

LANDP® Family



Servers and System Management

Version 5.0

LANDP® Family



Servers and System Management

Version 5.0

Note

Before using this information and the product it supports, be sure to read the general information under Appendix A, "Notices" on page 259.

First Edition (April 2000)

This book is based on the previous edition, *LANDP Servers and System Management Version 4.0*, SC33-1963-00, which remains applicable and current for users of LANDP® Version 4.0.

This edition applies to LANDP Family Version 5 (part number 0781197 in the United States of America, program number 5639-190 in Europe, the Middle East, and Africa) and to all subsequent releases and modifications, until otherwise indicated in new editions. Make sure you are using the correct edition for the level of product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the addresses given below.

At the back of this publication is a page titled "Sending your comments to IBM". If you want to make comments, but the methods described are not available to you, please address them to:

IBM United Kingdom Laboratories, User Technologies,
Mail Point 095, Hursley Park, Winchester, Hampshire, England, SO21 2JN.

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1992, 2000. All rights reserved.**

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

Contents

About this book	xiii
Who should read this book	xiii
What you need to know	xiii
How this book is organized	xiii
Conventions used in this book	xiv
DB2 Universal Database®	xiv
Windows 2000	xiv
Related information	xv
Web site	xv
Summary of Changes	xvii
Chapter 1. Introduction	1
Chapter 2. Wide area communication servers	3
Wide area communication options	3
Wide area communication support and data link control	4
Support for LANDP for DOS	4
Support for LANDP for OS/2	10
Support for LANDP for Windows NT	11
Support for LANDP for AIX	12
Asynchronous operation	12
Number of buffers for the data link controls	13
Translating data	13
Local session IDs for SNA and native X.25 servers	13
Cryptographic interface	15
SNA server	16
LU pooling support for LANDP for DOS, OS/2, and Windows NT	16
Special considerations for LANDP for AIX	18
Compression server	19
Native X.25 communication server	19
Special considerations for LANDP for OS/2 native X.25 server	20
Program-to-program communication server	20
Conversation event handles for the PPC server	22
TCP/IP wide area communications server	22
MQSeries Link server	23
Chapter 3. LANDP emulator support	25
LANDP 3270 emulator	25
Unsupported facilities	26
Assigning keys	26
DBCS considerations	27
Operations	28
Application programming interfaces for the IBM 3270 environment	38
LANDP 3270 emulator high-level language API (HLLAPI)	38

LANDP 3270 emulator API	38
LANDP 3287 printer emulator	38
DBCS considerations	41
Operations	41
Chapter 4. Remote change management services	43
Identifying files	46
Creating RCMS.LNF	47
Modifying RCMS.LNF	47
Translating data	48
Requesting phase remote release	48
Tivoli NetView DM commands	52
DELETE	53
INITIATE CLIST	54
REPLACE	57
RETRIEVE	58
RETRIEVE by CATALOG	60
SEND	60
SEND MESSAGE	61
Reporting process results	62
Chapter 5. I/O device servers	67
Financial printer server	68
Special considerations for LANDP for DOS	69
Special considerations for LANDP for OS/2	70
Special considerations for LANDP for Windows NT	71
Special considerations for LANDP for AIX	71
IBM 4748 printer server	72
Special considerations for LANDP for DOS	72
Special considerations for LANDP for OS/2	72
Special considerations for IBM DOS H7.0 and IBM OS/2 Warp H3.0	73
IBM 4770 printer server	73
Printer manager server	74
Modes of operation	74
Using the operator interface and the local resource manager	74
The printer manager server and DOS PRINT	75
Magnetic stripe reader/encoder server	76
Personal identification number (PIN) pad server	77
Chapter 6. Data management servers	79
Shared-file server	79
Shared-file access methods	80
The file pointer	81
Disk space management	81
Shared-file server structure	81
Log files	82
Segmented keys	89
Shared-file server lock management	89

DBCS mode restrictions	90
External Logging Replicator (XLR)	90
Shared-file server utilities	97
Shared-file distributor server	110
Shared-file replicator server	112
Query server	114
Operating modes	115
Query server log file	115
Query server utilities	117
ODBC query server	120
Operating modes	121
ODBC conformance levels	121
ODBC data source creation	122
ODBC query server utilities	122
OPENODB - creating connection criteria	122
Electronic journal server	123
Electronic journal files	124
Data integrity for LANDP for DOS, OS/2, and Windows NT electronic journal files	125
Data integrity for LANDP for AIX electronic journal files	125
Electronic journal environments	126
Search operations	126
DBCS mode restrictions	127
Electronic journal server utility	128
Store-for-forwarding server	128
Store-for-forwarding files	130
Data integrity for LANDP for DOS, OS/2, and Windows NT store-for-forwarding files	131
Data integrity for LANDP for AIX store-for-forwarding files	131
Forwarding stored records	132
Search operations	133
DBCS mode restrictions	133
Forwarding server	134
Transmission	134
Host computer considerations	136
DBCS mode restrictions	137
Shared DOS directory services	137
Creating a shared DOS directory	138
Utilities	138
Protecting your shared files	141
System security	141
Chapter 7. System management servers	145
System manager server	145
User identification and control functions	146
User session maintenance and control	147
Network SignON Coordinator/2 exit routine feature (OS/2 and DOS only)	147
User profile and application data maintenance	147

Date and time synchronization	148
LANDP workgroup common data maintenance	148
Retrieving defined record structures	149
Data validation with defined record structures	149
Alerts management	149
System and user log management	150
Operator messages support	151
DBCS mode restrictions	151
System manager operator	151
Customizing the system manager operator	152
Working with the LANDP for DOS system manager operator	152
Working with the LANDP for OS/2 system manager operator	165
Operator interface	182
Activating the operator interface	182
Messages and return codes	182
Presentation panels	183
Status messages	183
Command set	185
Function keys	189
Sample panels	190
Local resource manager server	192
Maximum number of LU sessions and printers	192
Service availability manager (SAM)	193
Chapter 8. Application integration servers	197
CICS interface server	197
DDE access server	198
Using the clipboard services	199
Batch machine facility	199
Accessing the batch machine	200
Processing jobs in batch mode	200
Batch machine log file management	200
MAIL program	200
Mail program commands	200
Object post box server	204
Batch machine operator	204
LANDP-DCE interoperability	206
LANDP-DCE interoperability characteristics	208
Performing a dce_login	208
Chapter 9. LANDP facilities and utilities	211
File transfer facility	211
Workstation requirements	212
Host computer requirements	212
Compatibility with other emulators	212
Using the file transfer facility commands	212
File transfer messages	223
Simplifying SEND and RECEIVE program use	224

Compatibility with other emulators	225
3270 send/receive facility	225
Workstation requirements	226
Host computer requirements	226
Using the 3270 send/receive facility commands	227
4707 monochrome display support	232
Changing mode using IBM DOS commands	233
Multiple virtual DOS machine relays	233
Microsoft Windows 3.1 support	234
Considerations for Microsoft Windows 3.1 support	235
General utility programs	236
VERSION utility program	236
APPLYFIX utility program	236
System verification utility programs	237
Displaying server status	237
Displaying LANDP for AIX status	239
Translating between ASCII and EBCDIC	239
LANDP for AIX record definition utility program	240
Chapter 10. Migration	245
Query server from FBSS/2 to LANDP for OS/2	245
Shared-file server to LANDP for OS/2 query server	246
LANDP for OS/2 query server to LANDP for Windows NT ODBC query server	247
Shared-file server and SQL data to LANDP for AIX query server	247
LANDP for OS/2 data migration utilities	248
EHCMGR1	249
EHCOS2Q and EHCOSQ	251
EHCMGROS and EHCMGR2	251
MIG.EXE data migration utility	252
Migration utility menu	253
Files required to perform a migration	253
Migrating shared-file data to SQL data	254
Migrating LANDP for OS/2 SQL data	256
Appendix A. Notices	259
Trademarks and service marks	261
Glossary	263
Bibliography	285
IBM LANDP Family	285
IBM Financial Branch System Services Licensed Programs	285
IBM Financial Branch System Integrator Licensed Programs	285
IBM Transaction Security System	285
Banking Self-Service	285
IBM workstations	286
IBM RISC System/6000®	286
IBM Local Area Network	287

IBM 3270	287
Wide Area Communications	287
IBM NetView	288
IBM Financial I/O Devices	288
Distributed Computing Environment	289
Encryption and Decryption	289
IBM VisualAge C++	289
IBM VisualAge Generator	289
IBM VisualAge Smalltalk	289
Java	289
IBM Personal Communications	290
IBM Communications Server	290
WorkSpace On-Demand	290
MQSeries	290
Index	291

Figures

1.	Communication Server Overview (LANDP for DOS)	5
2.	Communication Options (LANDP for OS/2)	10
3.	Communication Options (LANDP for Windows NT)	11
4.	Communication Options (LANDP for AIX)	12
5.	MQSeries Link server in a LANDP workgroup	23
6.	RCMS/NetView DM	45
7.	External logging replicator, servers	91
8.	External logging replicator, servers after takeover	92
9.	Shared file server with external log replicator (XLR)	95
10.	Shared-file server log file header record	99
11.	Shared-file server log file detail record	101
12.	Information from shared-file server statistics file	106
13.	Shared-file server profile definition	108
14.	Shared-file distributor server	110
15.	Shared-file distributor server topology	111
16.	Shared-file replicator server	112
17.	Example of shared-file distributor/shared-file replicator topology	113
18.	Store-for-forwarding Operation	129
19.	LANDP for DOS SMOP Logon panel	153
20.	LANDP for DOS SMOP main menu	154
21.	LANDP for DOS SMOP System Status and Commands panel	155
22.	LANDP for DOS SMOP LAN Common Data panel	157
23.	LANDP for DOS SMOP User Profile: System Data panel	159
24.	LANDP for DOS SMOP User Profile: Application Data panel	160
25.	LANDP for DOS SMOP System Log File: Search Criteria panel	162
26.	LANDP for DOS SMOP System Log File: Browse panel	163
27.	LANDP for DOS SMOP Display of Messages for the Operator panel	164
28.	LANDP for OS/2 SMOP Logon window	165
29.	LANDP for OS/2 SMOP main window	167
30.	LANDP for OS/2 SMOP LAN System Status and Common Data window	168
31.	LANDP for OS/2 SMOP LAN System Status and Common Data, File pull-down	169
32.	LANDP for OS/2 SMOP LAN System Status window	170
33.	LANDP for OS/2 SMOP LAN Common Data, Browse and Update window	171
34.	LANDP for OS/2 SMOP User Profile Data window	172
35.	LANDP for OS/2 SMOP User Profile Data, Users pull-down	173
36.	LANDP for OS/2 SMOP User Profile: System Data window	173
37.	LANDP for OS/2 SMOP Delete User pop-up window	174
38.	LANDP for OS/2 SMOP User Application Program Data window	175
39.	LANDP for OS/2 SMOP System Log File window	176
40.	LANDP for OS/2 SMOP System Log File, View pull-down	177
41.	LANDP for OS/2 SMOP Search Log Item entry window	178
42.	LANDP for OS/2 SMOP Log Search Range Criteria entry window	179
43.	LANDP for OS/2 SMOP System Log File, Options Pull-down	180
44.	LANDP for OS/2 SMOP Messages for the Operator window	181

45.	Operator Interface Status and Command panel: 3287 Printer Emulator . . .	190
46.	Operator Interface Status and Command panel: Printer Manager	191
47.	LANDP-DCE Interoperability	207
48.	MIG.EXE main menu	253

Tables

1.	Exclusive LU Management	17
2.	LU Pooling Management	18
3.	SBCS operator information	30
4.	DBCS operator information	31
5.	RCMS.LNF record format	46
6.	RCMS.LNF example: one qualifier	46
7.	RCMS.LNF example: two qualifiers	47
8.	RCMS.LNF example: three qualifiers	47
9.	RCMSPLAN files record format	51
10.	Examples of DELETE command	53
11.	CLISTEXE.STS record format before processing	56
12.	CLISTEXE.STS record format after processing	56
13.	CLISTEXE.STS record format for non-zero return codes	56
14.	Examples of REPLACE command	57
15.	Examples of RETRIEVE command (1)	58
16.	Examples of RETRIEVE command (2)	59
17.	Examples of RETRIEVE by CATALOG command	60
18.	Examples of SEND command	61
19.	Log entry common part	62
20.	Event number description	63
21.	Log entry variable part	63
22.	Log entry variable part	63
23.	Log entry variable part	64
24.	Error class description	64
25.	Log entry variable part	64
26.	Log entry variable part	65
27.	Log entry variable part	65
28.	Identification of files used by RCMS	65
29.	Commands for Changing System Status and Data	156
30.	Operator interface commands	185
31.	EHCMGR1 field type conversion: RDF to SQL	248
32.	EHCMGR1 error messages	250
33.	Field Conversions: Shared-File to LANDP for OS/2 SQL	255
34.	Field Conversion: Shared-File to LANDP for AIX SQL	256
35.	Field Conversion: LANDP for OS/2 to LANDP for AIX	257

About this book

This book provides information about the following IBM® LAN Distributed Platform (LANDP®) Family products:

- LANDP Family Version 5.0
with its components:
 - LANDP for DOS
 - LANDP for OS/2®
 - LANDP for Windows NT
- IBM LANDP for AIX®, Version 2 Release 1.0 (LANDP for AIX)

Who should read this book

The information provided in this book is intended for people who manage LANDP installations, and for anyone who requires detailed information about the LANDP servers.

What you need to know

You should be familiar with the operating systems that support your LANDP environment. In addition, you should be familiar with LANDP distributed client/server processing and how to install and customize a LANDP system.

How this book is organized

This book contains the following chapters:

- Chapter 1, “Introduction” provides an introduction to the LANDP servers and systems operations.
- Chapter 2, “Wide area communication servers” describes the servers used to provide communications with host computers, workstations in LANDP workgroups, and any other connected processors.
- Chapter 3, “LANDP emulator support” describes the emulators provided with LANDP, and their APIs.
- Chapter 4, “Remote change management services” describes the RCMS system which is used to manage LANDP software and data sources.
- Chapter 5, “I/O device servers” describes the servers used to support input and output devices in LANDP.
- Chapter 6, “Data management servers” describes the servers used to manage data in a LANDP workgroup.
- Chapter 7, “System management servers” describes the servers and services provided to help manage a LANDP system.

About this book

- Chapter 8, “Application integration servers” describes the servers provided to enable access to non-LANDP environments, and to enable resource and data sharing with non-LANDP programs.
- Chapter 9, “LANDP facilities and utilities” describes the features, facilities, and utilities provided by LANDP.
- Chapter 10, “Migration” describes the migration paths provided by LANDP which enable you to upgrade your current configuration.

As well as these ten chapters, there is a glossary, a bibliography, and an index at the back of the book.

Conventions used in this book

A graphic, like the one shown here, appears in the margin at the beginning of each major section.



This graphic shows the component or components of the LANDP family to which the section relates. This example shows that the paragraph relates to LANDP for DOS and OS/2 only. This does not necessarily imply that the item being described *runs* on each platform shown, but it does indicate that it *affects* LANDP on each platform.

Individual paragraphs may also show an icon such as the following:  This example shows that the paragraph relates to LANDP for DOS only.

DB2 Universal Database®

In this book, all references to DB2® or DB2/2 apply to IBM DB2 for OS/2 and to IBM DB2 Universal Database®.

Windows 2000

In this book, all references to Windows NT apply to Microsoft Windows NT and to Microsoft Windows 2000.

Related information

The LANDP family is supported by the following books. In this book, references to other LANDP books use the shortened title shown here. For the full title, order number of these publications, and a comprehensive list of LANDP-related literature, refer to “Bibliography” on page 285.

LANDP Introduction and Planning

This book provides a brief description of the components and features of the LANDP family, and gives information about planning a LANDP system.

LANDP Installation and Customization

This book provides information about installing, customizing, and distributing the LANDP family.

LANDP Programming Reference

This book describes the application programming interfaces that are used to develop user servers and client applications.

LANDP Programming Guide

This book gives guidance on writing application programs to use the interfaces described in the *LANDP Programming Reference*.

LANDP Problem Determination

This book describes how to use trace tools, diagnostic programs, alerts, and return codes to debug code while developing LANDP applications and user servers, or resolve problems while using LANDP family components.

LANDP Servers and System Management

This book provides detailed information on the LANDP servers, and describes how to manage and administer a LANDP system.

Web site

For more information about LANDP please visit our web site at:
<http://www.ibm.com/software/ts/landp/>

About this book

Summary of Changes

This manual has been updated to reflect enhancements made to LANDP in Version 5. The major changes in this version are:

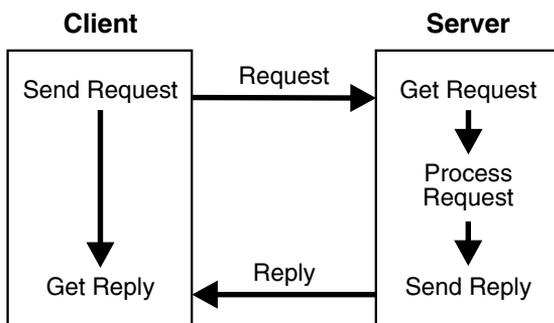
- The LANDP MQSeries Link server enables LANDP applications to access the Message Queueing Interface of MQSeries®
- The LANDP TCP/IP wide area communications server enables existing SNA wide area communication networks to be replaced with TCP/IP networks without impact to LANDP applications interfacing to the LANDP SNA or PPC servers. The TCP/IP wide area communications server also supports LANDP's 3270 emulator over the TELNET protocol.
- The LANDP ODBC query server on Windows NT supports access to various relational databases through the LANDP API using industry standard ODBC drivers.
- The External Logging Replication (XLR) feature of the Shared File server, when used with the Service Availability Manager, provides improved performance and availability of replicated Shared File databases.
- The enhanced Java support enables access to LANDP services from devices not running LANDP code, for example, browser-based applications.
- Support for the IBM 9069 transaction printer has been added.
- The range of servers supported by LANDP on the Windows NT platform has been extended to be more comparable to the function available on OS/2. The additional servers available on Windows NT include Electronic Journal, Store for Forwarding/forwarding, System Manager, PPC and the 4748 DBCS printer servers.
- In addition to the new function which LANDP V5 delivers, the levels of operating systems and other system software with which LANDP operates have been updated.

