users. Other values should be selected only for configuration needs. The parameters contained in the NETWORK.INI file can be grouped into three categories: system parameters, capacity parameters, and tuning parameters.

Table 6. NETWORK.INI [network] Section Parameter Values

Parameter	Description	Valid Values	Default
computername	Specifies the name that identifies this workstation to the network	Up to 15 alphanumeric characters or special characters, ! # \$ % & () ¢ _ { } ^ ` _	Name specified at installation
lanroot	Specifies the directory where DOS LAN Services is installed and starts	A fully qualified path (drive letter and path)	C:\NET
autologon	Indicates whether to display a window to log on to a domain when DOS LAN Services starts	Yes, No	Yes

Table 6. NETWORK.INI [network] Section Parameter Values (continued)

Parameter	Description	Valid Values	Default
autostart	Indicates which services to start when NET START is entered	NetBEUI, Basic, Full, Predir, Messenger, Netpopup, Peer	Full
username	Specifies the name that identifies you to the network	Up to 20 alpha-numeric characters or ! # \$ % & () ¢ _ { } ^ ` -	Name specified at installation
domain	Specifies the name of the domain this workstation belongs to	Up to 15 alpha-numeric characters or ! # \$ % & () ¢ _ { } ^ ` -	Name specified at installation
reconnect	Indicates whether to reconnect persistent connections at logon	Yes, No	Yes
!slogon	Indicates whether to validate the logon	Yes, No	Yes
numbigbuf ^(1, 2)	Indicates number of big buffers to use (Full and virtual redirectors only)	0 to 4096	2
sizebigbuf ^(1, 2)	Indicates the size of big buffers to use (Full and virtual redirectors only)	4096 to 32768	4096
numworkbuf ⁽¹⁾	Indicates the number of work buffers to use	2 to 16	2
sizworkbuf ⁽¹⁾	Indicates the size of work buffers to use	512 to 16384	1024
extraheap ^(1, 2)	Allocates extra heap space for the redirector. This parameter is needed when using file intensive applications such as databases.	1023 to 32768	0
autocache ⁽¹⁾	Automatically allocates NumBigBuf , SizBigBuf , and ExtraHeap based on the amount of XMS memory. Overrides NumBigBuf , SizBigBuf , and ExtraHeap parameters. This parameter improves performance.	Yes, No	Yes
printbuftime ⁽¹⁾	Specifies the amount of time in seconds before an end-of-print- job notification is sent to the server after the print job is submitted	0 to 65535	0

Table 6. NETWORK.INI [network] Section Parameter Values (continued)

Parameter	Description	Valid Values	Default
lanas ⁽¹⁾	Specifies the number of LAN adapter cards used by your workstation	0 to 7	1
passwordcaching	Indicates whether to cache passwords in a file. This saves passwords to servers in a password-protected file so the user does not have to enter a password for each server that is accessed.	Yes, No	Yes
browsealias ^(1, 3)	Indicates whether netnames or aliases are displayed when browsing network resources in the Windows interface	Yes, No	Yes
timesync	Indicates whether DOS LAN Services will synchronize the time on the local machine to the time on the domain controller. If timesync is set to Yes , the time required to log on is increased.	Yes, No	No
multilogon ⁽⁴⁾	Allows the user to logon multiple times in the same domain.	Yes, No	Yes
securelogon	Prevents local logon, if the user ID cannot be validated on the domain.	Yes, No	No

Note:

1. This parameter must be manually added to the NETWORK.INI file to change the default. The installation program does not add this parameter to the NETWORK.INI file.
2. This parameter is overridden by autocache=yes.
3. Browsing for aliases is valid only on LAN Server domains. If this parameter is set to **Yes**, aliases are browsed. If no aliases exist on the domain, netnames are browsed automatically. If this parameter is set to **No**, browsing for aliases is not attempted.
4. This parameter is valid only when both the user ID and the domain are 8 bytes or less in length.

NETWORK.INI Messenger Parameters

The following table provides information about the parameters in the [messenger] section of the NETWORK.INI file.

Table 7. NETWORK.INI [messenger] Section Parameter Values

Parameter	Description	Valid Values	Default
logfile	The name of the file where received messages are logged	Alphanumeric characters. See your DOS user's guide for valid characters to use when creating file names	MESSAGES.LOG
sizemembuf	Indicates the size of message buffers to use in KB	512 to 4096	512
nummsgnames	Indicates the number of message names to be added to the workstation	2 to 8	2

NETWORK.INI Netpopup Parameter

The following table provides information about the parameter in the [netpopup] section of the NETWORK.INI file.

Table 8. NETWORK.INI [netpopup] Section Parameter Values

Parameter	Description	Valid Values	Default
msgtimeout	Indicates the length of time, in seconds, a message is displayed if Esc is not pressed. If the value is -1, a message is displayed until Esc is pressed.	-1 to 1800	60

NETWORK.INI Peer Parameters

The following table provides information about the parameter in the [peer] section of the NETWORK.INI file.

Table 9. NETWORK.INI [peer] Section Parameter Values

Parameter	Description	Valid Values	Default
a20monitor	Saves the state of the A20 line during task switching. Some memory managers fail when this parameter is set to 1. If you are using a memory manager and experience hangs, set this parameter to 0.	0 to 1	1
numshares	Indicates the maximum number of resources you can share.	2 to 256	10
numreq	Indicates the number of request buffers the Peer service will use when it starts.	3 to 6	3

Table 9. NETWORK.INI [peer] Section Parameter Values (continued)

Parameter	Description	Valid Values	Default
xmitsize	Indicates the size, in bytes, of the transmit buffers the Peer service will use when transferring data. (NumReq * XmitSize < 48K)	512 to 32768	2048
tasktimeslice	Indicates the amount of time (ticks) the foreground and background will each run (TimeSlice=FB). The default for the foreground is 5, and the default for the background is 4 (TimeSlice=54). Value Ticks ----- ----- 0 2 1 4 2 6 3 10 4 14 5 22 6 30 7 42 8 56 9 72	00 to 99	54

Table 9. NETWORK.INI [peer] Section Parameter Values (continued)

Parameter	Description	Valid Values	Default
openmode	<p>Indicates the mode files will be opened in when an open request is received.</p> <p>0 User Open mode requested by the application that is running.</p> <p>1 DENY-NONE sharing mode if read-only access is granted to .EXE or .COM files.</p> <p>COMPATIBILITY mode for .BAT files or if write access is granted to .EXE or .COM files.</p> <p>2 DENY-NONE sharing mode if read-only access is granted to .EXE or .COM files.</p> <p>DENY-WRITE sharing mode if read-only access is granted to .BAT files.</p> <p>COMPATIBILITY mode for .BAT files or if write access is granted to .EXE, .COM, or .BAT files.</p> <p>3 DENY-NONE sharing mode on all compatibility mode opens.</p>	0 to 3	3
filesharesize	<p>Indicates the bytes allocated for the DOS storage area used to record file-sharing information.⁽¹⁾</p>	512 to 32768	2048
sharelocks	<p>Defines the maximum number of active locked ranges in files that you can share with the Peer service.⁽¹⁾</p>	20 to 1000	20

Table 9. NETWORK.INI [peer] Section Parameter Values (continued)

Parameter	Description	Valid Values	Default
fmshare	Indicates whether you want to be able to share directories from the Windows File Manager ⁽²⁾	Yes, No	Yes

Note:

1. If SHARE.EXE is started before the Peer Service, the values that were specified when SHARE.EXE was started are used.
2. This parameter must be manually added to the NETWORK.INI file to change the default. The installation program does not add this parameter to the NETWORK.INI file.

Chapter 5. CONFIG.SYS File Parameter Descriptions

This chapter describes OS/2 functions that can improve overall system performance. Additional information concerning OS/2 memory management and file usage is provided to explain the various memory and hard disk space requirements.

Default values are specified in the CONFIG.SYS file during system installation. Additional parameters are specified in the CONFIG.SYS file during the installation of some application programs. Refer to the application program documentation for additional information concerning the parameter values that can be changed. For more information about parameters in the CONFIG.SYS file, refer to *Command Reference*.

Changing CONFIG.SYS Parameter Values

The CONFIG.SYS file can be edited with an ASCII text editor. The structure of the CONFIG.SYS file should be understood before editing it. The CONFIG.SYS file is updated automatically during the install process.

Activating Changed Parameter Values

If you change the value of a parameter in the CONFIG.SYS file, you must shut down and restart your workstation for the change to take effect.

Memory Management

The CONFIG.SYS file parameters for memory and storage management can be grouped into two categories: system parameters and tuning parameters. The following lists specify the memory- and storage-related parameters contained in the CONFIG.SYS file.