Chapter 1. Introduction



LANDP is a client/server product. Any requester of services is a client and any provider of services is a server. Applications that are developed for a LANDP environment request services from other processes called servers. This also means that when a server requests services it too becomes a client.

When clients request services, the client/server mechanism routes the request to the server, the server processes requests, and the client/server mechanism routes the replies back to the client.



The LANDP servers provide support for a variety of services:

- **Communications**

LANDP's communications services include a series of wide area communication servers and emulators. These are described in Chapter 2, "Wide area communication servers" on page 3 and Chapter 3, "LANDP emulator support" on page 25.

- **I/O device support**

Input/Output (I/O) device servers support a variety of I/O devices. The supported devices are directly connected to the workstation or system that has the corresponding device driver and server installed. These are described in Chapter 5, "I/O device servers" on page 67.

- **Data management**

Data management servers provide support to store, retrieve, and update data in LANDP workgroups. They are described in Chapter 6, "Data management servers" on page 79.

- **System management**

System management servers, together with a system management operator and operator interface, provide services that a LANDP workgroup administrator needs. These services allow you to define and maintain user profiles, to synchronize date and time, and to maintain common data in a LANDP workgroup. These services are described in Chapter 7, “System management servers” on page 145.

- **Application integration**

Application integration servers, together with LANDP’s batch machine facility, enable access to non-LANDP environments and simplify resource and data sharing with non-LANDP programs. These services are described in Chapter 8, “Application integration servers” on page 197.

There is also information about:

- The use of Remote Change Management Service (RCMS) to manage LANDP software and data sources. See Chapter 4, “Remote change management services” on page 43.
- Other features, facilities, and utilities which are used in the maintenance of a LANDP system. See Chapter 9, “LANDP facilities and utilities” on page 211.
- Migrating from an earlier version of FBSS or LANDP. See Chapter 10, “Migration” on page 245.

Chapter 2. Wide area communication servers



The LANDP wide area communication servers provide the communication services required by other servers or by client applications. This support includes connections to host computers, workstations in other LANDP workgroups, or any processors connected through supported links.

A client application, running in one of the workstations in the LANDP workgroup, can communicate with a host application or an application running in a workstation in another LANDP workgroup.

Several communication servers supporting different communication links and protocols can be present in a LANDP workgroup or workstation. For example, a LANDP SNA server, a LANDP PPC server, and a LANDP X.25 native server can be present in the same LANDP workgroup or in the same workstation.

Wide area communication options



The LANDP family wide area communication servers support a variety of possible configurations:

- Communication support limited to a single workstation.
- Communication support for an entire LANDP workgroup.
- Multiple communication servers installed in a LANDP workgroup to provide wide area communication with:
 - One or more hosts
 - A workstation in another LANDP workgroup
 - Any processor connected through a supported link

The communication servers can be of the same type or different types. Typically, one host processes the transactions originating in the production site, then other communication servers can be used to access other systems, such as:

- A backup system
- Publicly available information, such as the Internet
- Relational databases providing management information

The LANDP family provides the following wide area communication servers:

- System Network Architecture (SNA) server
- Native X.25 communication server (for LANDP for DOS and OS/2)
- Program-to-program communication (PPC) server (for LANDP for OS/2, Windows NT, and AIX)
- TCP/IP wide area communication server (for LANDP for DOS, OS/2, and Windows NT)

wide area communication, DOS

An SNA server is provided for each supported operating system (DOS, OS/2, Windows NT, and AIX). These servers use the LANDP common API. This means that your applications require little or no modification to send requests to any SNA server installed in your LANDP workgroup, regardless of the server's location and the operating system on which it runs.

Wide area communication support and data link control



The wide area communication functions are built in three layers:

- Protocol handling layer
- Data link control layer
- Physical layer provided by a communication control adapter

These layers operate together in one workstation.

Support for LANDP for DOS



LANDP for DOS supports:

- Synchronous data link control (SDLC)
- Token-ring data link control (TRDLC)
- X.25 data link control (X25DLC)
- If FBSI is installed, device cluster attachment data link control (DCADLC) is also supported

These link protocols are supported using IBM® communication adapters. The communication server must be in the workstation where the communication adapter is installed.

Because the DOS operating system does not have the required data link control support, the SDLC, TRDLC, and X25DLC servers are included in LANDP for DOS. These servers provide the required data link control services and the interface to the communication adapters. These servers are internally used by the LANDP for DOS SNA and native X.25 servers.

When selecting your system components, consider the hardware interrupts required by the various adapters. An inadequate choice of equipment can result in interrupt conflicts.

It is also important that the parameters selected for the communication servers correspond to the parameter definitions in the host. See the *LANDP Installation and Customization* for more information.

The following figure shows how the LANDP for DOS communication servers interact with the data link controls.

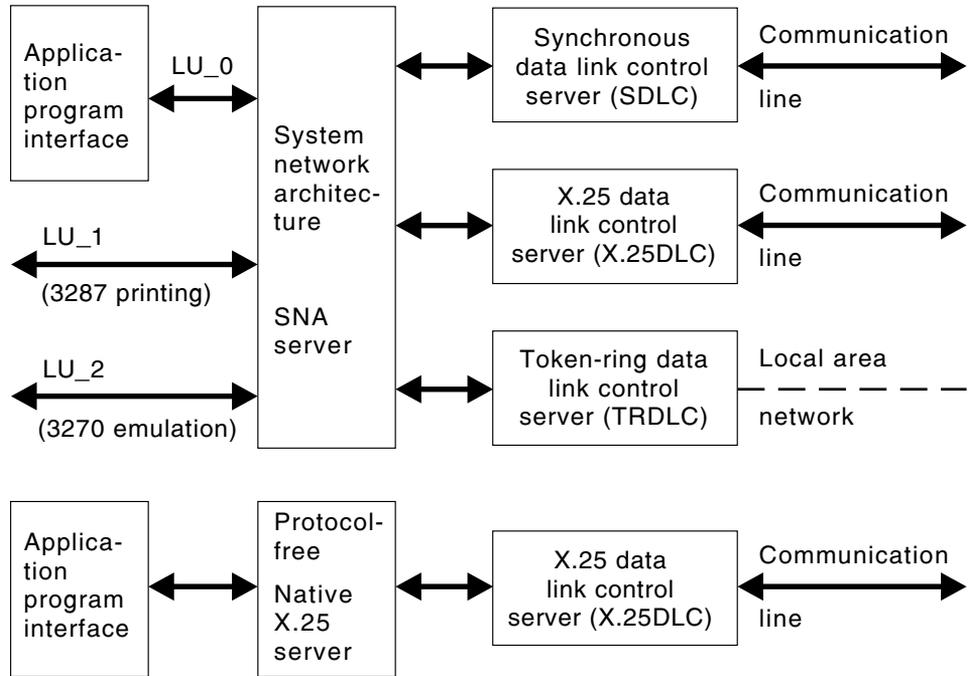


Figure 1. Communication Server Overview (LANDP for DOS)

The TCP/IP wide area communication server does not use the data link control server supplied with LANDP. It uses the TCP/IP implementation of PC/TCP to provide the data link control layer.

SDLC server

SDLC operates as a half-duplex secondary on a point-to-point or multipoint line. Leased and manually dialed connections are supported.

For personal workstations such as IBM ThinkPads or ValuePoint machines, the maximum supported line speeds are the same as those supported by the adapters.

For older machines, the maximum line speeds may be lower. The table below shows some examples of maximum line speeds for a selection of personal workstations.

19200	PC AT®, Personal System/2 Models 30(H21), 50 and above
9600	PC XT/286, Personal System/2 Models 30(002) and 30(021)
4800	PC and PC XT™

Token-ring data link control server

Token-ring data link control is included in LANDP for DOS, enabling SNA communication using a token-ring IEEE 802.2 session.

Token-ring data link control supports communication with:

- IBM 3720, 3725, and 3745 Communication Controller
- IBM 3174 Establishment Controller
- IBM 4702 Branch Automation Processor
- IBM 9370 Information System
- IBM AS/400

The SNA/token-ring communication server configuration can be configured as:

- One specific computer acting as the SNA/token-ring gateway for the entire LANDP workgroup. Only one token-ring session is used to multiplex all the SNA/token-ring traffic.
- Each computer having its own SNA/token-ring server by using an independent token-ring session to communicate with the primary computer.
- An intermediate solution between these two extremes.

Each SNA/token-ring server corresponds to a different physical unit (PU). The network addressable unit configuration, seen by the primary SNA computer, depends on the parameters chosen at customization (single or multiple PUs and their relationship to the LUs).

Performance requirements, availability of storage in the computers, and other criteria influence the choice of SNA/token-ring configuration.

X.25 data link control server

The support for X.25 data link control, used by the SNA/X.25 and native X.25 servers, is provided as follows:

IBM PC X.25 Communications Adapter

Designed to operate with the IBM Personal System/2 Model 30 and IBM PC, PC XT, PC XT286, and PC AT. Software support is provided by the IBM PC X.25 Communications Support Program.

IBM X.25 Interface Co-Processor/2

Designed to operate with the IBM Personal System/2 models using Micro Channel® architecture.

Software support is provided by:

- IBM X.25 Co-Processor Support Program
- Realtime Interface Co-Processor DOS Support

IBM X.25 Interface Co-Processor

Designed to operate with ISA-architecture (AT-bus) computer systems, including ValuePoint systems, using AT-bus technology.

Software support is provided by:

- IBM X.25 Co-Processor Support Program
- Realtime Interface Co-Processor DOS Support

The adapter type determines the speed limit of the network communication.

Only one X.25 adapter or coprocessor can be installed in a computer, but several adapters and coprocessor can be present in a LANDP workgroup.

The hardware requirements are the same for the SNA/X.25 and native X.25 communication servers.

Multiple virtual circuits using the same physical network connection are accepted. Multiple PU support for SNA, multiple native connections, and a mixture of both are supported. The X.25 data link control supports up to 20 concurrent virtual circuits. You select the network connection and virtual circuit parameters during customization. These must agree with the Post Telephone and Telegraph Administration (PTT) parameters. The application can control the virtual circuit connection process and parameters.

The functions provided by the SNA/X.25 and the native X.25 servers are not influenced by the choice of X.25 support. The application programming interfaces remain unchanged.

The X.25 Co-processor can work with networks according to the CCITT 1980 or CCITT 1984 recommendations. For compatibility reasons, however, the application works according to the CCITT 1980 recommendations.

The function of the trace tool varies depending on whether an adapter or coprocessor is used. When you use the X.25 adapter, the trace data is always being collected; when you use the coprocessor, data collection is started and stopped by the trace tool user. Performance improves when the trace function is not operating. When using the coprocessor, the frame level and packet level are displayed.

If you unload LANDP for DOS with the FREE function while using the X.25 coprocessor, the basic support software is still loaded. You must use the X.25 TERM function to free all memory. This also avoids later loading errors. Consult the *IBM X.25 Co-Processor Support Program User's Guide*.

Number of buffers for LANDP for DOS data link controls

SDLC, TRDLC, and X25DLC servers manage the message buffer pools. Each buffer contains:

- 265 bytes for SDLC
- 256 bytes for TRDLC
- 128 bytes for X25DLC

wide area communication, DOS

The number of buffers for each communication server is defined during customization. Make the buffers as large as possible, within the limitations imposed by storage and the customization program. The memory for each buffer, including its corresponding header, occupies:

- 290 bytes for SDLC
- 272 bytes for TRDLC
- 144 bytes for X25DLC

Defining an insufficient number of buffers leads to:

- The application getting the corresponding nonzero return code when sending data
- Decreased performance and, in extreme cases, a loss of contact with the host when receiving data

The number of buffers depends on the size of the request unit (RU) of the messages that will be interchanged. The maximum RU size is 4096 for all LU types.

The formula used to calculate the number of buffers needed is the same for all types of data link control.

If the pacing defined for each type of SNA LU is the same, the following formulas can be used to calculate the number of required buffers for pacing values 1 or higher:

$$\begin{aligned} &\text{Number of LU}_0 * (\text{RU size} / \text{Buffer size}) * (\text{Pacing} + (\text{Pacing} - 1)) \\ &\text{Number of LU}_1 * (\text{RU size} / \text{Buffer size}) * (\text{Pacing} + (\text{Pacing} - 1)) \\ &\text{Number of LU}_2 * (\text{RU size} / \text{Buffer size}) * (\text{Pacing} + (\text{Pacing} - 1)) \end{aligned}$$

Note that the buffer size varies depending on the type of data link control.

Number of buffers (265 bytes) for SDLC: The number of buffers required for SDLC depends on the following parameters defined during customization:

- Number of possible SNA LU_0 concurrent application sessions in the entire LANDP workgroup
- Number of possible 3287 printer emulation sessions (LU_1) in the entire LANDP workgroup
- Number of possible 3270 emulation sessions (LU_2) in the entire LANDP workgroup
- Pacing defined for each LU
- Outbound RU size defined

Number of buffers (256 bytes) for TRDLC: The number of buffers required for TRDLC depends on the following parameters defined during customization:

- Number of possible SNA LU_0 concurrent application sessions in the entire LANDP workgroup
- Number of possible 3287 printer emulation sessions (LU_1) in the entire LANDP workgroup
- Number of possible 3270 emulation sessions (LU_2) in the entire LANDP workgroup
- Pacing defined for each LU
- Outbound RU size defined

The number of buffers is limited by the buffer pool size. The maximum buffer pool size is 64KB.

Number of buffers (128 bytes) for X25DLC: The number of buffers required for X25DLC depends on the following parameters defined during customization:

- Number of possible native X.25 concurrent application sessions in the entire LANDP workgroup
- Number of native X.25 messages that can be received without confirmation for each application session
- Number of possible SNA LU_0 concurrent application sessions in the entire LANDP workgroup
- Number of possible 3287 printer emulation sessions (LU_1) in the entire LANDP workgroup
- Number of possible 3270 emulation sessions (LU_2) in the entire LANDP workgroup
- Pacing defined for each LU
- Outbound RU size defined

The total number of required buffers for X25DLC equals the number of buffers required for SNA/X.25 plus the number of buffers required for native X.25 (if both are used).

If the number of native X.25 messages that can be received without confirmation is the same for each application, the following formula can be used to calculate the number of required buffers for native X.25:

Number of session * (Message size/Buffer size) * Number of messages

Support for LANDP for OS/2



The wide area communication functions in LANDP for OS/2 (SNA, native X.25, and PPC, but not TCP/IP) are based on IBM SecureWay® Communications Server for OS/2 Warp Version 6.

Figure 2 shows the interaction between LANDP for OS/2 and IBM Communications Server for OS/2 Warp.

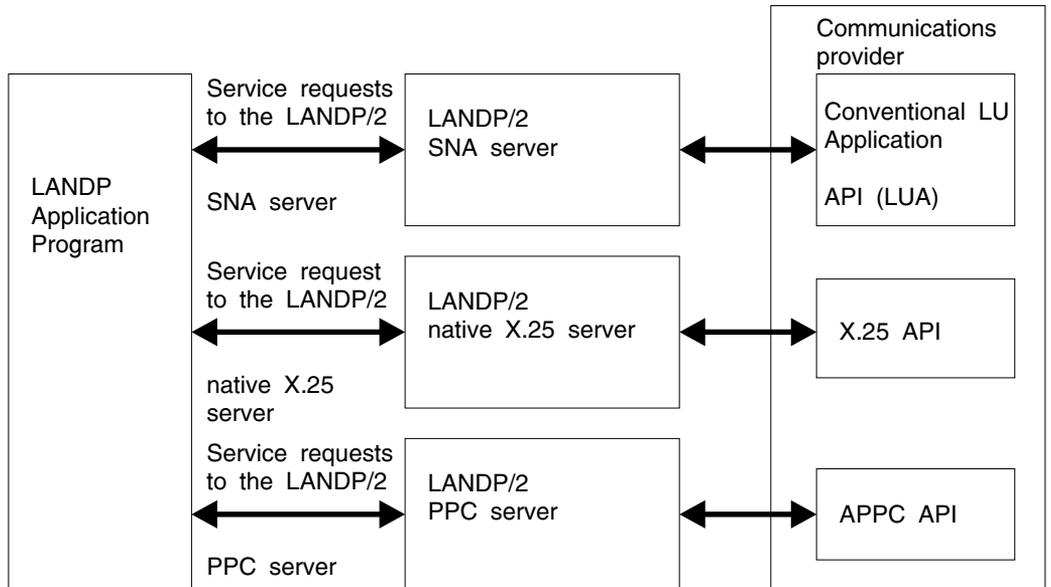


Figure 2. Communication Options (LANDP for OS/2)

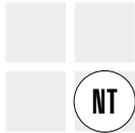
The service requests to the SNA server are transformed to use the conventional LU application API (LUA) of IBM Communications Server for OS/2 Warp.

The service requests to the PPC servers are transformed to use the Advanced Program-to-Program Communication (APPC) API functions of Communications Server for OS/2 Warp.

The service requests to X.25 servers are transformed to use the X.25 API functions of Communications Server for OS/2 Warp.

The TCP/IP wide area communications server uses the TCP/IP implementation of OS/2 Warp.

Support for LANDP for Windows NT



The SNA wide area communication functions in LANDP for Windows NT (SNA and PPC but not TCP/IP) are based on the following products:

- IBM SecureWay Personal Communications Version 4.3 for DOS/Windows, Windows 95, and Windows NT.
- IBM SecurerWay Communications Server for Windows NT, Version 6.0
- Microsoft SNA Server, Version 4.0

Figure 3 shows the interaction between LANDP for Windows NT and these SNA providers.

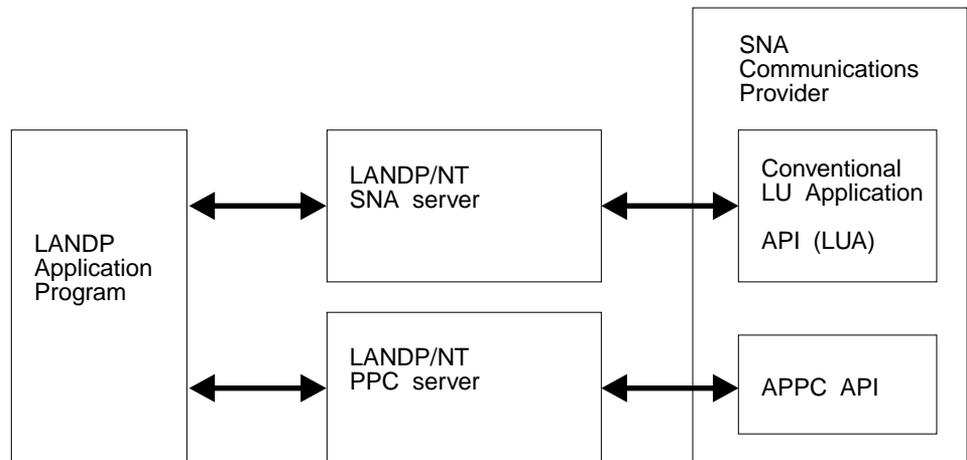


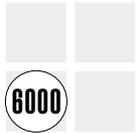
Figure 3. Communication Options (LANDP for Windows NT)

The service requests to the SNA server are transformed to use the conventional LU application API (LUA) of the SNA providers.

The TCP/IP wide area communications server uses the TCP/IP implementation of Windows NT.

wide area communication

Support for LANDP for AIX



The wide area communication functions in LANDP for AIX use the IBM AIX SNA Services/6000 or AIX SNA Server/6000 products. Figure 4 shows the interactions between LANDP for AIX and IBM AIX SNA Services/6000.

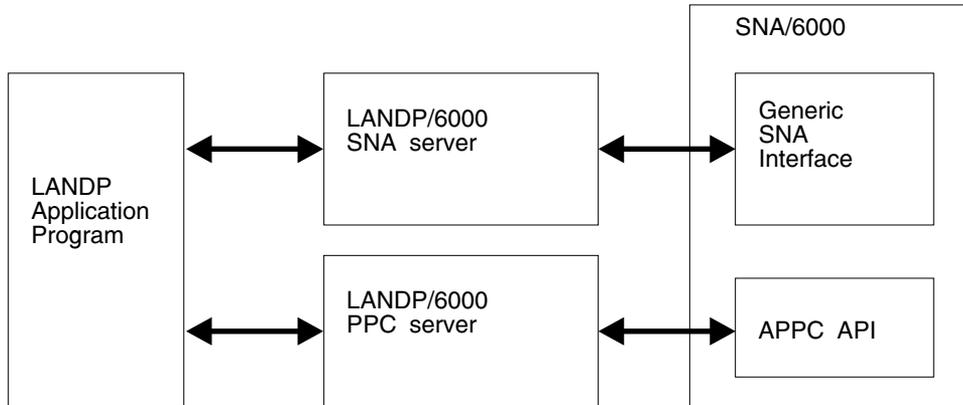


Figure 4. Communication Options (LANDP for AIX)

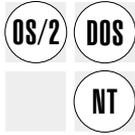
Asynchronous operation



The data link controls manage incoming and outgoing messages for the communication servers. The data link controls and the communication servers operate asynchronously.

As a result the data link controls have a queue of input and output messages in their buffers for transmission to the communication servers and the applications. This means that control returns to the application while the data link control independently processes the request. The data link controls send and receive messages, and manage polling, various time-outs, and the connection process.

Number of buffers for the data link controls



The number of buffers for each data link control has to be defined. For LANDP for DOS, they are defined during the customization process. Refer to “Number of buffers for LANDP for DOS data link controls” on page 7 for more information.

For LANDP for OS/2 and LANDP for Windows NT communication servers, the number of buffers is defined when the appropriate communications support is configured.

Translating data



Personal computer workstations use ASCII character codes while many host computers use EBCDIC. The communication server functions do not include code translation of the messages. If messages contain fields in character format, they may have to be translated between ASCII and EBCDIC. Routines that translate in both directions are supplied. For more information about these routines, refer to “Translating between ASCII and EBCDIC” on page 239.

Local session IDs for SNA and native X.25 servers



A LANDP SNA or X.25 session is an abstract entity, which is defined by:

- Workstation ID. Defined at customization.
- Local session ID. Defined at customization and related to an LU for SNA or an X.25 virtual circuit.

For SNA, the local session ID is provided as a parameter in each communication function call.

When working with LANDP for DOS, local session IDs used by client applications are between 01 through 15, and those used by server applications you have developed are between 16 through 30.

When working on LANDP for OS/2, Windows NT, and AIX, sessions 01 to 30 can be used by either client or server applications.

A modified SNA interface that allows for more than 30 user sessions per workstation is available when the SNA services are provided from an OS/2 or Windows NT workstation. When using this interface, the session identifier may be any two ASCII characters.

Local session addressing permits the same application to operate in any workstation in a LANDP workgroup. At the communication server, the combination of the local session ID and the calling workstation ID defines an intersystem communication path.

wide area communication

This communication path is an LU for the SNA server. When the SNA server uses X25DLC, all the LUs associated with a PU use a single virtual circuit.

For the native X.25 server, the session ID is correlated to a subscriber address that is defined during customization. Each conversation uses a single virtual circuit. If the session ID is configured for outgoing calls, the server will send "call packets" using the subscriber address and the destination identification configured for this session. This information must match the information at the other end of the communication line.

The native X.25 server also associates incoming calls with one of the session IDs on a specific workstation, based on customization information or on information provided in the define connection (DC) function call.

Part of the general customization information defines which communication servers are used for each local session in each computer. This information is read during workstation initialization and used with the workstation ID to build the local session routing tables. These tables are used to route the local session messages to the appropriate server. The maximum number of SNA local sessions permitted in the entire LANDP workgroup per SNA server is 240 for DOS.

Cryptographic interface



By using the IBM Transaction Security System (TSS), the LANDP family supports data encrypted communications between the host and the LANDP workgroup.

Two options are supported:

- Functions of the LANDP cryptographic interface can be incorporated in client applications.

In this case, cryptographic implementations of the forwarding server, LANDP 3270 emulator, and LANDP 3287 emulator that complement the SNA server must be installed.

See the *LANDP Programming Reference* for more information about the functions associated with the cryptographic interface.

- You can use a combination of the LANDP for OS/2 SNA server, the LANDP ACSRDECR.DLL and ACSRENCR.DLL dynamic link library files, Communications Server for OS/2 Warp, and the LANDP security server provided by TSS.

In this case, client applications do not incorporate functions of the cryptographic interface, and you do not require cryptographic implementations of the forwarding server, LANDP 3270 emulator, and LANDP 3278.

When session level encryption is requested, the LANDP for OS/2 SNA server issues the required calls to the TSS security server to decrypt and encrypt session data. This lets existing SNA server applications take advantage of cryptographic services without modifying the code to use the functions of the cryptographic interface.

In addition, this feature allows the concurrent use of both SNA session level encryption and SNA compression.

Note: Any 16-bit application calling TSS with global (shared) segments will not work with LANDP for OS/2. The bypass is always to pass local data.

Cryptography is not supported by LANDP for Windows NT.

The LANDP for AIX forwarding server supports session level encryption only if the RS/6000® Transaction Security System is installed in the RISC System/6000®.

See the *LANDP Programming Reference* for additional information about the cryptographic interface.

SNA server



The SNA server provides host communication support for applications running in LANDP for DOS, OS/2, Windows NT, and AIX workstations.



The workstation in the LANDP workgroup with the SNA server installed is called a gateway. The gateway must have a connection to the host computer.

The SNA server supports the following types of LU-sessions:

- LU_0
- LU_1
- LU_2

LU_0 is supported for communication between customer-written client and server applications. One application runs in one of the workstations in the LANDP workgroup, the other runs on a remote system. Half duplex flip-flop and half duplex contention protocols are supported for SNA LU_0, LU_1, and LU_2. A system services control point-physical unit (SSCP-PU) session is available for forwarding alerts that are generated by applications or servers.

The SNA server has other characteristics:

- A communication session can be initiated by the workstation application or the host computer.
- SNA/X.25 conforms with general IBM practice and is compatible with the network packet switching interface (NPSI) support in the network control program (NCP).
- Using SDLC, Token Ring, or X.25, SNA communication with CICS® and IMS™ using VTAM® and NCP is supported.
- Communication with the IBM 4700 Finance Communication System and AS/400 over SDLC and Token Ring is also available.
- LANDP for DOS and OS/2 provide functions for changing the characteristics of an SNA X.25 connection. You can change the X.25 destination address of the SNA X.25 connection which is being used as a result of issuing these functions.
- Operator intervention is not required to bring the link up after a link failure has occurred. The operator does not have to activate the link using the SNA communications provider interface. Whenever a link failure is detected by the server, the server reactivates the link if possible.
- If you have FBSI installed, device cluster attachment (DCA) data link control (DCADLC).

LU pooling support for LANDP for DOS, OS/2, and Windows NT



The LANDP for DOS, OS/2, and Windows NT SNA servers support LU pooling. This allows LUs to be grouped into a small number of LU classes. The LU pooling support saves multiple repetitive entries in the host tables. LU pooling is managed by the SNA server, therefore applications do not have to determine which LUs they are going to use. Also, LU pooling support lets you balance the workload for all the physical units (PUs) or hosts used.

To use LU pooling support in your workgroup, it must be selected during customization. If LU pooling support is used, the SNA server associates an SNA session with a group of LU definitions. These groups of LU definitions are called pools.

If you choose exclusive LU support, the SNA server associates each SNA session with one LU definition as in previous LANDP releases.

Both types of LU management can be simultaneously supported.

Table 1 and Table 2 on page 18 show the difference between exclusive LU management and LU pooling management. Both tables show workstation AA with one LU, workstation BB with two LUs, and workstation CC with four LUs.

WS-ID	SNA Session ID	LU Definition in SNA communications provider (LANDP for OS/2 only)	Host LU Definition
AA	SNA01	EHCAA01	LU01
BB	SNA01	EHCBB01	LU02
BB	SNA02	EHCBB02	LU03
CC	SNA01	EHCCC01	LU04
CC	SNA02	EHCCC02	LU05
CC	SNA03	EHCCC03	LU06
CC	SNA04	EHCCC04	LU07

wide area communication, SNA

Table 2. LU Pooling Management

WS-ID	SNA Session ID	SNA Pool ID	LU Definition in SNA communications provider (LANDP for OS/2 only)	Host LU Definition
AA	SNA01	G1	EHCG101 EHCG102	LU01 LU02
BB	SNA01			
BB	SNA02			
CC	SNA01	G2	EHCG201	LU03
CC	SNA02			
CC	SNA03			
CC	SNA04		EHCCC04	LU04

Note: SNA04 session for workstation CC is managed using exclusive LU management. Therefore, it has a dedicated LU definition in the SNA communications provider.

With exclusive LU management you must define a LU in the host for each LU defined in the SNA communications provider (for LANDP for OS/2), SNA-supporting product (for LANDP for Windows NT), or SNA session ID (for LANDP for DOS).

Using LU pooling management results in fewer host LU definitions.

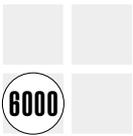
LU pooling can be useful for improving system throughput. However, LU pooling is not appropriate for certain types of sessions or host applications. For example, on some systems if a terminal with a CICS session is unexpectedly powered off, a bind automatically occurs when the session is brought up again. If LU pooling is used, the LU address may be reassigned and the resulting bind may be sent to a different terminal.

Printer spooling is another example where LU pooling can be inappropriate. If the host application associates a particular LU address with a print job and the pooled LU is assigned to a different printer, the following problems can occur:

- The print job could be sent to a printer with incompatible features
- The print job could be sent to a distant or uncontrolled location

This could cause a security breach or output that is unusable.

Special considerations for LANDP for AIX



The LANDP for AIX SNA server can share a PU with applications and user servers that use the SNA/6000 high level API. However, it cannot share a PU with programs that also use the generic SNA device driver interface.

The number of PUs for the LANDP for AIX SNA server is only limited by the number of communication adapter cards that you can install in your RS/6000.

Compression server



The compression server is used to decrease the size of normal-flow function management data (FMD) request units (RUs). This is useful when the maximum RU size is bigger than the DLC segment size because it reduces the number of transmitted segments. An RU whose composed length is greater than can be expressed in the compression header will not be compressed.

A form of SNA length-checked compression is used. Specifically, run-length encoding (RLE) is used.

This server is used as an SNA server exit. When compression is required, the SNA server calls the functions of the compression server interface. These actions are transparent to the user application.

Data must be compressed before encryption and decompressed after decryption. Therefore it is not possible to use encryption and compression for the same LU.

Native X.25 communication server



The native X.25 server provides a program-to-program, high-level, protocol-free communication using X.25 data link control. A virtual circuit may be used directly by an application. Note that all communication protocols above the virtual circuit must be provided by the application logic, that is, you must develop your own protocol functions. The native X.25 server is useful when communicating with:

- Another LANDP workgroup or a single workstation
- Non-SNA computers available through X.25 networks

The native X.25 and SNA/X.25 servers can coexist in the same workstation. They use different virtual circuits. The native X.25 hardware requirements are the same as for SNA/X.25 communication.

The native X.25 server provides the following features for applications:

- Distributed process communications with location transparency of the partner applications
- A solution to interconnect several LANDP workgroups through an X.25 network
- Easy interface and protocol that start and maintain a conversation, send and receive data, and close the virtual circuit with an X.25 partner application, using a reduced set of functions

An application can handle more than one LANDP conversation. LANDP sessions are used to serially share the X.25 virtual circuits. Applications handle the LANDP sessions that start the X.25 virtual circuits. However, the applications do not access information on X.25 configuration, parameters, and management. They access information on LANDP sessions only.

wide area communication, PPC, OS/2 AIX

See the *LANDP Installation and Customization* for more information about customizing the native X.25 server.

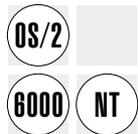
Special considerations for LANDP for OS/2 native X.25 server

The LANDP for OS/2 native X.25 server can support as many LANDP conversations as your configuration can hold. That is, the limitations are set by the memory requirements of the workstations, the traffic on the communication line, and IBM Communications Server for OS/2 Warp.

The server receives requests from applications that run in any client workstation and, when necessary, issues calls to the X.25 API of IBM Communications Server for OS/2 Warp which, in turn, sends the appropriate data to the link line. Since the communication services are provided by IBM Communications Server for OS/2 Warp, all its restrictions on communication profiles apply to the LANDP for OS/2 native X.25 server.

For proper operation of the server, a consistent configuration has to be defined and the appropriate IBM Communications Server for OS/2 Warp files for X.25 management have to be present in the appropriate directory. To achieve this objective, IBM Communications Server for OS/2 Warp should be installed or reinstalled using the correct configuration file.

Program-to-program communication server



The program-to-program communication (PPC) server lets a LAN application communicate with a partner application through an SNA LU_6.2 session. These SNA nodes are on remote systems connected through the communication facilities provided by the following communication management products:

- IBM SecureWay Communications Server for OS/2 Warp for LANDP for OS/2
- IBM AIX SNA Services/6000 or IBM AIX SNA Server/6000 for LANDP for AIX
- Microsoft SNA Server, Version 4.0 or IBM SecureWay Communications Server for Windows/NT for LANDP for Windows NT

The LANDP for OS/2 and Windows NT PPC server can support as many LANDP conversations as your configuration can hold. The limitations are set by the memory requirements of the workstations, the traffic on the communication line, and the supported communication management products.

The LANDP for AIX PPC server can support 25 LANDP conversations.

An application can handle more than one LANDP conversation. A LANDP conversation is an abstract entity, which is defined by:

- Workstation ID. Defined at customization.
- Conversation event handle. Dynamically set by applications.
- Process ID. Set by the operating system. This process ID allows applications that are running concurrently to use the same conversation ID.

The PPC server maps LANDP conversations into APPC conversations. These conversations are a means of serially sharing the SNA LU_6.2 sessions. These LANDP conversations can be mapped into any local LU, working concurrently against any partner LU.

A key characteristic of SNA LU_6.2 sessions is the fact that multiple concurrent sessions can be established between LUs. Applications handle LANDP conversations that allocate APPC conversations. However, these applications do not recognize the SNA LU session definitions, configuration, and management. The applications recognize LANDP conversations only.

The server gets requests from applications and issues calls, when necessary, to the APPC API of the supported communication management software which in turn sends appropriate data to the communication line.

Since the communication services are provided by the supported communication management products, the PPC server has to move within the boundaries of their communication profiles.

The PPC server provides the following features for applications:

- Distributed process communications with location transparency of the partner applications.
- The option of running on DOS, OS/2, Windows NT, or AIX, without loading extra APPC code, and to take advantage of services provided by the supported communication management products.
- A solution to interconnect several LANDP workgroups. If you use the PPC server, the connection is made through SNA LU_6.2 sessions and any DLC supported by the communication manager products.
- An interface that starts up and maintains a conversation, sends and receives data, and finishes the conversation with an APPC partner application using four functions (OP, CL, SD, and RD), plus an informative function (GA) and a query function (GS).

The advantage is that this interface has similar patterns to the ones used in the SNA server. Thus, LU_0 application developers are familiar with the application structure and can more easily adapt to the new LU_6.2 programming requirements.

- Session security and conversation security support. User ID and password can be directly managed by the application. Because of the multi-user characteristics of the RS/6000 systems that run IBM AIX, handling of session and conversation security in LANDP for AIX requires some additional activities. These are described in *AIX SNA Server/6000 Configuration Reference*, SC31-7014.
- A program exit for using your own data processing server (for instance, for cryptography purposes).

wide area communication, PPC, OS/2 AIX

For proper operation of the server, a consistent configuration has to be defined and the appropriate communications management product has to be installed in the appropriate directory. To achieve this objective, the communication management product should be installed or reinstalled using the correct configuration file.

See the *LANDP Installation and Customization* for more information about customizing the PPC server.

Conversation event handles for the PPC server

A conversation event handle must be provided in each communication function call. Since an application can have several concurrent, open conversations with the PPC server and more than one application can have open conversations with the PPC server at the same time, the conversation/event handle is required by the PPC server to identify the concurrent open conversations.

All the function calls in a conversation must use the same conversation event handle, which may be used for another conversation once the previous conversation is closed.

TCP/IP wide area communications server



The TCP/IP wide area communications server emulates both the SNA server and the PPC server and can be used in place of those servers to communicate over TCP/IP networks instead of SNA networks.

The TCP/IP wide area communications server supports:

- SNA LU2 sessions with a host computer using the TELNET tn3270 protocol
- SNA LU0, LU1 and LU2 sessions with a host computer using the MPTN (AnyNet®) protocol for SNA over TCP/IP
- PPC LU6.2 sessions with other computers using the MPTN (AnyNet®) protocol for SNA over TCP/IP

MQSeries Link server



The MQSeries Link server provides message queuing services to applications that use the Message Queuing Interface (MQI) provided by MQSeries for OS/2 and MQSeries for Windows NT. It extends the functions of the associated Queue Managers to the LANDP workgroup.