- **System Parameters**
 - **dllbasing**
 - **memman**
 - **protectonly**
- **Tuning Parameters**
 - **buffers**
 - **diskcache**
 - **swappath**

System Parameters

The following list contains the memory- and storage-related system parameters in the CONFIG.SYS file:

- **dllbasing**

The **dllbasing** parameter prevents fragmentation of the shared environment, allowing more efficient use of virtual memory. The **dllbasing** parameter may be set to ON or to OFF. The default value is ON. If **dllbasing** is set to ON, then the system will attempt to honor the base addresses (preferred load addresses) for DLLs. Honoring base addresses for DLLs is preferred because it improves system performance for loading DLLs.

However, sometimes there is an interaction between the DLL basing and an application's memory usage which will cause the system to run out of private memory. In this case, you should set **dllbasing** to OFF so that the system will ignore base addresses for DLLs. The system performance for loading DLLs may degrade marginally, but there will be more free private memory.

- **memman**

The **memman** parameter allows or prevents swapping and segment motion in your system. The first value, Swap or Noswap, specifies whether you want the system to swap memory segments to disk. If you want to run several large programs simultaneously, or if you have a program that is too large for available memory, set the first value to Swap. To prevent swapping, set the first value to Noswap. When you prevent swapping, the OS/2 program requires 5MB of physical memory to restart your system.

The second value, Move or Nomove, specifies whether segment swapping and storage compaction are permitted. Where timing is critical (for example, when process-control tasks are being performed), it is a good idea to prevent all swapping and segment motion in your system. By setting the second value to Nomove, you can prevent storage compaction.

Default value: Swap, Move

- **protectonly**

This required parameter controls the use of memory under 640KB. If the **protectonly** parameter is set to YES, memory under 640KB, which is usually reserved for DOS programs, becomes available for OS/2 programs. When this parameter is set to YES, you cannot run application programs in DOS (real) mode. Set the **protectonly** parameter to NO to run DOS programs in the lower 640KB of memory. If this parameter is set to NO, you can use both DOS and OS/2 programs.

Tuning Parameters

The following list contains the memory- and storage-related tuning parameters in the CONFIG.SYS file:

- **buffers**

Buffering, which deals with the timeliness of I/O devices, is a memory management method. The disk buffer is a 512-byte portion of storage that the OS/2 program uses to hold I/O information temporarily. It assists in helping the I/O device and the processor to work at the same time.

OS/2 uses buffers only for caching FAT and HPFS directory information. This may provide a performance boost when using those file systems. Buffers no longer cache partial sector reads.

By using the disk buffer, the operating system can read and write blocks of information. Once the information is read into the buffer, the processor is ready to process it. While the information is in the buffer or being processed, the input device can begin reading new pieces of information so the processor does not have to wait unnecessarily to process information and program instructions. The processor can then complete its operation because the next block of information to process is already in the buffer area.

If the processor completes its work and there is no information in the buffer to process, the processor must wait until the next block of information is read into memory by the input device. Similarly, if the input device reads information into the buffer before the processor has time to process the records, the buffer might reach its capacity and have to wait until the processor accepts more information to process.

The **buffers** parameter determines the number of disk buffers that the OS/2 program keeps in memory. Depending on how many programs you work with simultaneously, you may want to experiment with this number to maximize performance on your system.

If you simultaneously run a large number of programs, you can increase the speed of your system by increasing the **buffers** parameter value (for example, **buffers=70**). However, remember that when you increase the number of disk buffers, you decrease your available memory by 512 bytes for each buffer specified. Additional buffers may cause some programs to run more slowly because there is less memory available for the program. Also, memory swapping may occur more often, which slows down performance.

Default value: 60

- **diskcache**

Another method of improving system performance is *disk caching*, which allows a portion of the workstation's memory to be used as an additional hard-disk buffer. Disk caching assists the buffer area by holding the most frequently used information, thus enabling the processor to continue processing with fewer interruptions.

For example, suppose the input device is operating slower than the processor. Without disk caching, the processor slowly empties the buffers of information and eventually stops processing. The processor is delayed until the input device supplies additional information. In this situation, the performance of the input device, which should rely on the speed of the processor, is seemingly independent of processor processing.

The disk cache speeds up workstation operations that read from the hard disk by keeping frequently accessed information in a cache buffer. When a program requests hard-disk information that is already in the cache buffer, the disk cache sends the information directly to the program. This method of accessing information is much faster than if the information has to be read from the disk each time.

The proper setting for the caching threshold (128 sectors in the following example) and for the **diskcache** parameter depends on your environment. For example, if you want to cache shared programs, use a threshold of 128. If, however, you want to cache only random I/O, use a threshold of 4. You can determine the best size for the disk cache by experimenting with different values. The disk cache normally needs to be several megabytes in size to achieve the best performance.

An example of a **diskcache** statement is the following:

```
DISKCACHE=xxxx, 128, LW, AC:C
```

where:

xxxx	Is the amount of memory in KB you want to allocate for caching.
128	Specifies the maximum record size for caching, in number of sectors.
LW	Activates <i>lazy writing</i> (writing of cache memory during disk idle time).
AC	Defines the drives to be checked automatically at startup to determine if the file system associated with a drive is in an inconsistent state (for example, if files were left open because of a loss of power). The operating system runs CHKDSK with the /F option to correct any problems caused by improper system shutdown.

For more information about DISKCACHE, refer to *Command Reference*.

- **swappath**

With advanced memory management techniques, the OS/2 program attempts to keep only those portions of a program that are actually active in physical memory. Inactive portions of a program are placed in a special swap file where they remain until they are needed in physical memory.

The **swappath** parameter specifies the size and location of the swap area on the fixed disk. Using a swap file allows the operating system to maintain a greater number of programs in an active state than would be possible to fit into physical memory if a non-virtual memory system were being used.

Pages swapped from running programs go to a temporary file named SWAPPER.DAT. This file grows in size as you start or stop programs and load, modify, or save data files. The SWAPPER.DAT file can sometimes be hundreds of thousands of bytes in size, an indication that your system may need additional memory to improve performance.

You can change the **swappath** parameter value so that the SWAPPER.DAT file resides in another directory or on another logical drive. Placing the SWAPPER.DAT file on a separate logical drive limits the size of the SWAPPER.DAT file and can also improve swapper performance.

The swap file needs to be at least 512KB in size, so you must have at least 512KB of available storage on the swapping drive. The size of the swap file increases as the operating system overcommits storage.

The **swappath** parameter has two options:

minfree

Specifies the minimum amount of free space, in kilobytes, that you want to remain on the disk. When the swap file has used up all other space, the system generates a message.

initsize

Specifies the initial size, in kilobytes, to be allocated for the swap file.

The syntax of the **swappath** statement is:

```
swappath=d:\path minfree initsize
```

Default value: C:\OS2\SYSTEM 2048 2048

Chapter 6. High Performance File Systems

The high performance file system (HPFS) is an installable file system (IFS) that provides better performance than the file allocation table (FAT) file system. The HPFS is designed to provide faster access to very large disk volumes. The base operating system also supports the coexistence of multiple, active file systems on a single workstation.

If you select the HPFS as your standard file system, the base operating system installation procedure inserts an IFS statement as the first entry in the CONFIG.SYS file.

The IFS statement has the following format:

```
IFS=x:\OS2\HPFS.IFS CACHE:nnnn[/AUTOCHECK:drivelist]
```

x: Specifies the drive letter of the partition from which the OS/2 program is started.

/CACHE Specifies the amount of memory, in kilobytes, to use for the HPFS cache. The *nnnn* value must be a multiple of 2, or it is rounded down to a value that is a multiple of 2. If **/CACHE** is omitted, the default value is 10% of the total physical memory available. The minimum value is 64 and the maximum is determined by the amount of physical memory available, up to 2048 (2MB). A cache of at least 512KB is recommended.

/AUTOCHECK

Specifies a list of drive letters to be automatically verified by the CHKDSK command at system startup time if the HPFS detects that the drive may have been shutdown unexpectedly. The CHKDSK command may also be executed for the specified drives if hotfixes have taken place and need to be cleaned up. The *drivelist* value is a string of letters indicating the HPFS drives in the workstation. The drive letters are *not* separated by commas or spaces. The IPL volume is always checked if it is an HPFS volume. If any drive specified is not a valid HPFS drive, the drive is not checked. HPFS drives that are not specified are not checked, except for the IPL volume as previously stated.

If you install the HPFS on your system, your system files are replaced. If you want to restart DOS from diskette, the data files are no longer accessible.

HPFS Caching

Using the OS2.INI file, the system installation program sets up caching for the primary partition. The HPFS manages a cache of memory divided into blocks of 2KB. Data that is read from and written to the disk is transferred through this cache so that it can be used in satisfying future requests. This cache is separate from the **buffers** and **diskcache** parameters.

When a user requests data that is not present in the cache, the HPFS selects the least recently used (LRU) block, writes the data within that block to disk, if necessary, and then fills the block with the requested data. This significantly increases the chances that often-used data will be found in memory, thus saving the expense of a disk-read operation.

In addition, the file system takes advantage of its knowledge of cache block contents. Data not expected to be reused soon by the file system is placed in cache blocks marked for immediate reuse. In most cases, when a write request is received, it is not necessary to write the data immediately to disk. The HPFS takes such data and copies it to the cache block without actually performing the disk-write operation. Once the data is in the cache, it is written to disk as a background activity.

Also, because these *lazy-writes* are performed during disk idle time, incoming requests need not wait for large disk writes to complete. This optimizes overall system throughput and response time for all requesting processes.

The HPFS also contains additional levels of caching that speed up access to directories. The HPFS uses this cache to get quickly to the directory on which the request is being made.

The HPFS cache (CACHE) does not need the cache utility to initiate cache. CACHE.EXE changes the lazy-write parameters. It can be run from the OS/2 command line or from a RUN statement in the CONFIG.SYS file. The CACHE statement has the following syntax:

```
RUN=d:\OS2\CACHE.EXE [/LAZY:value]
                      [/MAXAGE:value]
                      [/DISKIDLE:value]
                      [/BUFFERIDLE:value]
```

where

d: Specifies the drive letter where OS/2 is installed. This should be the boot drive.

/LAZY Enables and disables lazy-writing for the specified drive. If lazy-writing is enabled, the contents of CACHE is written to disk only during disk idle time. If no drive is specified, lazy-writing is either enabled or disabled for all drives serviced by the HPFS. If *value* is Off, lazy-writing is disabled. If *value* is On, lazy-writing is enabled. The default is On. **Attention:** If you set the **/LAZY** option to On, always run Shutdown from the desktop before turning off your workstation. Failure to do so will cause loss of data if the contents of the HPFS cache buffers have not been written to the disk. For safety, and if performance is not a concern, set the **/LAZY** option to Off, which causes a write, through the cache, to the disk.

/MAXAGE Specifies the maximum time, in milliseconds, that a newly-written cache block for a drive can be in memory before it is flushed (written to disk). The *value* can be from 0 through 65535, with a default value of 5000. A value between 1000 and 20000 is recommended. The **/MAXAGE** *value* takes precedence over the **/BUFFERIDLE** *value*. To ensure the effectiveness of lazy-writes, specify a greater value for the **/MAXAGE** option than for **/BUFFERIDLE** option. The **/LAZY** option must be set to On for the **/MAXAGE** option to be effective.

/DISKIDLE Specifies the minimum time, in milliseconds, that a disk must be idle before it can accept data from the CACHE. The *value* can be from 0 through 500000, with a default value of 1000. The *value* must be greater than the **/BUFFERIDLE** *value*. The **/LAZY** option must be set to On for the **/DISKIDLE** option to be effective.

/BUFFERIDLE Specifies the minimum time, in milliseconds, that a dirty cache

block for a drive must be idle before it is written opportunistically when the disk subsystem is idle. The *value* can be from 0 through 500000, with a default value of 500. A value between 100 and 1000 is recommended. The **/MAXAGE** *value* takes precedence over the **/BUFFERIDLE** *value*. To ensure the effectiveness of lazy-writes, specify a greater value for the **/MAXAGE** option than for the **/BUFFERIDLE** option. The **/LAZY** option must be set to On for the **/BUFFERIDLE** option to be effective.

Extended Attribute Support

The file systems can support additional information, called *extended attributes*, about files and directories. Through extended attributes, a program can attach information to a file system object (a file or directory), describing the object to another program, to the operating system, or to the file system driver (FSD) managing that object.

For example, the name of the file's originator can be stored on a file object. Other uses include categorizing file objects, such as icons or bit maps, and describing formats of data in a file object, such as a data record.

A file object may have a list of extended attributes associated with it. These are not part of a file object's data and are managed by the file system that manages objects. An extended attribute must have a name and a value; the name is restricted to the same character set as a file name.

The maximum size of an extended attribute is 64KB and the value is arbitrary data. However, a standard set of conventions for extended attribute use is provided so that data is more easily communicated between programs.

Attention: Extended attributes may be lost if the file is rewritten or copied by a previous version of the OS/2 program, DOS, or an application written before the latest version of the OS/2 program.

386 High Performance File System Support

The 386 HPFS option is an installable file system (IFS) designed for 386 and higher workstations with large disk systems. The 386 HPFS option is loaded by a statement in the CONFIG.SYS file. The statement has the following format:

```
IFS=x:\IBM386FS\HPFS386.IFS  
    [/A:drivelist]
```

where:

- d:** Specifies the drive letter of the partition where the IBMLAN directory for LAN Server resides.
- x:** Specifies the drive letter of the partition where the HPFS386.IFS resides. This should be the boot drive.
- /A** Specifies a list of drive letters to be automatically verified by the CHKDSK command at system startup if the HPFS detects that the drive may have been shut down unexpectedly. The CHKDSK command may also be executed for the specified drives if errors have been hotfixed and need to be cleaned up. The *drivelist* value is a string of letters indicating the HPFS drives in the workstation. The drive letters are *not* separated by commas or spaces. An

asterisk (*) after the **/A** can be used to specify all drives on the system. The IPL volume is always checked, if it is an HPFS volume. If any drive specified is not a valid HPFS drive, the drive is not checked. HPFS drives that are not specified are not checked, except for the IPL volume as previously stated.

HPFS386.INI Parameter Descriptions

This file contains the initialization parameters for the 386 HPFS option. The parameters are grouped into components. The component groups start with the component name enclosed in square brackets. Each component name appears on a line by itself (a comment is allowed). The components include the following:

```
[filesystem]           ; General file system parameters
[lazywriter]          ; Lazy writer parameters
[DASD_Limits]         ; Directory Limits parameters
```

General rules:

- The component names and parameters are not case-sensitive. They can be entered in upper case, lower case, or a mixture of upper and lower case.
- Wherever a blank appears in the syntax for a parameter, it can be left out or additional blanks can be added. For example, the following syntaxes are all valid:

```
cache size = 4096
cache size=4096
cache size= 4096
```

- Any text after a semicolon (;) up to the end of the line is treated as a comment.
- All components and parameters are optional. If you do not specify a parameter, the 386 HPFS option uses a default setting for the parameter.

The [filesystem] section specifies general file system parameters. If you make any changes to these parameters, they do not take effect until you reboot the system.

- **useallmem** = [yes|no]

This parameter specifies whether the 386 HPFS option should use memory above the 16MB boundary, provided this system is configured with more than 16MB. Some adapters, for example the IBM Token Ring Busmaster Server/A, cannot do direct memory access (DMA) to memory above the 16MB boundary. If you have a LAN or disk adapter that cannot do DMA to memory above the 16MB boundary, the 386 HPFS option must use only memory below 16MB so that the adapter can put data into the file system buffers. The **useallmem** parameter will be set to YES if all of your adapters can access memory above the 16MB boundary. You can set the **useallmem** parameter to NO if any of your LAN or disk adapters cannot access memory above the 16MB boundary. The default setting is YES, unless LAN Server Installation/Configuration detects a known problem adapter.

The **useallmem** parameter indicates to the 386 HPFS option that it should use memory above 16MB as much as possible. The **useallmem** parameter has the following effects:

- If the machine has less than 16MB of memory, then setting **useallmem** to NO has no effect on the 386 HPFS option.
- If the machine has more than 16MB of memory, and **useallmem** is set to NO, then the 386 HPFS option does the following:
 - Uses only memory below 16MB for cache blocks.

- Allocates all server buffers that can be used for transmission to the network adapter or disk device drivers below 16MB.
- Any buffers used internally by the file system, as well as the heap, will not be limited to their location, and may be located above 16MB as determined by the OS/2 program.
- Memory conflicts or out-of-memory problems occur more frequently.
- If the machine has more than 16MB of memory, and **useallmem** is set to YES, then the 386 HPFS option does the following:
 - Allocates cache blocks without regard to memory location.
 - Transfer buffers allocated by the server without regard to location.
 - If the 386 HPFS option attempts to mount a volume as directed by the OS/2 program and the disk device driver does not support physical addresses above 16MB, the volume is mounted for direct IOCTL read and write. Normal file service operations are not allowed and an error message, HFS0109, is displayed.
 - On server startup, the server attempts to determine whether all networks to which it belongs are capable of addressing 16MB of memory. If it determines that the networks cannot support this, the server fails to start and logs error message NET3183.
 - If the original IBM 16/4 Token Ring Busmaster adapter is used, the **useallmem** parameter cannot be set to YES.

- **cachesize = nnnn**

This parameter specifies how many kilobytes of memory the 386 HPFS option should claim for its cache. The **cachesize** parameter must be a minimum of 256KB. The maximum value is determined by the size of available memory. If the cachesize is not specified, the 386 HPFS option will choose a value based on the amount of system memory. At an OS/2 command line enter: cache386 /0 to see the cachesize value. If used, the Tuning Assistant calculates a value for cachesize and insert the statement into the HPFS386.INI file.