The applications that request services from the LANDP MQSeries Link server can run on any workstation within the LANDP workgroup. The workstation on which the LANDP MQSeries Link server is to run must have a MQSeries Server installed.

Figure 5 shows a possible configuration where A, B, C, and D are different machines. A LANDP MQSeries Link server must be in the same machine as an MQSeries Queue Manager (machine B in Figure 5). There can be more than one LANDP MQSeries Link server in a single workgroup (not in this example).

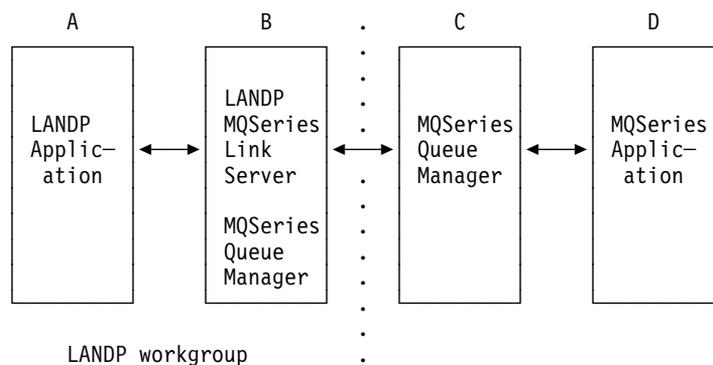


Figure 5. MQSeries Link server in a LANDP workgroup

Some applications may intend to use the default MQSeries Queue Manager. It is expected that one will be defined. Please refer to MQSeries documentation for more details.

Messages being sent or retrieved by the MQSeries Link server are limited in size by the requirements of the CPRB interfaces. For LANDP for OS/2 and Windows NT, the maximum message length is 57000 bytes. This can be restricted by the customization of the LANDP MQSeries Link server.

It is suggested that the MQSeries Queue Manager is defined to have a similar maximum message length to that specified for the LANDP MQSeries Link server. This can be achieved by use of the attribute `MaxMessageLength`. For more details, refer to the MQSeries documentation.

For security control, the MQSeries Link server relies on the services of the MQSeries Object Authority Manager (OAM). If access to MQSeries entities is to be controlled, the

access policies for the OAM have to be defined. This should be planned as part of the configuration of the MQSeries Queue Manager.

Data conversion of messages is enabled in the LANDP API with the GQ option EHCQG_CONVERT. When requested, conversion is attempted by the MQSeries Queue Manager.

The loader parameters can be set to create log files, as described in *LANDP Installation and Customization*. If log files are being created, some management of them is required.

When logging is switched on, a log file is created for each work station that requests a session with the MQSeries Link server. The log file is named as *XXhmmss.ddd* where:

XX is the workstation identifier of the LANDP client that initiated the session
hmmss is the session's start time in hours, minutes, seconds
ddd is the day of the year in the range 001 to 366.

Queues used by the MQSeries Link server should be predefined. Some sample queue definition files have been provided to aid application development. These files can be found as follows:

For Windows Nt in \EHC\EHCN500\SAMPLE
For OS/2 in \EHC\EHC0500\SAMPLE

Chapter 3. LANDP emulator support

This chapter describes the LANDP emulator support. It includes information about:

The LANDP 3270 emulator

This emulator provides the basic facilities of an IBM 3270 Information Display System terminal. It enables workstation operators to access host applications written for the 3270 environment.

The LANDP 3287 printer emulator

This emulator provides local printer support for host applications. It lets supported printers be used as if they were IBM 3287 Printers.

LANDP emulator APIs

Two sets of API functions are provided to access the 3270 and 3287 emulators from user-developed applications.

LANDP 3270 emulator



The LANDP 3270 emulator emulates the basic facilities of an IBM 3270 Information Display System terminal. Using the 3270 emulator, workstation operators can access host computer applications written for the 3270 environment.

Communication with the host computer can be started and controlled by:

- A workstation application.

A low-level and a high-level LANDP API are provided to enable application access to the 3270 emulator.

- The workstation operator.

A hotkey combination is used to switch to the 3270 emulation mode and back.

Each workstation using IBM 3270 Information Display System facilities must have the LANDP 3270 emulator installed. Up to five 3270 emulators can be installed in the same workstation. However, the LANDP for OS/2 and Windows NT 3270 emulators can only be used in a virtual DOS machine (VDM).

The 3270 emulator displays from 24 to 49 lines of 80 or 132 characters. (This may be limited by the characteristics of the display adapter that is installed in the workstation.) The last line of the workstation screen is used as the operator information area (OIA).

A LANDP workstation with the LANDP 3270 emulator installed is regarded by the host computer as an IBM 3274 Model 51C Control Unit, communicating through SNA/SDLC protocols.

The emulator uses a LU_2 session of the SNA server. When more than one 3270 emulator is installed in a workstation, the emulator with ID=1 uses the SNA session ID 51, the emulator with ID=2 uses the SNA session ID 52, and so on.

3270 terminal emulator, OS/2 DOS Windows NT

X.25 network communication with a host computer can be started by an application or by the workstation operator. The 3270 emulator can automatically connect to the X.25 network either when the workstation operator presses the hotkey to switch to the 3270 emulator, or through an application using the 3270 emulator application programming interface (low level) and establishing connection to the 3270 emulator keyboard.

At customization, files are built that contain the following information:

- Key assignment set
- Character set translation tables
- Display attributes

Unsupported facilities

The 3270 emulator does not support all the facilities of an IBM 3270 Information Display System terminal. The following list corresponds to the unsupported facilities related to the 3274 Control Unit, 3278 Display Station, and 3279 Color Display Station.

- IBM 3270 family displays with a screen size smaller than 24 by 80.
- Under Windows NT, 132 column screen size is not supported.
- Programmed symbols
- Magnetic stripe reader
- Light pen
- Entry assist
- Security keylock
- The following 3278 Display Station keys:
 - Indent
 - Click
 - Cursor blink

Assigning keys

Default key assignments and translation tables are provided. They can be used as is or changed as required. To provide support for national language special characters, modify the ASCII to EBCDIC and the EBCDIC to ASCII translation tables to send and receive the correct national language codes.

The 3270 emulator manages the keyboard. During customization, a specific 3270 emulator operation or an ASCII character string can be assigned to any key. The same function can be assigned to more than one key, and more than one function can be assigned to a key. For example, the *cursor left* and *delete* functions can be assigned to one key to produce a backspace that erases the preceding character. Unassigned keys default to ASCII characters.

Special keys that are managed by the DOS/V DBCS code generator for DBCS input cannot be assigned to 3270 emulator operations or ASCII character strings. Also, DBCS strings cannot be assigned to keys.

All national language keyboards are fully supported by loading the appropriate national language keyboard routine. The national language keyboard routine, or any other resident program that intercepts the keyboard interrupt, must be loaded before loading the LANDP program. For information on loading national language keyboard routines, refer to the IBM DOS manuals (see “Bibliography” on page 285).

Hotkeys to enter and exit the 3270 emulators are defined at customization. The default 3270 emulator hotkey values are Alt+F5 to enter and Alt+Esc to exit. If the LANDP for OS/2 3270 emulator is used, change the exit hotkey so that it does not conflict with the Alt+Esc key used by the OS/2 operating system.

If more than one 3270 emulator is defined for a workstation, one *enter* hotkey, and one or more *exit* hotkeys for each 3270 emulator must be defined during customization.

Hotkeys can be defined so that the workstation operator can start the next successive emulator as the previous one is ended. This is referred to as *chaining emulators*.

See *LANDP Installation and Customization* for more information about chaining emulators, configuring keys and hotkeys, and installing the 3270 emulator.

DBCS considerations

The LANDP for OS/2 and Windows NT 3270 emulators operate in MVDM mode only, and do not support DBCS mode operation.

For LANDP for DOS in DBCS mode, the 3270 emulator uses two screen positions to display each DBCS character. Also, the LANDP for DOS 3270 emulator uses the ASCII-EBCDIC translation server to convert DBCS characters between PC code and host code. This means that an ASCII-EBCDIC translation server and the code pages to be used by this server must be specified during customization.

The tables below show DBCS code page support:

Country (and code)	PC Code Page
Korea (82)	Combined code page 949 (SBCS 1088, DBCS 951)
Taiwan (88)	Combined code page 938 (SBCS 904, DBCS 927) Combined code page 950 (SBCS 1114, DBCS 947)
People's Republic of China (86)	Combined code page 1381 (SBCS 1115, DBCS 1380) Combined code page 1386 (SBCS 1115, DBCS 1380)

Country (and code)	Host Code Page
Korea (82)	Combined code page 933 (SBCS 833, DBCS 834)
Taiwan (88)	Combined code page 937 (SBCS 037, DBCS 835)
People's Republic of China (86)	Combined code page 935 (SBCS 836, DBCS 836) and Combined code page 1388 (SBCS 836, DBCS 4933)

Operations

Operating the 3270 emulator may involve the following tasks:

- Starting a LANDP 3270 emulator
- Displaying the emulation screen
- Connecting to the X.25 network
- The operator information area
- Status indicator codes
- Cryptography errors
- Host data stream errors
- Other error codes
- Common reasons for some status indicator codes
- Printing and saving panels
- Resetting the insert mode

Starting a LANDP 3270 emulator

In a DOS workstation, the customization process results in a `LOADER` statement for the 3270 emulator module being included in your `AUTOFBSS.BAT` file. When you run `AUTOFBSS` to start up `LANDP`, the emulator is loaded automatically and starts running at the same time. To see the 3270 emulator host session, press the appropriate hotkey. By default this is `Alt+F5`. For more information see “Assigning keys” on page 26.

For an OS/2 workstation, the customization process builds the `AUTOFBSS.CMD` file. This is used to start `LANDP`, and builds an `AUTOFBSS.BAT` file if there are any `LANDP` modules, such as `EMU3270`, that need to be run in a `VDM` session. You need to do the following to use a 3270 emulator on OS/2:

1. Type `AUTOFBSS` in an OS/2 window, from the `LANDP` directory.
This runs `AUTOFBSS.CMD` and starts the main part of `LANDP`.
2. Make sure you have set up the `VDM` session for `LANDP` for OS/2, as described in the chapter on setting up OS/2 workstations in *LANDP Installation and Customization*.
3. In a DOS box or DOS full-screen session, point to the same directory as the OS/2 window that is running `LANDP`, and type `AUTOFBSS` again.
This runs `AUTOFBSS.BAT`, which loads the `EMU3270` module.
4. To see the 3270 emulator host session, press the appropriate hotkey.

For a Windows NT workstation, the customization process builds the `AUTOFBSS.BAT` file. This is used to start `LANDP`, and builds an `AUTO_VDM.BAT` file if there are any `LANDP` modules, such as `EMU3270`, that need to be run in a `VDM` session. You need to do the following to use a 3270 emulator on Windows NT:

1. Type `AUTOFBSS` in a Windows NT window, from the `LANDP` directory.
This runs `AUTOFBSS.BAT` and starts the main part of `LANDP`.

2. Make sure you have set up the VDM session for LANDP for Windows NT, as described in the chapter on setting up Windows NT workstations in *LANDP Installation and Customization*.
3. In a DOS box or DOS full-screen session, point to the same directory as the Windows NT window that is running LANDP, and type `AUTO_VDM`
This runs `AUTO_VDM.BAT`, which loads the `EMU3270` module.
4. To see the 3270 emulator host session, press the appropriate hotkey.

Displaying the emulation screen

To see the 3270 emulation screen the workstation operator presses the *enter* hotkey. To hide the 3270 emulation screen, the workstation operator presses the *exit* hotkey. The emulator continues to run in the background. The hotkeys can be any key on the keyboard, pressed with the Alt key. They are defined at customization.

The enter hotkey is disabled when hardware interrupts are processed or when an IBM DOS facility call is in progress. If the 3270 emulator is running in a VDM, the emulator is disabled when a LANDP call is in progress. However, if you have pressed the hotkey, the 3270 emulator is activated once the process finishes and without pressing the hotkey again. The delay involved is usually not noticeable by the workstation operator.

If the exit hotkey is pressed while data is being transmitted, the 3270 emulator does not end until the transmission is finished.

Connecting to the X.25 network

Communication with the host computer can be started either by an application or by the workstation operator. The 3270 emulator can automatically connect to the X.25 network either when the workstation operator presses the hotkey to switch to the 3270 emulator, or through an application using the 3270 emulator application programming interface (low level) and establishing connection to the 3270 emulator keyboard.

The 3270 emulator releases the X.25 network when one of the following actions takes place:

- The workstation operator presses the *Release* key of the 3270 emulation keyboard. This key is defined at customization. For an SSCP to LU session, press the key once; for an LU to LU session, press the key twice.
- The workstation operator uses the end connection command (EC) of the operator interface. For information on the operator interface, refer to "Operator interface" on page 182.

Operator information area

The last line of the workstation screen is used as the operator information area (OIA). The corresponding fields are described in the following tables.

Table 3. SBCS operator information

Column	SBCS operator information	
1	4	System available
2	B	Secondary LU: B SNA protocol
3		Primary LU:
	⊙	System operator
	?	Unowned
	□	My job
10	X	Inhibited keyboard
12	M	Machine check
	*	Terminal wait
	?	Retry
	S	System wait
	f	Minus function
	⊙	Operation action not valid
	#	Numeric field
	√	Communication check
	P	Program check
	D	Device not working
	O	Operator communication check
14 – 16	SIC	Status indicator code (see “Status indicator codes” on page 34)
20	√	Communication check reminder
21 – 24		X.25 cause and diagnostic (CLEAR/RESET packet)
26	#	Cursor in numeric field
35	↑	Shift key pressed
36	C	Ctrl key pressed
37	A	Alt key pressed
39	⇄	Caps locked
41	N	Numeric pad locked
43	â	Insert on
49 – 50		Workstation ID
52		Session short name
54 – 61		Session long name (8 characters)
70 – 71		Cursor row
72	/	Separator
73 – 74		Cursor column (80 column screen)
73 – 75		Cursor column (132 column screen)

<i>Table 4. DBCS operator information</i>		
Column	DBCS operator information	
1	4	System available
2	B	Secondary LU: B SNA protocol
3		Primary LU:
	<input checked="" type="checkbox"/>	System operator
	?	Unowned
	<input type="checkbox"/>	My job
5 – 8		Keyboard status, defined by DOS/V FEP
9	↑	In shift status
10 – 13		Keyboard status, defined by DOS/V FEP
17		DBCS translation mode, defined by DOS/V FEP
20	X	Inhibited keyboard
22	M	Machine check
	⊛	Terminal wait
	?	Retry
	S	System wait
	f	Minus function
	<input checked="" type="checkbox"/>	Operation action not valid
	#	Numeric field
	X	Communication check
	P	Program check
	D	Device not working
	O	Operator communication check
24 – 26	SIC	Status indicator code (see "Status indicator codes" on page 34)
30	X	Communication check reminder
31 – 34		X.25 cause and diagnostic (from CLEAR/RESET packet)
36	#	Cursor in numeric field
55	C	Ctrl key pressed
56	A	Alt key pressed
57	⚡	Caps locked
58	N	Numeric lock
59	^	Insert on
61 – 62		Workstation ID
64		Session short name
66 – 73		Session long name
75 – 76		Cursor row
77	/	Separator
78 – 79		Cursor column (80 column screen)
78 – 80		Cursor column (132 column screen)

3270 terminal emulator, OS/2 DOS Windows NT

Note: Columns 19 to 57 are also used by the DOS/V code generator (FEP) when in DBCS conversion mode. The status indicator displayed on the OIA is overwritten by the FEP and cannot be changed by the LANDP 3270 emulator even if the status changes.

The 3270 emulator locks the keyboard when finding a situation not explicitly handled by its internal logic. An *X X nnn* indicator is displayed, where *nnn* is the status indicator code.

Note that the error can be due to a previous malfunction; then, the status indicator code does not correspond to the actual problem. In any case, the correct action is to check the host computer and communication equipment, refer to the supervisor function trace and the LANDP manuals, and check for a hardware or configuration error.

Status indicator codes

This explains the three digit Status Indicator Codes that LANDP's 3270 emulator places in the Operator Information Area (OIA) .

The codes fall into several ranges:#

- 3xx - Cryptography errors
- 4xx - Host data stream errors
- 6xx - Release errors
- 7xx - Connect errors
- 8xx - Read Host errors
- 9xx - Send Host errors

Cryptography errors

Codes in the 3xx range indicate cryptography errors and are individually defined, as follows:

- 394** A hardware or software error occurred while generating the encryption key or initial value.
- 395** A hardware or software error occurred while sending the encryption initial value to the host.
- 396** A hardware or software error occurred while verifying the encryption data sent from the host.
- 397** A hardware or software error occurred while sending the encryption verification response to the host.
- 398** A hardware or software error occurred while decrypting data sent from the host.
- 399** A hardware or software error occurred while encrypting data to be sent to the host.

Host data stream errors

Codes in the 4xx range indicate host data stream errors and are individually defined, as follows:

- 401** Host data stream contained an invalid command code, an invalid data stream order sequence, or invalid structured field.
- 402** Host data stream contained an invalid address in a Set Buffer Address Erase Unprotected to Address or Repeat to Address data stream order sequence.
- 403** Host data stream contained a Read Modified, Read Modified All, Read Buffer or Erase All Unprotected command which was followed by subsequent data.
- 498** An unexpected negative response was received from the host. This probably indicates a host protocol or communications error.

To further diagnose the exact reason for the failure, examine the contents of the host datastream to determine the cause of the error.

Other error codes

Codes of 6xx and above are generated when a SNA function ends with a return code which indicates that an error has occurred. The first digit indicates the SNA function which failed (6=Release, 7=Connect, 8=Read Host and 9=Send Host), and the last two digits are derived from the last three bits of each character of the LANDP return code received from the SNA server, the LANDP workgroup server or the router.

For example, a timeout on a send causes an L8 return code to be returned. 'L8' in ASCII is hex '4C38', and so the OIA error code is 940, which is derived as follows:

- 9 - means 'Send'
- 4 - from 'L'. In ASCII this is hex 4C, binary 0100 1100;
The last three bits are 100, which is decimal 4.
- 0 - from '8'. In ASCII this is hex 38, binary 0011 1000;
The last three bits are 000, which is decimal 0.

Each digit of the xx can therefore only be in the range 0-7 and can be derived from several alternative return code characters:

OIA	:	x'30'	-	x'38'	-	x'40'	-	x'48'	-	x'50'	-	x'58'
digit	:	x'37'	-	x'3F'	-	x'47'	-	x'4F'	-	x'57'	-	x'5F'
-----	:	-----	-	-----	-	-----	-	-----	-	-----	-	-----
0	:	0		8				H		P		X
1	:	1		9		A		I		Q		Y
2	:	2				B		J		R		Z
3	:	3				C		K		S		
4	:	4				D		L		T		
5	:	5				E		M		U		
6	:	6				F		N		V		
7	:	7				G		O		W		

In practice, only certain codes are used. This table gives the OIA error codes for return codes in the ranges I1-I4, L0-LM, P0-PF, PZ and U1-U9.

```
OIA : LANDP
code : return codes
---- : -----
x00 : P0, P8
x01 : P1, P9, PA
x02 : P2, PB, PZ
x03 : P3, PC
x04 : P4, PD
x05 : P5, PE
x06 : P6, PF
x07 : P7
x11 : I1
x12 : I2
x13 : I3
x14 : I4
x40 : L0, L8, LH
x41 : L1, L9, LA, LI
x42 : L2, LB, LJ
x43 : L3, LC, LK
x44 : L4, LD, LL
x45 : L5, LE, LM
x46 : L6, LF
x47 : L7, LG
x50 : U8
x51 : U1, U9
x52 : U2
x53 : U3
x54 : U4
x55 : U5
x56 : U6
x57 : U7
```

To further diagnose the exact reason for the failure, examine the LANDP function trace to determine the failing SNA function and its return code.

Status indicator codes

The following tables list the common reasons for some of the status indicator codes that may be generated.

Status indicator codes

701	LUA definition error <ul style="list-style-type: none"> • DOS: SNA.CFG and LAN.CFG mismatch • OS/2: Communications Server LUA definitions mismatch • AIX: POSTSNA.CFG and LVA.CFG files do not agree
703	No communication buffers. Configuration error.
705	LANDP internal error. For example, (OS/2) server-managed cryptography has been requested but the Communications Server exits are missing or invalid.
712	Modem powered off, or communication adapter or cabling hardware problem.
713	No X.25 circuit available.
740	LAN time-out.
741	Hardware failure or configuration error (LAN.CFG or COM.CFG).
742	The gateway is not accessible for one of the following reasons: <ul style="list-style-type: none"> • Configuration error • The gateway is powered off • LANDP not loaded at the gateway
743	Session not defined, configuration error.
744	Irrecoverable gateway error.
745	Server not loaded. Configuration error.
746	Irrecoverable workstation error.
747	LAN adapter not present or support software not loaded.
750	<ul style="list-style-type: none"> • DOS: No DLC loaded, configuration problem • OS/2: Communications Server not started or link not started • AIX: SNA/6000 not started or link not started • AS/400: communication problem
751	Pending contact. Communications error.
754	Communication adapter hardware error.
756	Communication adapter software microcode error.

Status indicator codes

801	LUA definition error or LANDP internal error <ul style="list-style-type: none"> • DOS: SNA.CFG and LAN.CFG mismatch • OS/2: Communications Server LUA definitions mismatch • AIX: POSTSNA.CFG and LVA.CFG files do not agree
803	An attempt has been made to read a message from the host when there were no pending messages, or to send a message to the host when the host was not ready to receive messages. If a program is running that uses either the LANDP 3270 emulator API or the LANDP 3270 emulator high-level language API (HLLAPI) to communicate with the host, check the logic of this program to ensure it is not trying to communicate with the host at the wrong time.
811	No session with the host.
812	Modem powered off, or communication adapter or cabling hardware problem.
840	LAN time-out.
841	Hardware failure or configuration error (LAN.CFG or COM.CFG).
842	The gateway is not accessible due to one of the following reasons: <ul style="list-style-type: none"> • Configuration error • The gateway is powered off • LANDP not loaded at the gateway
843	Session not defined, configuration error.
844	Irrecoverable gateway error.
845	Server not loaded. Configuration error.
846	Irrecoverable workstation error.
847	LAN adapter not present or support software not loaded.
850	<ul style="list-style-type: none"> • DOS: No DLC loaded, configuration problem • OS/2: Communications Server not started or link not started • AIX: SNA/6000 not started or link not started
851	Pending contact. Communications error.
854	Communication adapter hardware error.

Status indicator codes

856	Communication adapter software microcode error.
901	LUA definition error <ul style="list-style-type: none">• DOS: SNA.CFG and LAN.CFG mismatch• OS/2: Communications Server LUA definitions mismatch• AIX: POSTSNA.CFG and LVA.CFG files do not agree
903	No communication buffers. Configuration error.
907	Communication server protocol error.
911	No session.
912	Modem powered off, or communication adapter or cabling hardware problem.
940	LAN time-out.
941	Hardware failure or configuration error (LAN.CFG or COM.CFG).
942	The gateway is not accessible due to one of the following reasons: <ul style="list-style-type: none">• Configuration error• The gateway is powered off• LANDP not loaded at the gateway
943	Session not defined, configuration error.
944	Irrecoverable gateway error.
945	Server not loaded.
946	Irrecoverable workstation error.
947	LAN adapter not present or support software not loaded.
950	<ul style="list-style-type: none">• DOS: No DLC loaded, configuration problem• OS/2: Communications Server not started or link not started• AIX: SNA/6000 not started or link not started
951	Pending contact.
954	Communication adapter hardware error.
956	Communication adapter software microcode error.

Note that column 12 in the OIA shows the reason why the keyboard is inhibited. When the keyboard locks because of either machine check (M), program check (P), or communication check (√), columns 14 through 16 show the status indicator code corresponding to those values displayed in column 12. If you are working in DBCS mode, column 22 shows the reason for the keyboard being inhibited and columns 24 through 26 show the status indicator code corresponding to the values displayed in column 22. Communication check is represented as (X). The other reasons for the keyboard being inhibited are represented the same way as in SBCS mode. If the keyboard is inhibited in DBCS conversion mode, keyboard status areas 1 and 2 and conversion information (columns 19 through 57) are not displayed.

Printing and saving panels

The 3270 emulator sends the current panel to the printer or saves it on disk. Printing can be started either by the host computer or by the workstation operator. The host computer can print the current panel using a host computer started local copy; note that the host computer cannot save the current panel.

The workstation operator can print the current panel pressing the *PrtSc* and *Left Shift* keys. Only the workstation operator can save the current panel on disk. If the workstation operator presses *PrtSc* and *Right Shift* keys, the panel is appended to the FBSS3270.SCN file. When an error occurs, the keyboard locks and a disk error indicator appears in the OIA.

Regardless of how printing is started, the standard BIOS print screen routine is called. If a nonstandard printer is attached to the workstation and its corresponding print screen routine replaces the BIOS routine, printing is also supported.

Resetting the insert mode

If you are working in DBCS mode, you can reset the insert mode by pressing the *Reset* key and the *Enter* key.

Application programming interfaces for the IBM 3270 environment



LANDP provides two ways to interact with a host program written for the IBM 3270 Information Display System:

- An operator can directly use the LANDP 3270 emulator to interface to the host system. In this case the workstation appears as a 3270 terminal.
- A client application can use the functions of one of the LANDP 3270 emulator APIs:
 - LANDP 3270 Emulator High-Level Language API (HLLAPI)
 - LANDP 3270 Emulator API

Using the API functions, the client application can:

- Send “keystrokes” representing commands or messages to the host
- Read the data sent by the host
- Read the information intended for the operator information area of a 3270 display station.

At the same time, the client application can also, for example:

- Access information stored on a local fixed disk, or request services from the shared-file server
- Display information for the workstation operator

LANDP 3270 emulator high-level language API (HLLAPI)

The LANDP HLLAPI provides a way to implement a programmed operator. This API is accessible from the high-level programming languages supported by LANDP.

File transfer programs can be used under application control.

This interface is largely compatible with the EHLLAPI interface, provided by IBM Communications Server for OS/2 Warp.

LANDP 3270 emulator API

The LANDP 3270 API functions enable client applications to communication with host applications.

Refer to *LANDP Programming Reference*, for more information about the LANDP 3270 APIs.

LANDP 3287 printer emulator



The LANDP 3287 printer emulator is an optional internal application that provides a means to print host data on a printer attached to a LANDP workstation. This emulator is loaded into memory when LANDP is initialized, and runs in the background. The 3287 printer emulator automatically checks the availability of workstation-attached printers.

Only one 3287 printer emulator can be installed in a workstation or a virtual DOS machine (VDM).

The LANDP for OS/2 and Windows NT 3287 emulator only operates in a VDM and does not support DBCS mode operation.

3287 printer emulator, OS/2 DOS Windows NT

The 3287 emulator supports a variety of print modes, including:

- Personal printer data stream (PPDS)
- PCL4 emulation
- PCL5 emulation
- HP emulation

This means that the IBM Proprinter®, IBM 4019, IBM 4029, IBM 4039, or any functionally equivalent printer attached to a parallel printer port is supported.

If your printer differs from the supported printers in the control characters used to reset it before a new listing, specify the characters when customizing the attribute table for the emulator.

Also, all the printers supported by the financial printer server that are connected to a serial port are supported. However, the financial printer server must be installed in the same workstation to support these printers. See “Financial printer server” on page 68 for more information.

To manage printers attached to parallel printer ports, the 3287 printer emulator can be used with the operator interface and the local resource manager server. However, these interfaces cannot be used for printers attached to a serial port or when the 3287 emulator is installed in a VDM.

The LANDP 3287 printer emulator has the following characteristics:

- It serves (through the EMU3287 version) up to three supported printers attached to parallel ports or redirected by the operating system to printers attached to serial ports or LAN printers.
- It serves (through the EMU3287R version) one printer redirected to the LANDP financial printer server.
Note: The host computer sessions that use the 3287 printer emulator must serve only one type of printer.
- It supports local printing, and the printer manager server enables printers to be shared among members of the workgroup. For more information see “Printer manager server” on page 74.
- It uses the SNA server and LU_1 sessions to communicate with one or more host computers. Up to five LU_1 sessions, numbered from 1 to 5, can be defined. The number corresponds to the priority for processing the print requests.
- It provides three exits to allow a user server to change buffer data to be printed, to know when a listing is ended, and to change unsupported SCS control characters into supported control characters.

The workstation operator can dynamically change printer assignments if the operator interface is installed. (See “Operator interface” on page 182 for more information.) Printer assignments can also be changed using an API function.

See *LANDP Installation and Customization* for additional information about configuring LU_1 parameters on the host.

The LANDP program provides an EBCDIC to ASCII translation table that translates International Host Code Page 500 (EBCDIC) to PC Code Page 850 (ASCII). This table can be modified during customization.

All printers using a 3287 printer emulator use the same EBCDIC to ASCII translation table. However, a different EBCDIC to ASCII translation table can be defined for each 3287 printer emulator.

DBCS considerations

When working in DBCS mode, the IBM 5575 Printers or IBM 5577 Printers for Korea and Taiwan, and the ESC/P printer for the People's Republic of China, have the same capabilities and limitations as a Proprinter.

The EBCDIC to ASCII translation table is not used when working in DBCS mode. In this case, the services of the "ASCII-EBCDIC translation server" on page 240 are used.

For more information about the API functions and other programming considerations, refer to *LANDP Programming Reference*.

See *LANDP Installation and Customization* for additional information about configuring LU_1 parameters on the host.

Operations

The 3287 printer emulator does not provide an automatic way to detect paper jams at the printer it serves. When a paper jam is detected, the workstation operator should either:

- Press the *not on line* key, clear the paper jam, and press the *on line* key (this choice means that the listing continues printing), **or**
- Press the *power off* key, clear the paper jam, ask the host computer operator to resend the listing, and press the *power on* key. This choice means that the listing restarts printing.

Chapter 4. Remote change management services



The RCMS feature uses the Tivoli NetView®Distribution Manager (Tivoli NetView DM) to manage LANDP software and data sources using a host system.

RCMS, together with Tivoli NetView DM, is used to:

- Keep application libraries up-to-date
- Safely distribute production data
- Track resources in the host computer and in the workstations
- Maintain the system software
- Manage problems with the remote node systems

The commands used in the interaction between RCMS and Tivoli NetView DM include requests for RCMS to:

- Send resources from the workstation to the host computer
- Receive resources at the workstation
- Receive a command list to be performed at the workstation
- Send a host computer operator message to the workstation operator

The combination of RCMS and NetView DM:

- Allows a host to manage application libraries and other files for a workgroup by using a shared DOS directory server or other LAN services to share directories. For more information on the shared DOS directory server, refer to “Shared DOS directory services” on page 137.
- Automatically converts files between the host EBCDIC and workstation ASCII format.
- Allows SNA session level cryptography using the LANDP for OS/2 SNA server with server-controlled cryptography.

The main features of RCMS:

- RCMS is efficient. The performance of the workstation with RCMS installed is not significantly affected.
- RCMS interacts with the SNA server, therefore at least one SNA server must be installed in the LANDP workgroup. The communication between Tivoli NetView DM and RCMS uses SNA LU_0 protocols; SDLC, token-ring, and X.25 data link control are supported.

The communication with the host computer is monitored by RCMS, but managed by the SNA server. RCMS makes its communication requests to the SNA server, and the SNA server performs the transmission.

RCMS first sets up an LU-LU session (session ID SNA33) with the host computer. It then polls the host computer for a command. Commands are processed and the wait loop is entered again.

RCMS, OS/2 DOS Windows NT

- The interval where RCMS receives control to process pending requests is determined at initialization.
- RCMS can use the system manager server to manage the log files.

Tivoli NetView DM is a host computer application that manages and controls distribution of software and data resources in a distributed data processing system. It keeps backup copies of node resources, tracks the status of the resources, and keeps the resources up-to-date.

RCMS supports Tivoli NetView Distribution Manager up to Version 1.6. More information about Tivoli NetView DM can be found in the following publications:

- *Tivoli NetView Distribution Manager: General Information*, GH19-6792
- *Tivoli NetView Distribution Manager: Planning*, SH19-6589
- *Tivoli NetView Distribution Manager: User's Guide*, SH19-6795

The interaction between RCMS and Tivoli NetView DM is shown in Figure 6 on page 45. In the figure, workstation AA has the SNA server installed. This workstation provides communication to the host computer for all the workstations in the LANDP workgroup.

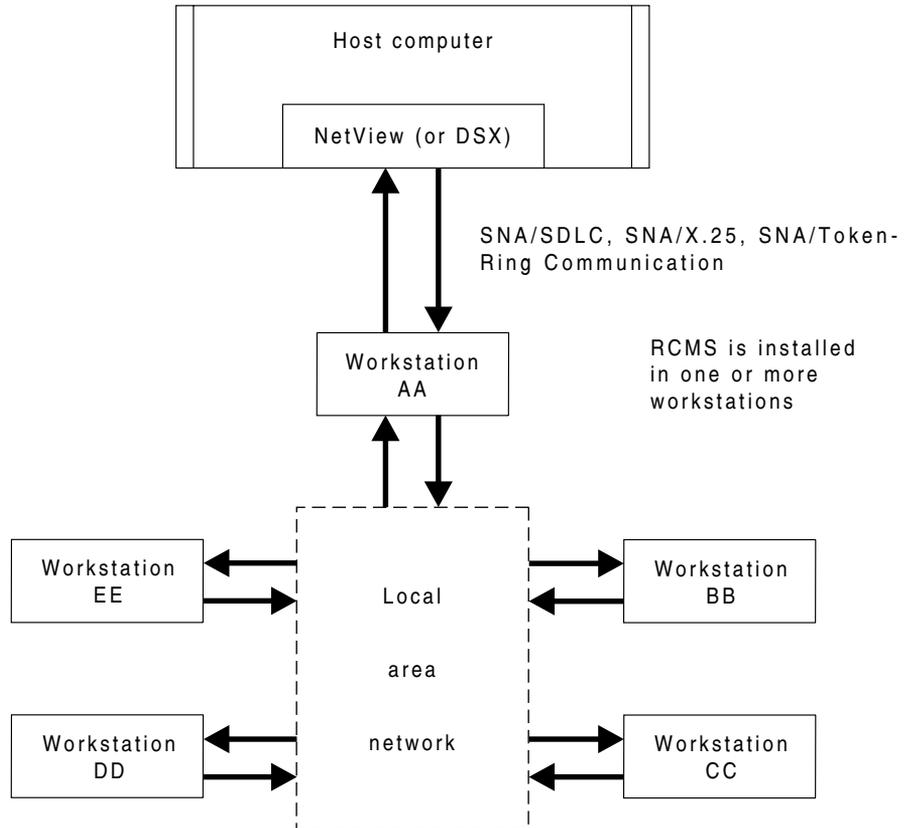


Figure 6. RCMS/NetView DM

RCMS supports file distribution in both binary and text mode. Binary mode is used for non-EBCDIC files distribution (such as .EXE files or text files stored in ASCII format in the NetView DM repository). Text mode is used to receive EBCDIC data as ASCII data or ASCII data as EBCDIC data.

EBCDIC data consists of variable length records delimited by carriage return (CR) or line feed (LF). When the host computer sends an EBCDIC file to a workstation, RCMS converts it to an ASCII file. When the workstation sends an ASCII file to the host, RCMS converts it to an EBCDIC file.

If operating in DBCS mode, the translation mode is specified during customization:

- Standard ASCII-EBCDIC and EBCDIC-ASCII translation.
- ASCII-EBCDIC translation with ASCII SI/SO characters changed to EBCDIC SI/SO characters, and EBCDIC-ASCII translation with EBCDIC SI/SO characters changed to ASCII SI/SO characters.
- Standard ASCII-EBCDIC translation, and EBCDIC-ASCII translation with EBCDIC SI/SO characters changed to blanks.

RCMS, identifying files, OS/2 DOS Windows NT

When the host computer retrieves a file from a workstation, the corresponding Tivoli NetView DM RETRIEVE command requires a file type specification. This information is provided through a parameter, using the Tivoli NetView DM PREPARE PLANS utility program in the host computer. RCMS uses the file type information to determine whether the file needs to be converted.