- **maxheap = nnnn**

This parameter sets a limit on the size of the heap. *nnnn* is the maximum number of kilobytes to which the heap can grow. The 386 HPFS option allocates heap memory as needed. If this parameter is used, the 386 HPFS option allocates memory for the heap up only to the amount specified. If this parameter is not used, there is no limit on the heap size. Use this parameter only if you need to reserve memory on the system for other applications that may be running. The minimum value is 64KB. The maximum value is determined by the size of available memory minus the size of the cache. If **maxheap** is not specified, the default is to have no limit on the heap size.

- **lanroot = d:\path**

This parameter specifies the drive and path of the directory for the LAN Server software. The installation program fills in this parameter for you. You do not need to change this parameter.

- **fsprealloc = nn**

This parameter specifies how many big buffers to allocate when the file system is initialized. If neither the **fsprealloc** nor **srvprealloc** parameters are used, the file system allocates big buffers as needed. The allocation of big buffers can take a little time. Allocating the big buffers at initialization improves the performance of the first requests that need big buffers. The buffers are not freed until the system is shut down. The minimum value for **fsprealloc** parameter is 2. The maximum is

64. If both the **fsprealloc** parameter and **srvprealloc** parameters are specified in this file, the **fsprealloc** parameter is used and the **srvprealloc** parameter is ignored.

- **srvprealloc** = *nn*

This parameter specifies how many big buffers to allocate when the server is started rather than when the file system is initialized. This parameter, like the **fsprealloc** parameter, can improve the performance of the first requests that need big buffers. The buffers are freed when the server is stopped. The minimum value for the **srvprealloc** parameter is 2. The maximum is 64. If both the **fsprealloc** and **srvprealloc** parameters are specified in this file, the **fsprealloc** parameter is used and the **srvprealloc** parameter is ignored.

The [lazywriter] section specifies settings for the lazy writer. If you make any changes to these parameters, they do not take effect until you reboot the system. You can use the CACHE386 program to change the internal setting of these parameters while the system is running. When you reboot the system, the parameters are set to the values in this file.

- **lazy** = [drives:] on|off

This parameter specifies whether the lazy writer is to be turned on or off for the specified drives. The [drives:] can be a series of drive letters. For example, LAZY = CDFG: ON turns on the lazy writer on drives C:, D:, F:, and G:. It would not change the settings for drive E: or H:. An asterisk (*) can be used for the drive letter to indicate that all drives are to have the setting. This line can be used multiple times to achieve the settings you want for your drives. The default value is to turn the lazy writer off for all drives.

- **maxage** = [drives:] *nnnn*

This parameter specifies the maximum number of milliseconds that can pass before the lazy writer writes the contents of a buffer to the disk. The [drives:] can be a series of drive letters. For example, maxage = cdfg: 5000 would set the maximum buffer age to 5000ms on drives C:, D:, F:, and G:. It would not change the settings for drive E: or H:. An asterisk (*) can be used for the drive letter to indicate that all drives are to have the setting. This line can be used multiple times to achieve the settings you want for your drives. The minimum value is 0. The maximum value is 1000000. If **maxage** is not specified, the default value is 10000.

- **bufferidle** = [drives:] *nnnn*

This parameter specifies the maximum number of milliseconds during which a buffer is not used before the lazy writer writes the buffer contents to the disk. The [drives:] can be a series of drive letters. For example, bufferidle = cdfg: 500 would set the buffer idle time to 500ms on drives C:, D:, F:, and G:. It would not change the settings for drive E: or H:. An asterisk (*) can be used for the drive letter to indicate that all drives are to have the setting. This line can be used multiple times to achieve the settings you want for your drives. The minimum value is 0. The maximum value is 500000. If the **bufferidle** parameter is not specified, the default value is 1000 for all drives.

The [DASD_Limits] section specifies settings of parameters for the Directory Limits function. If you make any changes to these parameters they do not take effect until you restart the server. (To stop and restart the server, at an OS/2 command prompt, enter the command NET STOP SERVER and then the command NET START SERVER.)

- **ThreshAlertNames** = [drives:] [user1] [user2] [group1] [group2] ...

This parameter lists the users or groups that are to be notified when a Directory Limits threshold is crossed on the specified drives. Any mixture of user names or

group names can be used. All of the names must appear on one line. This line can be used multiple times to achieve the settings you want for your drives. The installation program initially sets **ThreshAlertNames** to ADMINS, the group comprised of administrators. If the **ThreshAlertNames** parameter is removed, the default is to have no user or group names.

- **ThreshAlertDelay** = [drives:] *nn*

This parameter specifies how many minutes to wait before sending another alert for a threshold that was previously crossed on the specified drives. If a threshold is crossed more than once within the delay period, an alert is sent only for the first occurrence. An alert is sent if a higher threshold is crossed during the delay period. This parameter is used to cut down on the number of alerts that can be generated when there is a lot of disk activity and the disk size is within the threshold. This line can be used multiple times to achieve the settings you want for your drives. If the parameter **ThreshAlertDelay** is not specified, the default is 10 minutes on all drives.

- **ThreshAlertUser** = [drives:] yes|no

This parameter specifies whether to send an alert to the user whose disk usage caused a threshold to be crossed on the specified drives. This line can be used multiple times to achieve the settings you want for your drives. The default is yes for all drives.

- **DirFullAlertNames** = [drives:] [user1] [user2] [user3] ...

This parameter lists the users or groups that are to be notified when a Directory Limit is reached. Any mixture of user names or group names can be used. All of the names must appear on one line. This line can be used multiple times to achieve the settings you want for your drives. The installation program initially sets **DirFullAlertNames** to ADMINS, the group comprised of administrators. If the **DirFullAlertNames** parameter is removed, the default is to have no user or group names.

- **FullAlertDelay** = [drives:] *nn*

This parameter specifies how many minutes to wait before sending another alert for a Directory Limit that was previously reached. If a Directory limit is reached more than once within the delay period, an alert is sent only for the first occurrence. This parameter is used to cut down on the number of alerts that can be generated when there is a lot of disk activity and the Directory Limit is reached several times. This line can be used multiple times to achieve the settings you want for your drives. The default is 10 minutes on all drives.

- **DirFullAlertUser** = [drives:] yes|no

This parameter specifies whether to send an alert to the user whose request failed because a Directory Limit was reached. This line can be used multiple times to achieve the settings you want for your drives. The default is yes for all drives.

Cache386 Utility

The Cache386 utility can be used to display statistics about the 386 HPFS option cache and to change parameters for the lazy writer. Cache386 uses the following syntax:

```
CACHE386 [/STATS: value]  
          [/OPTIONS:d:]  
          [/LAZY:d:value]  
          [/MAXAGE:d:value]  
          [/BUFFERIDLE:d:value]
```

where

- /STATS** Displays cache statistics. The *value* variable can be set to CLEAR or DYNAMIC. If *value* is CLEAR, the current statistics are reset. If *value* is DYNAMIC, the statistics displayed are updated approximately once per second. Press Ctrl+C to stop displaying statistics that are dynamically updated.
- /OPTIONS** Displays current configuration parameters. If no drive is specified, the options for all the drives are displayed.
- /LAZY** Enables and disables lazy-writing for the specified drive.
Attention: If you set the **/LAZY** option to ON, always run shutdown from the desktop before turning off your workstation. Failure to do so will cause loss of data if the contents of the HPFS cache buffers have not been written to the disk. For safety, and if performance is not a concern, set the **/LAZY** option to OFF, which causes a write, through the cache, to the disk. If lazy-writing is enabled, the contents of the cache are only written to disk during disk idle time. If no drive is specified, lazy-writing is either enabled or disabled for all drives serviced by the HPFS. If *value* is OFF, lazy-writing is disabled.
When lazy-writing is OFF, all writes to disk are immediately attempted in a synchronized order. This may cause a degradation in performance because each write must wait for a response from the disk manager before data is written. If *value* is ON, lazy-writing is enabled. The default is ON.
- /MAXAGE** Specifies the maximum time, in milliseconds, that a dirty cache block for a drive can be in memory before it is flushed (written to disk). The *value* can be from 0 through 1000000, with a default value of 5000. A value between 1000 and 20000 is recommended. The **/MAXAGE***value* takes precedence over the **/BUFFERIDLE***value*. To ensure the effectiveness of lazy-writes, specify a greater value for the **/MAXAGE** parameter than for the **/BUFFERIDLE** parameter. The **/LAZY** option must be set to ON for the **/MAXAGE** parameter to be effective.
- /BUFFERIDLE** Specifies the minimum time, in milliseconds, that a dirty cache block for a drive must be idle before it will be written opportunistically when the disk subsystem is idle. The *value* can be from 0 through 500000, with a default value of 500. A value between 100 and 1000 is recommended. The **/MAXAGE***value* takes precedence over the **/BUFFERIDLE** *value*. To ensure the effectiveness of lazy-writes, specify a greater value for the **/MAXAGE** parameter than for the **/BUFFERIDLE** parameter. The **/LAZY** option must be set to ON for the **/BUFFERIDLE** parameter to be effective.

Note:

Refer to the \IBM386FS\HPFS386.INI file for the size of the cache and more information about the 386 HPFS option parameters.

Local Security Support

On servers with the 386 HPFS option with Local Security installed, all processes are subject to file access permissions. While a process is running, different users may log on and log off locally. Each time a user logs off locally, the process loses the permissions granted to that user and has only the permissions given to the group LOCAL. When another user logs on, the active process gains the permissions given to that user, in addition to the permissions granted to the group LOCAL.

Local Security is initialized by the PROTSHELL statement in the CONFIG.SYS file. The PROTSHELL statement has the following format:

```
PROTSHELL=path\SECURESH.EXE [/N] [/F:pathname] shellcmd
```

where:

<i>path</i>	Specifies the path to the SECURESH.EXE file. If the IBMLAN directory is installed on C:\, the default path would be C:\IBMLAN\NETPROG.
/N	Indicates that the system should not be put in secure mode. The /N option allows consistent behavior in the processing of the PRIVINIT.CMD file without forcing the user to use two completely different PROTSHELL statements in the CONFIG.SYS file. If /N is specified, all processes in the system are considered privileged. The default does not specify this option.
/F	Specifies a path and file name other than \PRIVINIT.CMD as the command file containing OS/2 commands to start privileged processes. The default does not specify this option.
<i>shellcmd</i>	Specifies the user shell program and the associated arguments for the shell program. An example value for <i>shellcmd</i> is: C:\OS2\PMSHELL.EXE.

Omitting SECURESH.EXE from the PROTSHELL statement in the CONFIG.SYS file and just starting the normal user shell does not secure the file system. Instead, the workstation runs in unsecured mode. Therefore, the CONFIG.SYS file must be a protected file to prevent unprivileged users from editing it. If SECURESH.EXE is not included on the PROTSHELL statement, the \PRIVINIT.CMD file will not be executed.

Appendix A. File System Comparison Chart

The chart on the following pages compares the characteristics of four file systems: JFS, 386 HPFS, HPFS, and FAT.

Note that when 386 HPFS is installed, it *replaces* HPFS.

Table 10. File System Comparison: JFS, 386 HPFS, HPFS, and FAT

Characteristic	Journalled File System (JFS)	386 High Performance File System (386 HPFS)	High Performance File System (HPFS)	FAT File System
Maximum volume size	2TB (terabytes)	64GB (gigabytes)	64GB (gigabytes)	2GB (gigabytes)
Maximum file size	2TB (terabytes)	2GB (gigabytes)	2GB (gigabytes)	2GB (gigabytes)
Maximum file name length	254 characters (bytes)	254 characters (bytes)	254 characters (bytes)	11 (8.3 format) characters
Allows spaces and periods in file names	Yes	Yes	Yes	No; however, a period is required between file name and extension
Standard file and directory attributes (bit flags)	Within file system	Within file system	Within file system	Within file system
Extended attributes (64KB text or binary with keywords)	Within file system	Within file system	Within file system	In a separate file
Maximum path length	260 characters (bytes)	260 characters (bytes)	260 characters (bytes)	64 characters (bytes)
Bootable file system	No	Yes	Yes	Yes
Logical volume type	LVM only	LVM or compatibility (must be compatibility for Fault Tolerance)	LVM or compatibility	LVM or compatibility
Allows dynamic volume expansion	Yes	No	No	No
File system scales up when using SMP	Yes	No	No	No
Allows for fault tolerance	No	Yes	No	No
Allows for local security	No	Yes	No	No
Requires a separate proof of license during installation	No	Yes	No	No
Average wasted space per file	1/2 block size (256 to 2048 bytes)	1/2 sector (256 bytes)	1/2 sector (256 bytes)	1/2 cluster (1KB to 16KB), approximately 0.001% of volume
Allocation information for files	Near each file in its INODE	Near each file in its FNODE	Near each file in its FNODE	Centralized near front of volume
Free disk-space information	Centralized in file structure represented by B+tree	Located near free space in bit maps	Located near free space in bit maps	Centralized near front of volume

Table 10. File System Comparison: JFS, 386 HPFS, HPFS, and FAT (continued)

Characteristic	Journalled File System (JFS)	386 High Performance File System (386 HPFS)	High Performance File System (HPFS)	FAT File System
Directory structure	Sorted B+tree	Sorted B-tree	Sorted B-tree	Unsorted linear list; must be searched exhaustively
Directory location	Close to files it contains	Located near seek center of volume	Located near seek center of volume	Root directory at front of volume; others scattered
Read-ahead	32MB	Sensitive to data type	Cache reads in 8KB blocks	Optional
Cache replacement strategy	Segmented least recently used (LRU) algorithm	Sensitive to data type	Modified least recently used (LRU) algorithm	Simple least recently used (LRU) algorithm
Write-behind (lazy write)	Optional; can be handled on a per-file basis	Optional; can be handled on a per-file basis	Optional; can be handled on a per-file basis	Optional; can be handled on a per-file basis
Caching program	None; lazy write parameters can be set on the IFS=JFS.IFS line in the CONFIG.SYS file)	CACHE386.EXE (values specified for the IFS and RUN statements in the CONFIG.SYS file) and HPFS386.INI	CACHE.EXE (in the CONFIG.SYS file)	DISKCACHE (in the CONFIG.SYS file)
Maximum cache size	Physical memory available	Physical memory available	2MB	14MB
Cache threshold	No cache threshold	Sensitive to data type	Variable, up to 64KB	Variable, up to 64KB; 3.5KB default
LAN Server access control lists	Within file system	Within file system	In separate file (NET.ACC)	In separate file (NET.ACC)

Appendix B. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs

and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department LZKS
11400 Burnet Road
Austin, TX 78758
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

IBM
OS/2

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

Other company, product, and service names may be trademarks or service marks of others.

Index

Numerics

- 386 High Performance File System
 - comparison with other file systems 111
- 386 HPFS
 - comparison with other file systems 111
- 386 HPFS cache values
 - /BUFFERIDLE 108
 - /LAZY 102, 108
 - /MAXAGE 108
 - /OPTIONS 108
 - /STATS 108
- 386 HPFS Option 5
- 386 SMB server 5

A

- A20MONITER parameter 92
- access control
 - information 6
 - invalid access attempt 50
- accessalert parameter 50
- account
 - guest 52
 - NET ACCOUNTS command 70
- activating change parameter
 - CONFIG.SYS file 97
 - IBMLAN.INI file 29
 - NETWORK.INI file 89
- adapter
 - file transfers 42
 - LAN 42
 - network 31
- adapter driver 31
- alert
 - accessalert parameter 50
 - administrative 50, 73
 - alerter parameter 86
 - Alerter service 73
 - alernames parameter 50
 - alertsched parameter 56
 - buffer size 73
 - diskalert parameter 52
 - erroralert parameter 52
 - logonalert parameter 53
 - netioalert parameter 53
 - network I/O 53
 - numalerts parameter 35
 - program tasks 35
 - rate messages sent 56
 - resource access 50
 - sizalertbuf parameter 73
- alerter parameter 86
- Alerter service
 - capacity parameter 73
 - IBMLAN.INI parameters 73
 - path and file name in IBMLAN.INI 86
 - sizalertbuf parameter 73
- alernames parameter 50

- alertsched parameter 56
- alias
 - cleanup shares for dynamic 72
 - NetBIOS names 32
 - sharing startup 72
- AT command 52
- attribute, extended 103
- audit trail
 - auditing parameter 50
 - recording session use 50
 - selecting and deselecting auditing 50
 - size 53
 - writing to a file 60
- auditing parameter 50
- AUTOCACHE parameter 89
- autocheck 101
- autodisconnect parameter 51, 84
- AUTOLOGON parameter 89
- autopath parameter 51
- AUTOSTART parameter 89

B

- backing up
 - applications 74
 - files 74
 - server 71
 - user data files 74
- batterymsg parameter 80
- batterytime parameter 80
- BROWSEALIAS parameter 91
- Browser Section 85
- buffer
 - alerts 73
 - big (64KB) 57, 62, 66
 - big buffer header information 2
 - buffering modes for read and write requests 41
 - CONFIG.SYS file 98
 - data 2
 - datagrams 36
 - errors 36
 - for large file transfers 57, 64, 66
 - for messages 73
 - for opened files 44, 61
 - for small read and write requests 41
 - freeing big buffers (64KB) 64
 - header information 2
 - internal network 1
 - memory management 98
 - named pipe 40
 - number of (requester) 38
 - number of (server) 57, 62, 66
 - parameters 15
 - processing large transfers and read-aheads 62
 - read-ahead 1, 5, 42, 59
 - requester 37
 - serial devices 40
 - server 57, 66

buffer (*continued*)
 server network 73, 6
 size 38, 58
 write-behind 5, 42, 59
buffers parameter 98