SNA session level cryptography requires a workstation with a cryptographic server installed. A cryptographic server is not supplied with the LANDP program. To write a cryptographic server, refer to *LANDP Programming Reference*.

A facility provided with NetView DM retains phases in hold state. This feature lets a workstation with RCMS installed request NetView DM to release a phase that is being held.

NetView DM processes the release request sent by RCMS and sends a response to RCMS. Once NetView DM accepts the request, it schedules the phase to be run as soon as possible.

For more information see "Requesting phase remote release" on page 48.

Identifying files

Tivoli NetView DM identifies a resource by its name, node origin, and file control name. The resource name of a Point of Sale Controller (POSC) node consists of one, two, or three qualifiers. The first of these qualifiers is called an *alias*. RCMS matches the NetView DM resource names with workstation FAT paths or file names by reading the contents of the file RCMS.LNF. RCMS.LNF contains a list of aliases associated with paths or file names. All entries in RCMS.LNF are ASCII characters.

Bytes	Description
1 - 8	<i>Alias</i> . These entries must have unique names. The names comprise the uppercase letters (A-Z), numbers (0-9), and the special characters #, \$, @, &, %, and _.
9 - 72	Path name or complete file name. The names comprise the uppercase letters (A-Z), numbers (0-9), and the special characters #, \$, @, &, %, and _.
73	Blank.

The following examples illustrate the process, depending on the number of qualifiers:

Tivoli NetView DM qualifier	AAA
RCMS.LNF entry	AAA C:\TEST\TEST.DAT
Result	The file C:\TEST\TEST.DAT is accessed

If the resource name has one qualifier, the string specified in RCMS.LNF can be the path or file name. The qualifier AAA is an alias for the file specification.

<i>Table 7. RCMS.LNF example: two qualifiers</i>	
Tivoli NetView DM qualifier	BBB.TEST
RCMS.LNF entry	BBB C:\UTIL\
Result	The file C:\UTIL\TEST is accessed

If the resource name has two qualifiers, the second one is added at the end of the string specified in RCMS.LNF. The result of this addition is a file name specification without a file extension. The qualifier BBB is an alias for the disk drive plus the path name of the file.

<i>Table 8. RCMS.LNF example: three qualifiers</i>	
Tivoli NetView DM qualifier	CCC.TEST.DAT
RCMS.LNF entry	CCC C:\
Result	The file C:\TEST.DAT is accessed

If the resource name has three qualifiers, the string specified in RCMS.LNF is the path, the second qualifier is the file name, and the third qualifier is the file extension. The qualifier CCC is an alias for the disk drive of the file.

Creating RCMS.LNF

You create the RCMS.LNF file, and specify its initial contents, at customization. At run time, it must reside in the system file directory, which is also specified during customization. You can edit the RCMS.LNF using LNFEDIT.EXE. Every line should match the record format formerly described.

Modifying RCMS.LNF

The LNFEDIT.EXE program is used to add, delete, or change, the contents of the RCMS.LNF file. This program runs only under Windows NT in a DOS window, IBM DOS, and IBM OS/2 VDM. You can run this program in every workstation where you want to modify the RCMS.LNF contents, or you can run it in one of these workstations and then copy the resulting RCMS.LNF to the other workstations. The LNFEDIT.EXE program and the LNF.PAN file are in the distribution diskettes and must be placed in the same directory as RCMS.LNF.

To run the LNFEDIT.EXE program, enter:

```
LNFEDIT
```

RCMS, phase remote release, OS/2 DOS Windows NT

Tivoli NetView DM commands can be used to replace RCMS.LNF in a workstation. The commands used are RETRIEVE and REPLACE, as follows:

- With the RETRIEVE command, the host computer obtains a copy of the new RCMS.LNF file from the workstation where the file has been modified.
- With the REPLACE command, the new RCMS.LNF file is sent from the host computer to the workstations where the file is to be placed.

Notes:

1. The contents of RCMS.LNF may be destroyed, if they are updated with an editor other than LNFEDIT.
2. If you are using a double-byte character set (DBCS) version of OS/2, run the LANDP for OS/2 LNFEDIT program on a PC DOS VDM (unless Chinese language support has been specified—in which case, run it on a DOS/V VDM).

Translating data

When you retrieve a file on a workstation from the host computer, the corresponding Tivoli NetView DM RETRIEVE DATASET command requires a file type specification. RCMS uses the file type information to determine whether the file needs to be translated to EBCDIC. If you send a non-binary dataset to a workstation using the SEND DATASET NetView DM command, the file is translated back to ASCII.

Requesting phase remote release

A facility provided with Tivoli NetView DM Version 1.2 and subsequent releases enables you to have phases in hold state. Thus, a workstation with RCMS installed can request Tivoli NetView DM to release an Tivoli NetView DM *held phase*.

Tivoli NetView DM processes the release request sent by RCMS and sends a response to RCMS. Once Tivoli NetView DM accepts the request, it schedules the phase to be run as soon as possible.

The information about the phases to be released is kept in an internal file called RCMSPLAN.TXT. To create this file:

1. Write an input file called RCMSPLAN.NEW using any ASCII editor. This file must contain the phase specifications. Refer to Table 9 on page 51 for information about the specific record format.
2. Place RCMSPLAN.NEW in the process path specified at customization.
3. Run EHCPLN at the workstation where RCMS is installed.

After creating the RCMSPLAN.TXT file, you may want to modify the phase specifications. To do so:

1. Write an input file called RCMSPLAN.xxx using any ASCII editor. xxx must be:

- NEW, if you are going to run EHCPLN at the workstation where RCMS is installed.
- The workstation identification (WS_ID), specified at customization, of the workstation where the file will be placed, if you are going to run EHCPLN at a workstation where RCMS is not installed.

This file must contain the phase specifications. Refer to Table 9 on page 51 for information about the specific record format.

2. If you are going to run EHCPLN locally, place RCMSPLAN.NEW on the process path specified at customization. If you are going to run EHCPLN remotely, place RCMSPLAN.WS_ID on any directory in the remote workstation. Make sure, though, that the process path in the workstation where RCMS is installed is accessible.
3. Run EHCPLN. You can run it locally or remotely.

Running EHCPLN locally

```
EHCPLN [/O:operation]
```

Running EHCPLN remotely

```
EHCPLN [/O:operation] /W:dest_ws /D:dest_path [/S:sou_path]
```

where:

operation Specifies the kind of modification you wish to apply to RCMSPLAN.TXT. This parameter can take the following values:

- R If the file RCMSPLAN.TXT does not exist, it is created based on the contents of RCMSPLAN.NEW. If RCMSPLAN.TXT already exists, its contents are copied to RCMSPLAN.OLD to save the contents from previous operations. RCMSPLAN.TXT contents are then replaced by the contents of RCMSPLAN.NEW. This operation is supported when running EHCPLN locally only.
- A Adds the new phases specified in the input file to RCMSPLAN.TXT.
- U Updates phase specifications in RCMSPLAN.TXT based on the contents of the input file. The EHCPLN utility program updates specifications for the phases that meet the following criteria:
 - The plan name on the input file and RCMSPLAN.TXT is the same.
 - The phase name on the input file and RCMSPLAN.TXT is the same.

dest_ws Specifies the ID of the workstation where the target RCMS for this operation is running. This parameter is required for remote operation only.

dest_path Specifies the path where RCMSPLAN.TXT can be found. This parameter is required for remote operation only.

sou_path Specifies the path where RCMSPLAN.WS_ID can be found. This parameter is required for remote operation only. If you do not specify this parameter, the current directory is used.

RCMS, phase remote release, OS/2 DOS Windows NT

The following example shows the usage of the EHCPLN program:

```
EHCPLN /O:U /W:BB /D:T:\RCMSOS2\PROCESS
```

If you enter this command at workstation AA, the EHCPLN program reads the new specifications from RCMSPLAN.AA, which resides on the current directory (no /s: parameter has been specified), and updates RCMSPLAN.TXT.

RCMSPLAN.TXT resides on T:\RCMSOS2\PROCESS on workstation BB. However, if you leave out the destination path, /D:T:\RCMSOS2\PROCESS, the command will fail because RCMSPLAN.TXT is not accessible.

After running EHCPLN, RCMS gets the necessary information for releasing phases from the RCMSPLAN.TXT file. This file has been updated with the information contained in RCMSPLAN.NEW/RCMSPLAN.WS_ID. The RCMSPLAN.TXT specifications previous to running EHCPLN are stored in RCMSPLAN.OLD.

Any modification without notifying RCMS, through the EHCPLN utility program, will not take effect.

If you want to modify RCMSPLAN in the host:

1. RETRIEVE the file from the workstation.
2. Make the necessary changes to the file.
3. SEND it back to the workstation.
4. Run the EHCPLN utility through an INITIATE CLIST command.

The advantage of remote phase release is that while the workstation is not ready to process the commands, Tivoli NetView DM does not waste time attempting to communicate with this workstation. After RCMS has issued a request for remote phase release, Tivoli NetView DM sends a response to RCMS. This response is used to update the status field in the file RCMSPLAN.TXT.

The record format for the files RCMSPLAN.NEW, RCMSPLAN.WS_ID, and RCMSPLAN.TXT is the same. However, for update operations, you must fill with # fields in RCMSPLAN.NEW and RCMSPLAN.WS_ID which you do not wish to update in RCMSPLAN.TXT.

Table 9. RCMSPLAN files record format

Offset	Length	Content
0	1	Workstation processing command flag: – Ignore this record + Process this record * Comment line.
1	8	Host computer application name for Tivoli NetView DM.
9	1	Status: W Waiting for the start date and time (if new record). R Released. Tivoli NetView DM has received a request to release the phase and has responded positively. H Held. Either phase held by condition, or destination held by the user. E Error. Tivoli NetView DM reported an error when an attempt was made to release this phase. D RCMS tried to release again a phase that was already done. Tivoli NetView DM has already completed this phase. N Phase not found.
10	1	Blank.
11	8	The name of the plan.
19	1	Blank.
20	8	The name of the phase.
28	1	Blank.
29	8	Start date. The earliest date when the phase can be released. If the current date is on or after this date, RCMS can request the release of the phase at a time within the time window. The date format is YY/MM/DD.
37	1	Blank.
38	8	Start time. The start time for the time window during which the phase can be released by RCMS. The format is hh:mm:ss.
46	1	Blank.
47	8	End time. The format is the same as the start time format.

The following example shows:

- The scenario where the EHCPLN program will run to update the phase specifications in RCMSPLAN.TXT:
 - RCMS is loaded on workstation S1
 - The process path is C:\RCMS0S2
 - The EHCPLN program will run on workstation S2, which shares the C: drive on workstation S1 as R:

RCMS, phase remote release, OS/2 DOS Windows NT

- Input file RCMSPLAN.S2 is stored in the process path on workstation S1.
 - The contents of C:\RCMSOS2\RCMSPLAN.TXT on workstation S1:
+NDM14 R CID0169 PHPOSC 94/02/01 00:00:00 24:00:00
+NDM14 R CID0170 PHPOSC 94/02/01 00:00:00 24:00:00
+NDM14 R CID0171 PHPOSC 94/02/01 00:00:00 24:00:00
+NDM14 R CID0172 PHPOSC 94/02/01 00:00:00 24:00:00
 - The contents of the input file C:\RCMSOS2\RCMSPLAN.S2 on workstation S1:
+NDM14 W CID0170 PHPOSC 94/02/02 ##:##:## 14:00:00
+NDM14 # CID0171 PHPOSC ##/##/## 15:30:00 ##:##:##
- The first record will update all the fields apart from the start time. The second record will update the start time.

The statement used to run the EHCPLN program in order to update RCMSPLAN.TXT with the contents of RCMSPLAN.S2 is:

```
EHCPLN /O:U /W:S1 /D:R:\RCMSOS2 /S:R:\RCMSOS2
```

The final contents of C:\RCMSOS2\RCMSPLAN.TXT on workstation S1 after running the EHCPLN program are:

```
+NDM14 R CID0169 PHPOSC 94/02/01 00:00:00 24:00:00  
+NDM14 W CID0170 PHPOSC 94/02/02 00:00:00 14:00:00  
+NDM14 R CID0171 PHPOSC 94/02/01 15:30:00 24:00:00  
+NDM14 R CID0172 PHPOSC 94/02/01 00:00:00 24:00:00
```

If the operation ends successfully, a message is displayed:

```
OK RC1 = n
```

where *n* can represent one of the following according to the /O: parameter option you choose. If you chose:

```
R n is the number of lines in new file RCMSPLAN.TXT  
A n is the number of lines appended to RCMSPLAN.TXT  
U n is the number of lines updated in RCMSPLAN.TXT
```

Tivoli NetView DM commands

Software and data resources can be transmitted from the host computer to the distributed system. The resources that can be transmitted are CLISTs, datasets, dumps, messages, panels, and programs. At the workstations, all these resources are files. The transmission is performed using Tivoli NetView DM commands. The following table shows which commands can be used with which resources.

RESOURCE	C O M M A N D						
	DELETE	INITIATE CLIST	REPLACE	RETRIEVE	RETRIEVE BY CATALOG	SEND	SEND MESSAGE
CLIST	Yes	Yes	Yes	Yes	Yes	Yes	
DATASET	Yes		Yes	Yes		Yes	
DUMP	Yes		Yes	Yes		Yes	
MESSAGE						Yes	Yes
PANEL	Yes		Yes	Yes		Yes	
PROGRAM	Yes		Yes	Yes		Yes	

DELETE

The DELETE command (X'0283') deletes a file from a directory on a workstation disk.

RCMS regards the request as not valid, if RCMS cannot access the file to be deleted or the file does not exist on the workstation path; the host computer is notified of the error.

The table below shows some examples:

Example 1	
Tivoli NetView DM qualifier	AAA
LNF entry	AAA C:\FB2\TEST.DAT
Result	The workstation deletes a file named TEST.DAT from the C drive, FB2 subdirectory.
Example 2	
Tivoli NetView DM qualifier	BBB.TEST
LNF entry	BBB C:\UTIL
Result	The workstation deletes a file named TEST from the C drive, UTIL subdirectory.
Example 3	
Tivoli NetView DM qualifier	CCC.TEST.DAT
LNF entry	CCC C:\
Result	The workstation deletes a file named TEST.DAT from the C drive, root directory.

INITIATE CLIST

Two commands are available, which can be issued only when working with Tivoli NetView DM 1.2 and following releases:

- The synchronous INITIATE CLIST command (X'0284') sends and processes a command list (CLIST) file at a workstation. Tivoli NetView DM waits for CLIST completion. RCMS sends an acknowledgement upon CLIST completion.
- The asynchronous INITIATE CLIST command (X'0288') sends and processes a command list (CLIST) file at a workstation. Tivoli NetView DM does not wait for CLIST completion. RCMS logs and modifies the acknowledgement in the RCMS.ACK file upon CLIST completion. Tivoli NetView DM also issues a QUERY command at fixed time intervals to check for CLIST completion and to find out the process result. The time interval is specified during plan creation.

The contents of the CLIST must be valid commands for the operating system where the CLIST is going to be initiated.

The CLIST file can be created at the host computer or at a workstation.

- If you create the CLIST at the host computer, the CLIST is stored in EBCDIC in the NetView DM CLIST repository.
- If you create the CLIST at the workstation and upload it to the host computer with the RETRIEVE CLIST command, the CLIST is stored in ASCII in the NetView DM CLIST repository.
- If you create the CLIST at the workstation and upload it to the host computer with the RETRIEVE DATASET command, you can specify uploaded data type as binary or non-binary:
 - If you specify binary, the CLIST is stored in ASCII in the NetView DM DATASET repository.
 - If you specify non-binary, the CLIST is stored in EBCDIC in the NetView DM DATASET repository.

If you plan to submit DATASETS as CLISTS, you must convert them first. For information on conversion of DATASETS into CLISTS, refer to the NetView DM documentation. See "Bibliography" on page 285 for full details.

If you stored CLISTS in ASCII, you must load RCMS with parameter /C: set to B. If you stored CLISTS in EBCDIC, you can omit this parameter or set it to E.

An example of a CLIST, which contains one statement to call a command file, follows:

```
CALL INITREP
```

A file called INITREP.CMD (for OS/2) or INITREP.BAT (for DOS and Windows NT) exists on the workstation.

Netview DM commands, INITIATE CLIST, OS/2 DOS Windows NT

The contents of this file can be as follows:

```
echo INITREP STARTED >> MYCLIST.LOG
C:\MYUTILS\REPORT.EXE
if errorlevel 1 goto INITREPFAILED
echo REPORT ENDED SUCCESSFULLY >> MYCLIST.LOG
goto ENDINITREP
:INITREPFAILED
echo REPORT FAILED >> MYCLIST.LOG
:ENDINITREP
```

When working with LANDP for OS/2, if the CLIST file contains START statements, RCMS *must* also have been loaded through a START statement. The CLIST file performance depends on the operating system underneath. Upon CLIST completion in IBM DOS, IBM OS/2, and Windows NT, the return code from the operating system is recorded in the CLISTEXE.STS status file, and the CLIST file is deleted.

When working with LANDP for DOS, the CLIST file is processed in the foreground, even though an application is running. Note that, during the CLIST process, the application stops; it continues running upon CLIST completion.

When working with LANDP for DOS, RCMS loads and executes the COMMAND.COM file to process the CLIST file. Note that, if the CONFIG.SYS file includes a SHELL statement, the COMMAND.COM path in the statement must be the same as that specified in the SET COMSPEC statement in the AUTOEXEC.BAT file.

When LANDP for DOS RCMS is loaded in expanded memory and a CLIST ends with the message:

```
CLIST too big or run overflow
```

you can prevent this from happening by increasing the environment size.

When working with LANDP for OS/2, the CLIST file is processed in the background if you load RCMS through a DETACH or LOADER command. However, if you load RCMS through a START command, the CLIST is processed in the foreground.

While the CLIST file is processed, the operating system takes full control of the system and other use of the workstation can be influenced.

When working with LANDP for DOS, the workstation operator must specify the /a:r parameter in the RCMS loading statement, which is included in the AUTOFBSS.BAT file. The LANDP for DOS loading statements are explained in the *LANDP Installation and Customization* manual.