C

cache 2
caching 6
charcount parameter 36
chartime parameter 37
charwait parameter 37
CHKDSK command 101, 103
cleanup parameter 72
cmdfile parameter 79
cmdtimer parameter 80
command
 AT 52
 CHKDSK 101, 103
 NET ACCOUNTS 70
 NET CONFIG 29
 NET ERROR 10, 13
 NET START 29
 NET STATISTICS 13
 NET VIEW 33
 NetBIOS 32, 34
 PATH 74
common interface file system 2
Communications Manager
 NetBIOS names 32
 NetBIOS resources 31
computer name 32
computername parameter 33
COMPUTERNAME parameter 89
CONFIG.SYS file
 definition 1
 PROTSHELL statement 109
 system parameters 97
 tuning parameters 98
 when changes take place 97
CONFIG.SYS parameters
 buffers 98
 diskcache 99
 dllbasing 97
 memman 98
 protectonly 98
 swappath 100
configfile parameter 82
connection
 disconnecting inactive sessions 51
 inactive 37
 to export path 77, 83
copy DCDB 82
copying file 74
core server 42

D

data transfer 2
datagram
 number of buffers 36

datagram (*continued*)
 reducing traffic 36
DCDB Replicator Service 82
DCDB Replicator service 82
 checking for changes 83
 IBMLAN.INI parameters 82
 logon 82
 pulse updates 84
 REPL.INI file 83
 system parameters 82
 tuning parameters 83
deny-none sharing mode 58, 61, 67
deny-write sharing mode 44, 61, 68
device driver 31
devicename\$ variable 31
devicename parameter 79
directory
 backing up 74
 copying 74
 export 74
 extended attributes 103
 import 74
 LOGS subdirectory 73
 maximum searches per server 65
 replicating 74
 searches 60
disconnecting
 inactive sessions 51
 sessions 38
disk
 caching 99
 space 52
diskalert parameter 52
diskcache parameter 99
diskcaching, definition 99
dllbasing parameter 97
domain
 additional domains 34
 default name 33
 parameter 33
domain control database (DCDB), definition 82
domain controller
 backup 71
 Netlogon service 70
 primary 70
 user and group definitions file 70
domain parameter 33
DOMAIN parameter 89
DOS LAN Requester
 performance components 9
 Sideband 6
driver
 loop-back 31
 network adapter 31
 protocol 30
 TCP/IP 31

E

editing
 CONFIG.SYS file 97

- editing (*continued*)
 - IBMLAN.INI file 97
 - NETWORK.INI file 89
- encryption 43
- error
 - buffer size 36
 - events causing error alerts 52
 - maximum 52
 - network I/O 53
 - running out of predefined resources 52
- error log
 - logging multiple error occurrences 43
 - size 35
 - writing to a file 60
- erroralert parameter 52
- export path 74
- exporter 74
- exportlist parameter 75
- exportpath parameter 76
- extended attribute support 103
- external source
 - guest account 52
- EXTRAHEAP parameter 89

F

- FAT
 - cache 2
 - comparison with HPFS 2
 - definition 2
 - improved performance 2
 - resource requests from 386 HPFS Option 6
- FAT file system
 - comparison with other file systems 111
- FCB 60, 67
- file
 - 386 HPFS I/O 6
 - access with Advanced server 6
 - backing up 74
 - buffering reads and writes 66
 - CONFIG.SYS 97
 - copy 74
 - FAT I/O 6
 - IBMLAN.INI 29
 - large transfer 3
 - locking 63, 68
 - NET.ACC 70
 - network I/O requests 1
 - NETWORK.INI 89
 - open 61, 67
 - random access 3
 - REPL.INI 77, 83
 - replicating 74
 - scatter/gather logic 6
 - sequential access 3
 - transfers 57, 62, 66
 - user and group definitions 70
- file allocation table 1
- File Allocation Table file system
 - comparison with other file systems 111
- file system
 - comparison chart 111

- files resource
 - backing up 74
 - buffering 44, 61
 - copying 74
 - driver 5, 103
 - extended attributes 103
 - HPFS 101
 - locking 39, 54, 65
 - maximum open 54, 65
 - open 39, 61, 67
 - read-ahead 59
 - replicating 74
 - SMB processor 5
 - transfers 62
 - write-behind 59
- FILESHARESIZE parameter 92
- FMSHARE parameter 92
- FSD (file system driver) 103

G

- GDT selectors, NetBIOS 2
- genalert parameter 86
- group
 - IDs 50
 - NET.ACC file 70
 - user and group definitions file 70
- guardtime parameter 77, 83
- guestacct parameter 52

H

- heap 18
 - considerations 18
 - definition 18
- heuristics
 - adjusting network buffer activity 20, 24
 - Requester service 38
 - Server service 58, 66
- High Performance File System
 - comparison with other file systems 111
- HPFS
 - comparison with other file systems 111
- HPFS cache
 - cache 2
 - cache size 101
 - caching 101
 - comparison with FAT 2
 - definition 101
 - disk idle time 102
 - improved performance 2
 - installable file system (IFS) 101
 - maximum block age 102
 - minimum block age 102
 - Sideband 6

I

- IBMLAN.INI capacity parameters
 - Alerter section 73
 - maxauditlog 53

IBMLAN.INI capacity parameters *(continued)*

- Messenger section 73
- Netrun section 74
- Networks section 32
- Requester section 34
- Server section 53
- Server section 386 HPFS 65

IBMLAN.INI file

- Alerter section 73
- Alerter service 86
- comments 29
- DCDB Replicator section 82
- definition 1
- description 29
- domain 33
- editing 29
- Generic Alerter service 86
- heuristics (requester) 38
- heuristics (server) 58, 66
- LSserver section 72
- LSserver service 86
- maximum NetBIOS sessions 10
- Messenger section 73
- Messenger service 86
- Netlogon section 70
- Netlogon service 86
- Netrun section 74
- Netrun service 86
- Networks section 30, 31
- Peer section 84
- performance parameters 14
- Remote IPL service 86
- remoteboot 86
- Remoteboot section 81
- Replicator section 74
- Replicator service 86
- Requester section 32
- Requester service 87
- restrictions on editing 29, 85
- sections 30
- separating values 29
- Server parameters 48
- Server section 48
- Server section (Peer) 48
- Server section 386 HPFS 65
- Server service 87
- Services section 85
- startup parameters 29
- Timesource service 87
- UPS section 78
- UPS service 87
- when changes take effect 29

IBMLAN.INI file capacity parameters

- Networks section 10
- Server section 10

IBMLAN.INI heuristics

- srvheuristics parameter 58
- srvheuristics parameter 386 HPFS 66
- wrkheuristics parameter 38

IBMLAN.INI parameters

- accessalert 50

IBMLAN.INI parameters *(continued)*

- alerter 50, 86
- alernames 50
- alertsched 56
- auditing 50
- autodisconnect 51, 84
- autopath 51
- batterymsg 80
- batterytime 80
- charcount 36
- chartime 37
- charwait 37
- cleanup 72
- cmdfile 79
- cmdtimer 80
- computername 33
- configfile 82
- devicename 79
- diskalert 52
- domain 33
- erroralert 52
- exportlist 75
- exportpath 76
- genalert 86
- guardtime 77, 83
- guestacct 52
- importlist 76
- importpath 76
- interval 77, 83
- keepconn 37
- keepsearch 37
- logfile 73
- logon 76, 82
- logonalert 53
- lsserver 86
- LSserver section 72
- LSserver section(Remoteboot) 72
- maintainserverlist 85
- maxchdevjob 53
- maxchdevq 54
- maxchdevs 54
- maxcmds 29, 34
- maxerrorlog 35
- maxlocks 54, 65
- maxopens 54, 65
- maxruns 74
- maxsearches 55, 65
- maxsessopens 55
- maxsessreqs 55
- maxsessvcs 53, 56
- maxthreads (Remoteboot) 82
- maxthreads (Requester) 29, 35
- maxusers 56
- maxwrkcache 37
- messdelay 80
- messenger 86
- Messenger section 73
- netioalert 53
- netlogon 86
- Netlogon section 70
- netrun 86

IBMLAN.INI parameters *(continued)*

- Netrun section 50
- netx 29, 31
- numalerts 35
- numbigbuf 57, 66
- numcharbuf 35
- numdgrambuf 36
- numfiletasks 57
- numreqbuf 57
- numservices 36
- numworkbuf 38
- othdomains 34
- othsrvnames 53
- password 77, 83
- peer 86
- Peer section 84
- printbuftime 38
- pulse (Netlogon) 70
- pulse (Replicator) 78, 84
- quiet 85
- random 78, 84
- randomize 71
- recharge 81
- remoteboot 86
- Remoteboot section 81
- replicate 75
- replicator 86
- requester 87
- rpldir 81
- rpln 81
- runpath 74
- scanpause 71
- scanperiod 71
- scantime 71
- scripts 70
- security 84
- server 87
- Server section 48
- Server section 386 HPFS 65
- Services section 85
- sesstimeout 38
- signals 79
- sizalertbuf 73
- sizcharbuf 36
- sizerror 36
- sizmessbuf 73
- sizreqbuf 58
- sizworkbuf 38
- svanndelta 58
- svannounce 58
- srvheuristics 58, 66
- srvhidden 58
- srvnets 53
- srvpipes 72
- svservices 53
- timesource 87
- tryuser 77, 83
- update 70
- ups 87
- UPS section 78
- useallmem 34

IBMLAN.INI parameters *(continued)*

- username 50
- voltlevels 80
- wrkheuristics 38
- wrknets 34
- wrkservices 34
- IBMLAN.INI system parameter
 - Messenger section 73
- IBMLAN.INI system parameters
 - DCDB Replicator section 82
 - lowbattery 79
 - LSserver section 72
 - Netlogon section 70
 - Peer section 84
 - Remoteboot section 81
 - Replicator section 75
 - Requester section 33
 - Server section 50
 - UPS section 79
- IBMLAN.INI tuning parameters
 - DCDB Replicator section 83
 - Netlogon section 70
 - Netrun section 74
 - Remoteboot section 82
 - Replicator section 77
 - Requester section 36
 - Server section 56
 - Server section 386 HPFS 66
 - UPS section 80

ID

- group 50
- machine 77, 83
- user 77, 83

IFS

- 386 HPFS server definition 103
- autocheck 101
- cache size 101
- description 101
- format 101
- statement in CONFIG.SYS 101
- import path 74
- importer 74
- importlist parameter 76
- importpath parameter 76
- integrity parameter 77, 83
- interval parameter 77, 78, 83, 84
- IOCTL (input/output control) 63

J

- JFS
 - comparison with other file systems 111
- Journal File System
 - comparison with other file systems 111
- Journal File System (JFS) 1

K

- keepconn parameter 37
- keepsearch parameter 37

L

- LAN
 - average size 10

- LAN (*continued*)
 - with many concurrent users 10
- LAN Server 1
 - design changes 1
 - improved functionality 1
- LANAS parameter 91
- LANROOT parameter 89
- LM10 parameter 32
- Local Security
 - definition 6
 - support 109
- log
 - audit log 53
 - message log file 73
 - writing to a file 60
- logfile parameter 73
- LOGFILE parameter 92
- Logical Volume Manager (LVM) 1
- logon
 - alerts 53
 - DCDB Replicator service 82, 83
 - Netlogon service 70
 - recording session use 50
 - Replicator service 76, 77
 - violations 53
- logon parameter 76, 82
- logonalert parameter 53
- logonverification 47
- logonwarningmsgs 47
- loop-back driver 31
- lowbattery parameter 79
- LSLOGON parameter 89
- Isserver parameter 86
- LSserver section
 - system parameters 72
 - tuning parameter 72
- LSserver section parameters
 - cleanup 72
 - srvpipes 72

M

- machine ID 77, 83
- maintainserverlist 85
- maxauditlog parameter 53
- maxchdevjob parameter 53
- maxchdevq parameter 54
- maxchdevs parameter 54
- maxcmds parameter 29, 34
- maxerrorlog parameter 35
- maxlocks parameter 54, 65
- maxopens parameter 54, 65
- maxruns parameter 74
- maxsearches parameter 55, 65
- maxsessopens parameter 55
- maxsessreqs parameter 55
- maxsessvcs parameter 53
- maxshares parameter 56
- maxthreads parameter
 - Remoteboot section 82
 - Requester section 29, 35

- maxusers parameter 56
- maxwrkcache parameter 37
- member server 71
- memman parameter 98
- memory
 - alerts 73
 - requester parameters 32
 - tuning considerations 15
- memory buffer
 - requester 37
- memory buffers
 - administrative alerts 73
 - server 57, 58, 66
- memory management 97
- message
 - alerts 50, 52, 53
 - controlling logging of errors 43
 - from additional domains 34
 - log files 73
- messaging name 32
- messdelay parameter 80
- messenger parameter 86
- Messenger service
 - capacity parameter 73
 - IBMLAN.INI parameters 73
 - path and file name in IBMLAN.INI 86
 - system parameter 73
- Messenger service parameters
 - logfile 73
 - sizmessbuf 73
- minfree variable 100
- minimizing memory usage 9
- MSGTIMEOUT parameter 92
- MTU size 16
- MULTILOGON 91
- multiple logons 48

N

- name used on the LAN
 - alias 32
 - computer name 32, 33
 - domain name 33
 - messaging name 32
- named pipe 40
- NET.ACC file 70
- NET ACCOUNTS command 70
- NET CONFIG command 29
- NET ERROR command 10, 13
- NET START command 29
- NET STATISTICS command 13
- NET VIEW command 33
- net1 parameter 29
- NETBEUI_NIF section
 - maximum NetBIOS sessions 10
 - maximum sessions 12
- NETBIOS
 - chain-send command 60
 - commands 32, 34
 - device driver 2, 6
 - maximum sessions 10, 12
 - names 32

NETBIOS (*continued*)
 resources 60
 sessions 32, 39
 NetBIOS over TCP/IP 6
 netioalert parameter 53
 netlogon parameter 86
 Netlogon parameters
 pulse 70
 randomize 71
 scanpause 71
 scanperiod 71
 scantime 71
 update 70
 netlogon password resynchronizing 70
 Netlogon service
 IBMLAN.INI parameters 70
 NET.ACC file 70
 netlogon service password resynchronization 70
 path and file name in IBMLAN.INI 86
 system parameters 70
 tuning parameters 70
 Netlogon service parameters
 scripts 70
 netrunc parameter 86
 Netrun service parameters
 runpath 74
 Netrun service
 capacity parameter 74
 network buffer 1
 path and file name in IBMLAN.INI 86
 tuning parameters 74
 Netrun service NetworkService
 for servers only 30
 number of 36
 numservices parameter 36
 permanent changes 29
 temporary changes 29
 to start with Requester service 34
 to start with Server service 53
 Netrun service parameters
 maxruns 74
 network adapter driver 31
 network file I/O 1, 5
 NETWORK.INI file 89
 NETWORK.INI file parameters
 A20MONITER 92
 AUTOCACHE 89
 AUTOLOGON 89
 AUTOSTART 89
 BROWSEALIAS 91
 COMPUTERNAME 89
 DOMAIN 89
 EXTRAHEAP 89
 FILESHARESIZE 92
 FMSHARE 92
 LANAS 91
 LANROOT 89
 LOGFILE 92
 LSLOGON 89
 MSGTIMEOUT 92
 MULTILOGON 91
 NETWORK.INI file parameters (*continued*)
 NUMBIGBUF 92
 NUMMSGNAMES 92
 NUMREQ 92
 NUMSHARES 92
 NUMWORKBUF 89
 OPENMODE 92
 PASSWORDCACHING 91
 PRINTBUFTIME 89
 RECONNECT 89
 SECURELOGON 91
 SHARELOCKS 92
 SIZEBIGBUF 89
 SIZEMEMBUF 92
 SIZWORKBUF 89
 TASKTIMESLICE 92
 TIMESYNC 91
 USERNAME 89
 when changes take effect 89
 XMITSIZE 92
 network parameters 32
 network tuning 1
 Networks section
 capacity variables 32
 netx parameter 31
 system variables 31
 netx parameter 31
 numalerts parameter 35
 NUMBIGBUF parameter 89
 numbigbuf parameter
 Server section 57
 Server section 386 HPFS 66
 tuning considerations 15
 wrkheuristics parameter 45
 numcharbuf parameter 35
 numdgrambuf parameter 36
 numfiletask parameter
 Server section 57
 tuning considerations 15
 NUMMSGNAMES parameter 92
 NUMREQ parameter 92
 numreqbuf parameter
 maxruns parameter 74
 Server section 57
 tuning considerations 15
 numservices parameter 36
 NUMSHARES parameter 92
 numworkbuf parameter 38
 NUMWORKBUF parameter 89

O
 OPENMODE parameter 92
 opportunistic lock time-out 63, 68
 opportunistic locking
 definition 24
 requesting 39
 specifying 59, 66
 OS/2 service requirements 1, 5
 othdomains parameter 34
 othsrnames parameter 53

P

- password
 - DCDB Replicator service 83
 - Replicator service 77
- password parameter 77, 83
- PASSWORDCACHING parameter 91
- PATH command 74
- peer parameter 86
- Peer section 84
- Peer Service 84
- Peer service 84
- Pentium optimization 6
- performance
 - definition 13
 - disk caching 99
 - heuristics 38, 58, 66
 - IBMLAN.INI fine-tuning options 38, 58, 66
 - IBMLAN.INI parameters 14
 - improving 13
 - network tuning 1
 - number of users 48
 - server 57
 - tuning 9, 29
- pipe
 - maximum per server 54, 65, 72
 - named 40
- primary server 70
- printbuftime parameter 38
- PRINTBUFTIME parameter 89
- processing SMBs 1
- protectonly parameter 98
- protocol 42
- PROTOCOL.INI capacity parameters 12
- PROTOCOL.INI file
 - maximum sessions 12
 - netbiosretries 25
- pulse parameter
 - Netlogon section 70
 - Replicator section 78, 84
- pulse update 70, 78, 84