Netview DM commands, INITIATE CLIST, OS/2 DOS Windows NT

The record format of the CLISTEXE.STS status file is described in the following tables:

Offset	Length	Contents
0	8	Date in the form dd/mm/yy
8	1	b (delimiter)
9	8	Time in the form hh:mm:ss
17	1	b (delimiter)
18	19	Start-of-execution message
37	1	b (delimiter)
38	12	CLIST file name

Offset	Length	Contents
0	8	Date updated
8	1	b (delimiter)
9	8	Time updated
17	1	b (delimiter)
18	36	End-of-execution message
54	1	b (delimiter)
55	8	RCMS return code for the processed CLIST

If the RCMS return code is not zero, the next fields in the record have the following format:

Offset	Length	Contents
63	4	bbbb
67	1	File identification number. See Table 28 on page 65.
68	2	bb
70	2	errno value (C error code)
72	2	bb
74	3	Operating system return code

REPLACE

The REPLACE command (X'0282') replaces a file in a directory on a workstation disk. The current file is overwritten when the new file is received from the host computer.

RCMS processes the command as a SEND command, if the file name does not exist. RCMS does not replace the workstation file, if the REPLACE command sends an empty file.

The table below shows some examples:

<i>Table 14. Examples of REPLACE command</i>	
Example 1	
Tivoli NetView DM qualifier	AAA
LNF entry	AAA C:\FB2\TEST.DAT
Result	The workstation receives a file named TEST.DAT in the C drive, FB2 subdirectory.
Example 2	
Tivoli NetView DM qualifier	BBB.TEST
LNF entry	BBB C:\UTIL
Result	The workstation receives a file named TEST in the C drive, UTIL subdirectory.
Example 3	
Tivoli NetView DM qualifier	CCC.TEST.DAT
LNF entry	CCC C:\
Result	The workstation receives a file named TEST.DAT in the C drive, root directory.

Netview DM commands, RETRIEVE, OS/2 DOS Windows NT

RETRIEVE

The RETRIEVE command (X'0391') retrieves a file from a directory on a workstation disk.

RCMS regards the request as not valid, if the file does not exist on the workstation disk; the host computer is notified of the error.

RCMS sends the file to the host computer, even if it is empty.

The table below shows some examples:

Example 1	
Tivoli NetView DM qualifier	AAA
LNF entry	AAA C:\FB2\TEST.DAT
Result	The workstation sends the file named TEST.DAT from the C drive, FB2 subdirectory.
Example 2	
Tivoli NetView DM qualifier	BBB.TEST
LNF entry	BBB C:\UTIL
Result	The workstation sends the file named TEST from the C drive, UTIL subdirectory.
Example 3	
Tivoli NetView DM qualifier	CCC.TEST.DAT
LNF entry	CCC C:\
Result	The workstation sends a file named TEST.DAT from the C drive, root directory.

If the file name contains one of the special characters * or ?, the RETRIEVE command performs the same function as the DIR command in IBM DOS. Then, RCMS sends the equivalent output for the directory specified to the host computer.

The table below shows some examples:

<i>Table 16. Examples of RETRIEVE command (2)</i>	
Example 1	
Tivoli NetView DM qualifier	DDD
LNF entry	DDD C:*.*
Result	The workstation sends the C drive root directory information to the host computer. The result is the same as using the IBM DOS DIR command (DIR C:*.*).
Example 2	
Tivoli NetView DM qualifier	EEE
LNF entry	EEE C:\FB2*.*
Result	The workstation sends the C drive, FB2 subdirectory information to the host computer. The result is the same as using the IBM DOS DIR command (DIR C:\FB2*.*).
Example 3	
Tivoli NetView DM qualifier	FFF
LNF entry	FFF C:\UTIL\TEST.*
Result	The workstation sends the C drive, UTIL subdirectory information, matching the specification TEST.*, to the host computer. The result is the same as using the IBM DOS DIR command (DIR C:\UTIL\TEST.*).

The file type parameter is ignored when the RETRIEVE command is used for DIR requests. Output is of the form:

```
The volume label in drive "disk_ID" is "volume label"
The Volume Serial Number is "volume number"
Directory of "disk_ID:\path\"
```

```
fname ext fsize date changed time changed
```

```
⋮
```

```
subdir <DIR>
```

```
⋮
```

```
⋮
```

```
number file(s) number bytes free.
```

RETRIEVE by CATALOG

The RETRIEVE by CATALOG command (X'0392') retrieves the contents of all the files in a directory on a workstation disk. Tivoli NetView DM can send only one qualifier. RCMS cannot send a file with attributes HIDDEN or SYSTEM.

RCMS regards the request as not valid, if the directory does not exist on the workstation disk; the host computer is notified of the error. If the directory exists but contains no files, the host computer is notified.

The table below shows some examples:

Example 1	
Tivoli NetView DM qualifier	AAA
LNF entry	AAA C:\FB2\FB3\
Result	The workstation sends all files from the C drive, FB2\FB3 subdirectory to the host computer.
Example 2	
Tivoli NetView DM qualifier	BBB
LNF entry	BBB C:\UTIL\
Result	The workstation sends all files from the C drive, UTIL subdirectory to the host computer.
Example 3	
Tivoli NetView DM qualifier	CCC
LNF entry	CCC C:\
Result	The workstation sends all files from the C drive, root directory to the host computer.

SEND

The SEND command (X'0281') adds a file to a directory on a workstation disk.

RCMS rejects the command, if the file name already exists; the host computer is notified of the error. If the path does not exist, RCMS creates it and places the file in the created path.

The table below shows some examples:

Table 18. Examples of SEND command	
Example 1	
Tivoli NetView DM qualifier	AAA
LNF entry	AAA C:\FB2\TEST.DAT
Result	The workstation receives a file named TEST.DAT in the C drive, FB2 subdirectory.
Example 2	
Tivoli NetView DM qualifier	BBB.TEST
LNF entry	BBB C:\UTIL
Result	The workstation receives a file named TEST in the C drive, UTIL subdirectory.
Example 3	
Tivoli NetView DM qualifier	CCC.TEST.DAT
LNF entry	CCC C:\
Result	The workstation receives a file named TEST.DAT in the C drive, root directory.

SEND MESSAGE

The SEND MESSAGE command (X'0285') sends a message from the host computer to the workstation, or indicates the status of the previous release *held phase*. The message is limited to 256 text characters.

RCMS logs the message on the EHCRCMS.MSG file, created at customization. The EHCRCMS.MSG file has the following format:

Offset	Length (bytes)	Content
0	8	Date in the form dd/mm/yy
8	1	Delimiter (blank)
9	11	Time in the form hh:mm:ss AM
20	2	Delimiter X'0D', X'0A' (CR,LF)
22	X	Message
X+22	2	End of log entry mark X'0D', X'0A' (CR,LF)

Note: The message length can be up to 256 bytes.

The message can be logged through the system manager server. Then, the workstation operator reads the message using the system manager operator.

Reporting process results

RCMS reports processing results of Tivoli NetView DM commands through two mechanisms:

- Using the system manager server, if it is loaded
- Using the EHCRCMS.LOG file

If the system manager server is present and loaded, the process results can be consulted using the system manager operator.

The process results are written in the file EHCRCMS.LOG, which is located in the process path specified in the file FBSSPATH.DAT. The EHCRCMS.LOG file is a wrap-around file with a fixed number of records, which you specify through the RCMS loading statement. Each record has a fixed length. The first record is the log header record, which specifies the sequence number of the last written entry.

There are two associated log entries for each Tivoli NetView DM command:

- The first log entry relates to the command reception
- The second log entry refers to the successful or unsuccessful completion of the command

Each entry consists of a common part and a variable part, the contents of which depend on the type of log entry.

The following tables describe the two parts of every type of log entry:

Offset	Length	Contents
0	4	Sequence number
4	1	b
5	8	Date (yy/mm/dd)
13	1	b
14	8	Time (hh/mm/ss)
22	1	b
23	4	Tivoli NetView DM command code in hexadecimal characters
27	1	b
28	2	Event number. See Table 20 on page 63.

Table 20. Event number description

Event No.	Event
1	ABEND_EVENT
2	TERMINATE_EVENT
3	SEND_FILE_EVENT
4	REP_FILE_EVENT
5	DEL_FILE_EVENT
6	RET_FILE_EVENT
7	RET_CAT_EVENT
8	INFORM_OP_EVENT
9	SYN_EXEC_LIST_EVENT
10	UNKNOWN_EVENT
11	SECURITY_EVENT
12	ROUTER_EVENT
13	ASYN_EXEC_LIST_EVENT
14	SNA_EVENT
15	QUERY_EVENT
16	SMGR_EVENT
17	UNATTEND_CLIST_EVENT
18	REMOTE_PHASE_RELEASE_EVENT

Table 21. Log entry variable part. Command reception.

Offset	Length	Contents
31	21	Message
53	26	Tivoli NetView DM qualifier

Table 22. Log entry variable part. Command successful completion.

Offset	Length	Contents
31	8	RCMS return code=0
39	1	b
40	36	Message

RCMS, reporting process results, OS/2 DOS Windows NT

Table 23. Log entry variable part. Command unsuccessful completion. Common part.

Offset	Length	Contents
31	8	RCMS return code≠0
39	1	b
40	1	Error class. See Table 24.
41	3	bbb

Table 24. Error class description

Error class	Cause/Action
1	Internal error. RCMS writes a file named EHCBAD.END that contains useful information about the error for your IBM representative. Contact your IBM representative.
2	Critical error. Configuration error, hardware error, client/server mechanism error; usually an unrecoverable error. Check configuration and hardware. If the problem persists, contact your IBM representative.
3	Data processing error. Check system status, log files, and traces.
4	Cryptography error. Check system status, log files, and traces.
5	Communications error. Check system status, log files, and traces.
6	Error detected by Tivoli NetView DM. Check system status, log files, and traces.
7	Translation error in DBCS mode. In LANDP for DOS, there is an error in a request to the ASCII-EBCDIC translation server or the ASCII-EBCDIC translation server is not loaded. In LANDP for OS/2 and Windows NT, it is any kind of translation error. In LANDP for DOS, make sure the ASCII-EBCDIC translation server is loaded and operational. For each operating system, make sure there are no DBCS characters in fields where they are not supported; such as, phase_id and plan_id in RCMSPLAN.NEW.

Table 25. Log entry variable part. Command unsuccessful completion. Operating system dependent errors.

Offset	Length	Contents
44	2	File identification number. See Table 28 on page 65.
46	1	b
47	2	errno value (C error code)
49	1	b
50	3	Operating system return code

Table 26. Log entry variable part. Command unsuccessful completion. LANDP server dependent errors.

Offset	Length	Contents
44	8	Server name
52	1	b
53	2	Function ID
55	1	b
56	8	LANDP server return code

Table 27. Log entry variable part. Command unsuccessful completion. Translation dependent errors for OS/2 and Windows NT operating system.

Offset	Length	Contents
44	2	File identification number
46	1	b
47	2	TransDt routine exit code value
49	1	b
50	3	TransDt routine return code value

Table 28. Identification of files used by RCMS

File No.	File name
1	RCMSPLAN.TXT
2	EHCRCMS.LOG
3	RCME.MSG
4	RCMS.LNF
5	RCMS.ACK
6	Data file
7	EHCRCMS.MSG

Chapter 5. I/O device servers



This chapter describes the servers that the LANDP family provides to support input and output devices. The following servers are described:

Financial printer server

The financial printer server lets client applications print data on several different types of single byte data financial printers.

IBM 4748 printer server

The 4748 printer server lets client applications print double-byte or single-byte data on the IBM 4748 Document Printer and the IBM 9055-001 Document Printer.

IBM 4770 printer server

The 4770 printer server lets client applications print data on the 4770 ink jet printer specifically designed to print on forms used in financial transactions and for retail point-of-sale.

Printer manager server

The printer manager server supports an environment where print requests come from more than one program at a time.

Magnetic stripe reader/encoder server

The magnetic stripe reader/encoder (MSR/E) server lets client applications use the features of MSR/E devices.

Personal Identification Number (PIN) pad server

The PIN Pad server provides functions to use the features of PIN pad devices.

These servers have several common characteristics:

- The I/O devices can be accessed from applications written in a high-level programming language.
- Devices may be configured to be used by:
 - One workstation only (can be the workstation to which the device is attached, or another)
 - Multiple workstations in the LANDP workgroup

The server supporting a device must be located in the same workstation as that to which the device is physically connected.

- The I/O device servers permit the devices to be shared. The client/server mechanism accepts the call from the application, and routes it to the server. An I/O device is used exclusively by the application that has acquired it. For the device to be shared, applications must:
 1. Acquire the device
 2. Perform the desired task
 3. Release the device for use by other applications

I/O device servers, financial printer server

- The I/O device server functions are usually synchronous. This means that the application does not receive control until a called function has completed processing.

However, I/O device server functions that require human intervention or which, for other reasons, require a long time to be processed, can (and at times must) be used as asynchronous functions. This means that the function is only started, and control is returned to the application. The application can check repetitively to see if the functions have finished, or enter the wait multiple loop. In the latter case, the application regains control when the started function ends.

Financial printer server



The financial printer server lets client applications print data on these printers:

- IBM 4009 Universal Banking Printer
- IBM 4712 Transaction Printer
- IBM 4722 Document Printer
- IBM 4772 Universal Financial Printer
- IBM 9055-002 Document Printer
- IBM 9068-S01 Multi-Purpose Passbook Printer
- IBM 9069 Multi-Purpose Transaction Printer

The financial printer server also supports:

- The magnetic stripe reader/encoder (MSR/E) unit on the 4722 and 4772 printers, 9055-002 printer, and 9068-S01 printer
- The MICR reader of the 9060 printer

See *LANDP Introduction and Planning* for a description of these products.

Each server supports a different combination of devices.

- In LANDP for DOS:
 - Up to five printers attached to a personal workstation, such as a Personal System/2, an IBM ValuePoint, or an IBM ThinkPad®. Four can be connected to serial ports and one to the parallel printer port.
 - Up to three printers attached to an IBM Personal Computer. Two can be connected to serial ports and one to the parallel printer port.
- In LANDP for OS/2:
 - Up to five printers attached to a personal workstation, such as a Personal System/2, an IBM ValuePoint, or an IBM ThinkPad. Four can be connected to serial ports and one to a parallel printer port.
 - A printer can also be directly attached to a parallel printer port. This configuration does not require a device driver and the printing facilities are similar to those offered by an IBM 4201 Proprinter. See the *LANDP Programming Reference* for additional information.

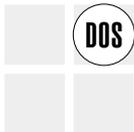
- In LANDP for Windows NT:
 - The IBM 4009 printer is not supported, except in 4722 emulation mode.
 - Up to five printers attached to a personal workstation, such as a Personal System/2, an IBM ValuePoint, or an IBM ThinkPad. Four can be connected to serial ports and one to a parallel printer port.
 - A printer can also be directly attached to a parallel printer port. This configuration does not require a device driver and the printing facilities are similar to those offered by an IBM 4201 Proprinter. See the *LANDP Programming Reference* for additional information.
- In LANDP for AIX:
 - Multiple serial printers can be attached to the RS/6000 server. A serial printer can also be attached to an Xstation.

A printer attached to a serial port may be shared among multiple workstations in the workgroup or may be dedicated for use by only one application. This option is selected during customization.

Some printers support A/B operator keys. This mode is also selected during LANDP customization. When in this mode, a print call is not accepted until the required printer interface (A/B) has been made active by pressing either the A or the B key (except when printing formatted documents, when the format definition specifies that the format is not to be shared using the A/B keys).

During customization, different print formats for documents, journals, passbooks, and the magnetic stripe reader/encoder, can be defined. These document formats are used when the printer server is working in page formatted mode. See the *LANDP Installation and Customization* for more information about defining these features.

Special considerations for LANDP for DOS



One device driver is needed for each printer attached to a serial port.

The printer device drivers come with the printers and require installation using a customization program. This method of customizing includes duplicating the device driver for each serial port and identifying the character tables to be used. (See the documentation that comes with the printer for additional information.)

The RS-232 serial interface card is not interrupt compatible with SDLC communication adapters. Therefore, printers attached to serial ports in LANDP for DOS and OS/2 workstations cannot be used in the same computer where SDLC communications equipment is installed.

The capability to download character sets is supported using the font code pages supplied with the printer device drivers. When the device driver is customized, the font code page file to support the required national character set is selected. If additional code pages are required, the client application must provide the required support.

I/O device servers, financial printer server

If a power-on reset (POR) sequence occurs, the device driver restores the customized character font to the printer.

The server interface for printers connected to the parallel port use BIOS INT 17H conventions. As this interface provides a subset of bidirectional link capabilities, only certain functions are available. No A/B sharing or asynchronous status is provided.

Some printers can work with two simultaneously active code pages. One is the standard character set code page. The other is the downloaded code page character set. See the documentation that comes with the printers for additional information.

Financial printers that have MSR/E capabilities cannot be attached to the parallel printer port if MSR/E functions are required.

Special considerations for LANDP for OS/2



All models of the printers are supported and can be attached to personal computer systems, using the parallel or serial port.

The interface between the RS-232 serial interface card and the printer server is provided using the device driver and its associated API subsystem.

The device-support code provides print redirection using a device monitor placed in the STARTUP.COM file. This means that data for a parallel port is redirected to a printer attached to a serial port.

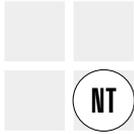
The printers can also be attached directly to the parallel port, using the file I/O subsystem. This provides for a subset of bidirectional link capabilities with only certain functions allowed at the server level. Functions that require bidirectional communication between the printer and the computer system are not allowed.

However, financial printers that have MSR/E capabilities cannot be attached to the parallel printer port if MSR/E functions are required.

The capability to download character sets is supported using the font code pages supplied with the printer device drivers. This requires no device driver customization. The code page is set by a client application using a function call. It is set according to the process environment. National Language Support (NLS) is provided by:

- Printer service functions that receive the requested code page ID, and set the code page at the device to switch fonts dynamically
- Using the system function calls described in the *LANDP Programming Reference*
- Preparing the system code pages using the NLS support provided by commands in the CONFIG.SYS file

Special considerations for LANDP for Windows NT



All models of the printers are supported and can be attached to personal computer systems, using the parallel or serial port.

Note: The 4009 printer is not supported, except in 4722 emulation mode.

The interface between the RS-232 serial interface card and the printer server is provided using the device driver and its associated API subsystem.