Q

- quiet 85

R

- random parameter 78, 84
- randomize parameter 71
- read-ahead
 - definition 59
 - in 386 HPFS Option 5
 - in LAN Server 1
 - in srvheuristics parameter 59
 - in wrkheuristics parameter 42
- recharge parameter 81
- RECONNECT parameter 89
- redirector 2
- Remote IPL
 - path and file name in IBMLAN.INI 86
 - service 81

- remote program execution 74
- remoteboot parameter 86
- Remoteboot section
 - system parameters 81
 - tuning parameters 82
- Remoteboot section parameters
 - configfile 82
 - maxthreads 82
 - rpldir 81
 - rpln 81
- REPL.INI file 77, 83
- replicate parameter 75
- replicator parameter 86
- Replicator section
 - tuning parameters 77
- Replicator service
 - backing up 74
 - checking for changes 77
 - DCDBREPL 85
 - exporter 74
 - IBMLAN.INI parameters 74
 - importer 74
 - logon 76, 77
 - path and file name in IBMLAN.INI 86
 - pulse update 78
 - REPL.INI file 77
 - separating IBMLAN.INI values 29
 - system parameters 75
- Replicator service parameters
 - exportlist 75, 76
 - guardtime 77, 83
 - importlist 76
 - importpath 76
 - interval 77, 83
 - logon 76, 82
 - password 77, 83
 - pulse 78, 84
 - random 78, 84
 - replicate 75
 - tryuser 77, 83
- requester
 - additional domains 34
 - buffers 37
 - computer name 33
 - data transfer 2
 - errors 45
 - names of networks running on 34
 - performance components 3, 7, 9
 - redirector 2
 - tuning options 38
- requester parameter 87
- Requester section
 - capacity parameters 34
 - heuristics 38
 - IBMLAN.INI parameters 32
 - other services started with 34
 - path and file name in IBMLAN.INI 87
 - system parameters 33
 - tuning parameters 36
- Requester section parameters
 - charcount 36

Requester section parameters (*continued*)

- chartime 36
- charwait 37
- computername 33
- domain 33
- keepconn 37
- keepsearch 37
- maxcmds 34
- maxerrorlog 35
- maxthreads 35
- maxwrkcache 37
- numalerts 35
- numcharbuf 35
- numdgrambuf 36
- numservices 36
- numworkbuf 38
- othdomains 34
- printbufftime 38
- sesstimeout 38
- sizcharbuf 36
- sizerror 36
- sizworkbuf 38
- useallmem 34
- wrkheuristics 38
- wrknets 34
- wrkservices 34

resource files

- invalid access attempt 50
- maximum shared per server 56

rpldir parameter 81

rpln parameter 81

run program remotely 74

runpath parameter 74

S

scanpause parameter 71

scanperiod parameter 71

scantime parameter 71

scatter/gather logic 6

scavenger, definition 60

scripts parameter 70

SECURELOGON 91

security parameter 84

serial device

- IBMLAN.INI parameters 36
- maximum requests per server 53
- maximum shared 54
- maximum shared queues 54

server

- announce 58
- announce rate 58
- backup 71
- buffers for requests 57, 58
- concurrent file and print requests 57
- data transfer 2
- datagram 36
- heuristics 58, 66
- member 71
- network buffer 2, 6
- networks running on 53
- opportunistic locking 59

server (*continued*)

- parameters 58
- performance components 3, 7, 9
- primary 70
- priority 60
- remote program execution 74
- roles 70
- timeout 38
- tuning options 58, 66
- varying announce rate 58

server maximums

- directory searches 65
- file locks 54, 65
- maximum searches per server 55
- open files 54
- open sessions 55
- requests for serial device queues 53
- resource requests 55
- shared resources 56
- shared serial device queues 54
- shared serial devices 54
- users 56
- virtual circuits 53

server message block 1

Server parameter 87

Server section

- capacity parameters 53
- heuristics 58
- IBMLAN.INI parameters 48
- other server running on 53
- system parameters 50
- tuning parameters 56

Server section 386 HPFS

- maxlocks 65
- maxopens 65
- maxsearches 65

Server section 386 HPFS parameters

- numbigbuf 66
- peer 86
- performance 14
- srvheuristics 66

Server section HPFS

- capacity parameters 65
- heuristics 66
- tuning parameters 66

Server section parameters

- alertsched 56
- maxauditlog 53
- maxchdevjob 53
- maxchdevq 54

server section parameters

- maxchdevs 54
- maxlocks 54
- maxopens 54

Server section parameters

- maxsearches 55
- maxsessopens 55
- maxsessreqs 55
- maxsessvcs 56
- maxusers 56
- netioalert 53

Server section parameters (*continued*)

- numbigbuf 55
- numfiletasks 57
- numreqbuf 57
- othsrvnames 53
- performance 14
- sizreqbuf 58
- svanndelta 58
- svannounce 58
- srvheuristics 58
- srvhidden 58
- srvnets 53
- srvservices 53

service

- Alerter 86
- Generic Alerter 86
- location 85
- LSserver 86
- Messenger 86
- Netlogon 86
- Netrun 86
- path names 85
- Peer 84
- Replicator 86
- Requester 87
- Server 87
- Timesource 87
- UPS 87

Services section

- default values 29
- IBMLAN.INI parameters 85
- path and file name in IBMLAN.INI 87
- restrictions on editing 85

Services section parameters

- accessalert 50
- alerter 86
- alertnames 50
- auditing 50
- autodisconnect 51
- autopath 51, 52
- erroralert 52
- genalert 86
- guestacct 52
- logonalert 53
- lsserver 86
- messenger 86
- netlogon 86
- netrun 86
- remoteboot 86
- replicator 86
- requester 87
- server 87
- timesource 87
- ups 87

session

- disconnecting 51
- inactive 51
- maximum open 54, 55, 65
- NetBIOS 32, 39
- timeout 38

sesstimeout parameter 38

- shared RAM
 - IBM 16/4 Token-Ring Adapter 16
- shared resource
 - concurrent file and print requests 57
 - inactive connections 37
 - locking files 39
 - maximum per server 56
 - maximum requests per server 55
- SHARELOCKS parameter 92
- Sideband 6
- signals parameter 79
- sizalertbuf parameter 73
- sizcharbuf parameter 36
- SIZEBIGBUF parameter 89
- SIZEMEMBUF parameter 92
- sizerror parameter 36
- sizmessbuf parameter 73
- sizreqbuf parameter 38, 58
- SIZWORKBUF parameter 89
- SMB (server message block) protocol
 - core 2
 - correct format 60
 - definition 2
 - error causing 52
 - file I/O 6
 - large file transfers 42
 - multiplexed 2
 - RAW 2
 - RAW read and write 42
- SMB (system message block) protocol 2
- svanndelta parameter 58
- svannounce parameter 58
- srvheuristics parameter
 - deny-write sharing mode 44
 - description 58, 66
 - performance tuning 14
 - Server section 58
 - Server section 386 HPFS 66
- srvhidden parameter 58
- srvnets parameter 53
- srvpipes parameter 72
- srvservices parameter 53
- statistics
 - cache 108
 - cache usage 14
 - checking 13
- storage management 97
- swappath parameter 100

T

- TASKTIMESLICE parameter 92
- TCPBEUI 6
- thread 35, 57, 59
- timesource parameter 87
- TIMESYNC parameter 91
- transferring file 57, 62, 66
- tryuser parameter 77, 83
- Tuning Assistant 10, 13

U

- update parameter 70
- ups parameter 87

- UPS service
 - system parameters 79
 - tuning parameters 80
- UPS service parameters
 - batterymsg 80
 - batterytime 80
 - cmdfile 79
 - cmdtimer 80
 - devicename 79
 - lowbattery 79
 - messdelay 80
 - recharge 81
 - signals 79
 - voltlevels 80
- useallmem parameter 34
- user
 - guest account 52
 - maximum per server 56
 - NET.ACC file 70
 - user and group definitions file 70
- user ID
 - alert messages 50
 - DCDB Replicator service 82, 83
 - guest 52
 - ID 77, 83
 - Replicator service 76, 77
- username parameter 85
- USERNAME parameter 89

V

- virtual circuit 39, 53
- voltlevels parameter 80

W

- write-behind
 - definition 59
 - in 386 HPFS Option 5
 - in srvheuristics parameter 59
 - in wrkheuristics parameter 42
- wrkheuristics parameter 38
- wrknets parameter 34
- wrkservices parameter 34

X

- XMITSIZE parameter 92



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.