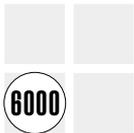
The printers can also be attached directly to the parallel port, using the file I/O subsystem. This provides for a subset of bidirectional link capabilities with only certain functions allowed at the server level. Functions that require bidirectional communication between the printer and the computer system are not allowed.

However, financial printers that have MSR/E capabilities cannot be attached to the parallel printer port if MSR/E functions are required.

The capability to download character sets is supported using the font code pages supplied with the printer device drivers. This requires no device driver customization. The code page is set by a client application using a function call. It is set according to the process environment. National Language Support (NLS) is provided by:

- Printer service functions that receive the requested code page ID, and set the code page at the device to switch fonts dynamically
- Using the system function calls described in the *LANDP Programming Reference*

Special considerations for LANDP for AIX



Only one server must be loaded in the RS/6000 system to support multiple printers attached to the RS/6000 system or Xstations.

The device driver diskettes that come with the printers contain font code pages that can be used to support a variety of character sets. Changing the font code page does not require customization.

National Language Support (NLS) is provided by:

- Printer server functions that receive the requested code page ID and set the code page at the printer to switch fonts
- Using system function calls (see *LANDP Programming Reference*)
- Defining a LANG environment variable
- Passing the code page and font as parameters when the server is loaded

IBM 4748 printer server



The 4748 printer server lets an application print data (both Single Byte Character Set (SBCS) and Double Byte Character Set (DBCS) data strings) or load the user-defined characters on an IBM 4748 Document Printer or an IBM 9055 Model 1 Document Printer. The application can also use functions provided by the magnetic stripe reader/encoder on the 9055 Model 1 printer.

See *LANDP Introduction and Planning* for a table of the code pages that LANDP supports.

The 4748 printer server supports up to two printers attached to the IBM Personal System/55 or IBM Personal System/ValuePoint serial port. One 4748 printer server must be configured for each workstation that has a printer attached. In a LANDP for DOS workstation, the appropriate device driver must be installed.

The printer may be shared among multiple workstations or it can be used only by one application. These options are configured during customization.

One way to share a printer is to use the A/B operator keys. This option is also selected during LANDP customization. When configured to use this option, a print call is not accepted until the required printer interface (A/B) has been made active by pressing either the A or the B key (except when printing formatted documents, when the format definition specifies that the format is not to be shared using the A/B keys). See the *LANDP Programming Reference* for information about sharing the printer or printer formats and whether pressing the A/B keys is required.

Special considerations for LANDP for DOS



One device driver is required for each printer attached to a serial port. The device drivers require customization. This step includes duplicating the device driver for each serial port. For more information about customizing the device driver, refer to *IBM 4748 Document Printer Programming Guide*, SA34-2090.

Special considerations for LANDP for OS/2



The supported printer can be attached to a Personal System/55 through the RS-232 Asynchronous Adapter Card. The interface between the RS-232 adapter card and the 4748 printer server is provided by COM.SYS or COMDMA.SYS and PBPCALLS.DLL.

During customization different formats for documents and passbooks can be defined. These are used when working in page formatted mode. The sharing mode of the 4748 printer is also defined during customization.

Special considerations for IBM DOS H7.0 and IBM OS/2 Warp H3.0



IBM DOS H7.0 and IBM OS/2 Warp H3.0 and later versions support the IBM Korean Standard (KS) code (code page 949), while the font file in the IBM 4748 Document Printer for Korea is based on the previous PC Code standard, hereafter PC code. The 4748 printer server converts DBCS IBM KS codes into PC codes automatically.

The conversion is made observing the following rules:

- IBM KS code range goes from X'8FA1' to X'FEFE'
- PC code range goes from X'8140' to X'FCFC' (X'nn7F' is excluded)
- If the given DBCS code is in the IBM KS code range and is a valid character, it is converted to the PC code that relates to the character. Otherwise, no conversion takes place.
- The IBM KS code in the user-defined character (UDC) range is converted to the related PC code as follows.

IBM KS code UDC range		PC code UDC range
-----		-----
X'C9A1' - X'C9FE'	→	X'B040' - X'B09E' (excludes X'B07F')
X'FEA1' - X'FEFE'	→	X'B09F' - X'B0FC'
X'8FA1' - X'A0FE'	→	X'B140' - X'B9FC' (excludes X'nn7F')

IBM 4770 printer server



The IBM 4770 printer server supports the IBM 4770 Ink Jet Transaction Printer.

The IBM 4770 Transaction Printer is an ink jet printer specifically designed to print on forms used in financial transactions and for retail point-of-sale. It uses thermal/bubble ink jet technology which results in a small footprint and quiet operation.

It can print up to six lines at 6.6 lines-per-inch on a check-sized cut form. It also supports three inch wide roll paper for a journal record or tear-off customer receipt (up to 5.5 lines-per-inch).

It supports a serial or parallel hardware interface. It also supports downloaded fonts but the client application must manage print pitch, line wrap, and so on.

No A/B interface is supported.

The 4770 printer server supports up to four printers attached to an IBM Personal System/2. Three can be connected to serial ports and one to a parallel printer port.

Printer manager server



The printer manager server supports an environment where print requests come from more than one program at a time. Although the printer manager server supports only printers attached to a parallel port, up to three parallel-attached printers in one workstation can be managed.

The print operations are serialized. One print request is completed before another is accepted. Locking situations are prevented. Print requests can come from:

- The 3287 printer emulator.
- The financial printer server, operating on a parallel-attached printer.
- User-developed printer servers (also the sample PROPRIN server).
- Non-LANDP programs, for example the DOS PRINT function or printing directly from an application. The printer manager server sees all such requests as coming from a non-LANDP resource.

The presence of the printer manager server in a workstation does not influence the operation of printers attached to a serial port.

Modes of operation

The printer manager server supports two modes of operation:

Alternate This is the default mode. The printer manager operates on a “first come–first served” basis.

When the printer is idle (after initialization, after a time-out, or after the last character is sent to be printed), the first print request arriving reserves the printer. The printer is reserved until the time-out expires or the last character is printed. Print Screen and Ctrl-Print Screen can only be used in alternate mode. When the printer manager server is in alternate mode, API functions are provided to assign a printer to one requester. See the *LANDP Programming Reference* for more information.

Exclusive The printer is assigned to one resource. Only print requests coming from this resource are served. All other requests will get a “busy” return code. Even if the assigned resource is not using the port, no other resource can use it.

To set a printer in exclusive mode you need the operator interface or the local resource manager. In exclusive mode, Ctrl-Print Screen returns a DOS error message.

Using the operator interface and the local resource manager

The printer manager server can be used with the operator interface or the local resource manager to:

- Issue commands to control the printing

- Display the status at a parallel printer port

If the optional operator interface is installed in the same workstation as the printer manager server, the following information is displayed for each parallel port:

- Number of the port
- Port status
- Use mode of the port (alternate or exclusive)
- Resources assigned
- Current owner of the port

The printer manager server can also be used without the operator interface or the local resource manager. If the operator interface and the local resource manager are not loaded, the port is assumed to be owned by the resource that first issues a request to print. Thus, when the printer is not being used, the first resource that requests to print is the one that acquires that port. The acquisition remains until the printer manager frees the port because of a time-out expiration. When the port is freed, it reverts to the initial state.

For more information about the operator interface, refer to “Operator interface” on page 182. For more information about the local resource manager, refer to “Local resource manager server” on page 192.

The printer manager server and DOS PRINT

To use DOS PRINT with the printer manager server, 3287 print emulator, financial printer server (on a parallel port), or your own printer server (on a parallel port), a special loader is used to load DOS PRINT. This loader remains resident and interacts with the printer manager server, preventing DOS PRINT from getting control if the printer is assigned to another requester.

The first command in the AUTOFBSS.BAT file is:

```
FBSSP <x:path PRINT command>
```

Where x:path is the path to the DOS PRINT.COM file

The FBSSP command word must be added before the normal PRINT command. This means that path parameters must also be provided. After this, later PRINT commands are issued in the normal way.

Magnetic stripe reader/encoder server



The magnetic stripe reader/encoder (MSR/E) is a desktop device that reads and encodes magnetic stripe documents. The following devices are supported:

- IBM 4717 MSR/E Unit (not supported on LANDP for Windows NT or AIX)
- IBM 4777 Magnetic Stripe Device

The MSR/E server provides functions to use the features of these devices. The available functions are described in *LANDP Programming Reference*.

On each platform, the server supports a different combination of devices.

In LANDP for DOS and OS/2

- The 4717 MSR/E is attached to the pointing device port on a personal workstation, such as a Personal System/2, an IBM ValuePoint, or an IBM ThinkPad.
- A 4717 MSR/E and a 4718 PIN pad can be attached to a single pointing device port using a special connecting cable.
- The 4777 MSR/E can be attached to an RS-232 serial port *or* to the pointing device port on an IBM Personal System/2.
- An MSR/E server is required for each workstation that has an attached MSR/E device.

In LANDP for Windows NT

- The 4717 MSR/E and 4718 PIN pad are not supported.
- The 4777 MSR/E can be attached to an RS-232 serial port.
- An MSR/E server is required for each workstation that has an attached MSR/E device.

In LANDP for AIX

- The 4717 MSR/E and a 4718 PIN pad are not supported.
- 4777 MSR/E devices can be attached to serial ports on the RS/6000 system where the server is installed. They can also be attached to Xstations on the same LAN where the RS/6000 system and server are installed.
- Only one server must be loaded in the RS/6000 system to support multiple MSR/E devices attached to the RS/6000 system or Xstations.

For a description of the supported models, see *LANDP Introduction and Planning*. For a complete description of the MSR/E devices, see *IBM 4700 Financial I/O Devices Programming Guide*, GC31-3770. For additional information about installing and configuring these devices, see *LANDP Installation and Customization*.

Personal identification number (PIN) pad server



PIN pads are used to read and encrypt personal identification numbers used in financial transactions. The following devices are supported:

- IBM 4718 Personal Identification (PIN) Keypad (not supported on LANDP for Windows NT)
- IBM 4778 PIN Pad Magnetic Stripe Reader

The PIN pad server provides functions to use the features of the PIN pad devices. For more information on the functions and how they are used, see the *LANDP Programming Reference*.

Each server supports a different combination of devices.

In LANDP for DOS and OS/2

- The 4718 PIN pad is attached to the pointing device port on a personal workstation, such as a Personal System/2, an IBM ValuePoint, or an IBM ThinkPad.
- A 4718 PIN Pad and a 4717 MSR/E can be attached to a single pointing device port using a special connecting cable.
- The 4778 PIN pad can be attached to an RS-232 serial port or to the pointing device port on an IBM Personal System/2.
- A PIN pad server is required for each workstation that has an attached PIN pad device.

In LANDP for Windows NT

- The 4718 PIN pad is not supported.
- The 4778 PIN pad can be attached to an RS-232 serial port.
- A PIN pad server is required for each workstation that has an attached PIN pad device.

In LANDP for AIX

- The 4718 PIN pad is not supported.
- 4778 PIN pad devices can be attached to serial ports on the RS/6000 system where the server is installed. They can also be attached to Xstations on the same LAN where the RS/6000 system and server are installed.
- Only one server must be loaded in the RS/6000 system to support multiple PIN pad devices attached to the RS/6000 system or Xstations.

See *LANDP Introduction and Planning* for a description of the supported models. For a complete description of the PIN pad devices, see *IBM 4700 Financial I/O Devices Programming Guide*, GC31-3770. See the *LANDP Installation and Customization* for additional information about installing and configuring these devices.

Chapter 6. Data management servers

This chapter describes the LANDP family servers that are used to manage data in a LANDP workgroup.

It contains information about the following servers:

Shared-file server

Provides database support to enable applications in a LANDP workgroup to access records in shared files.

Shared-file distributor server

Manages the distribution of disjoint database data among several shared-file servers.

Shared-file replicator server

Maintains exact copies of database data in different machines managed by shared-file servers.

Query server

Provides database services to applications that use structured query language (SQL) or the shared-file server interface.

ODBC query server

Provides database services to applications that use structured query language (SQL).

Electronic journal server

Provides a way to store data about all transactions performed in a LANDP workgroup for a user-defined period.

Store-for-forwarding server

Temporarily stores transactions intended for a host in the workgroup when the host communication is not available, or as a method to manage efficient host communication.

Forwarding server

Sends the transactions stored by the store-for-forwarding server to the host, and informs the store-for-forwarding server about the state of the host communication session.

This chapter also contains information on the shared DOS directory services.

Shared-file server



The shared-file server provides database support, enabling applications in a LANDP workgroup to access records in shared files. The shared-file server enables multiple applications to read the same file simultaneously, but prevents one user from changing a record while another user is changing it. A LANDP workgroup can support several shared-file servers. However, only one shared-file server can be loaded in a LANDP workstation. Any workstation that has a shared-file server requires a hard disk.

shared-file server, OS/2 DOS AIX Windows NT

To maintain shared files, the LANDP family provides various utility programs. See “Shared-file server utilities” on page 97 for more information.

The function of the shared-file server is independent of the programming language used to write the application. The shared-file server can use all the file systems supported by DOS, OS/2, AIX, and Windows NT.

The shared-file server manages the concurrent access of several applications to the same records or data files using two index files. The minimum disk space required for these index files is 4KB plus 60 bytes per lock.

To ensure continuous operation, the shared-file server uses two deadlock management techniques, one to avoid deadlocks and another for deadlock detection.

The shared-file server can operate in online or batch mode. This results in three possible status conditions: closed, online, and batch. Depending on the status of the server, the call syntax for the shared-file server varies and some functions are not allowed.

For double-byte character-set (DBCS) support, the shared-file server only manages DBCS information in user records.

Under OS/2 and Windows NT, when configured with the External Logging replicator (XLR) option, the shared-file server operates in fault-tolerant mode. XLR provides transaction-level replication of shared-file data on a separate workstation with no single-point-of-failure. This gives significantly better performance than the shared-file replicator server. See “External Logging Replicator (XLR)” on page 90.

Shared-file access methods

The shared-file server supports five ways to access records in the shared files:

Direct access

Records are accessed using a 32-bit binary record number.

Sequential access

Records are accessed sequentially from the current pointer using record numbers.

Indexed access

Records are accessed using keys. The keys are defined when the shared-file server is configured. Each key has two attributes:

- Change status - it can or cannot be changed
- Uniqueness - it is unique or is not necessarily unique

This results in four kinds of keys:

- Cannot be changed, unique (was primary)
- Can be changed, not unique (was secondary)
- Cannot be changed, not unique
- Can be changed, unique

A maximum of 15 keys can be defined.

Segmented keys are also supported. A segmented key is a key that is split inside different locations in the record. For more information about segmented keys, see “Segmented keys” on page 89.

Indexed sequential access

Records are accessed sequentially through the current pointer in indexed sequence. The sequence can be specified in a collate table. For more information about collate tables, see “Segmented keys” on page 89.

Direct indexed

Records are accessed by positioning an indexed program control block (PCB) pointer using a record number and scanning through the indexed pointer.

The file pointer

The shared-file server uses the concept of file pointers, sometimes called the current pointer. When the shared-file server is started, the file pointers point to the beginning of the file. The file pointers are updated to point to the next record when they have read a record. Records can be updated, inserted, deleted, or retrieved through any pointer.

Disk space management

The shared-file server uses a disk space management technique that makes it unnecessary to reorganize the files. If a record is deleted, the record space is marked as available for a later insert. This space is added to a chain of free records.

During customization the split size for records can be defined. Using this feature, the application can request the record length required rather than a fixed record length.

Shared-file server structure

Each shared-file server must be named during customization. The name must start with the six characters SHFILE and must contain two additional characters.

Each shared-file server can access up to 245 shared data and index files. If the total number of shared data and index files exceeds 245 files, the number of simultaneously open files must be specified when the server is loaded. This loading parameter can also be used to specify the maximum number of files used by an application. See *LANDP Installation and Customization* for information about using the /F loading parameter for the shared-file server.

Closed files are not protected against accidental erasing because they are not held.

The characteristics of the individual shared files are defined during customization in a table called the shared-file database description (DBD) file. A table of program control blocks (PCBs) is also defined. The PCBs describe the file pointers which associate a shared file with an access method.

The shared-file server, shared-file distributor server, and shared-file replicator server gather statistics about the primary functions used by the workstations. All statistics are

shared-file server log files, OS/2 DOS AIX Windows NT

available to the users of all connected workstations. During customization you can specify that the statistics are to be written in a file when all shared files are closed.

The shared-file server provides transactional data integrity by:

- Allowing simultaneous read-only access for several users.
- Opening the files in *deny write* mode. This means you cannot access a data file with system commands that access files in write mode, unless the data file is closed or the server is unloaded.
- Locking, at the record or shared-file level.

At the record level, only one program can change a record at a time. At the shared-file level, a shared file can be used exclusively by one workstation for batch processing.

- Logging:
 - Static and dynamic logging with one log file
 - Dynamic logging with one log file
 - Static and dynamic logging with two log files
 - Dynamic logging with two log files

Dynamic logging lets a program cancel a transaction if an error occurs or a workstation malfunctions. Static logging can restore all the shared files of a configuration from a backup copy and the log files.

Using two log files results in improved data integrity. If one log file gets damaged or you lose access to this file, you can continue your normal operation with the other log file, because both files contained the same information before one got damaged or you lost access to this file.

Note: For information on how to manage and handle backup strategies and log file generations, refer to “Shared-file server utilities” on page 97.

Log files

The shared-file server uses the log files, LOG.DAT and LOG2.DAT, to store information about the changes made to the data files. All the functions that involve changes (deletion, update, or insertion) to the data files are stored as log records. All records related to one transaction are chained, so that the shared-file server can identify the functions and the sequence in which they were performed for each single transaction.

The log files ensure integrity at the transaction level:

- At server loading time, the shared-file server scans the log files for any open transactions, which were neither committed nor rolled back in the previous working session. If the shared-file server finds any of these transactions, an automatic backout of these transactions takes place.
- At server run time, it is possible to roll back any transaction.

The log files also enable forward recovery. If data files are lost or damaged, it is possible to recover the data by using a backup of the data files and the log files that

includes all the changes done to the data files since that backup. The shared-file server reads the log files records and performs the logged functions again.

The log files must be open and, if the shared-file server is installed under a shared-file replicator server or a shared-file distributor server, initialization must be complete before any function calls can be served. Any attempt to use the shared-file server before these events are complete results in a return code.

For LANDP for DOS, OS/2, and Windows NT, the log files are (by default) placed in the root directory of the C: drive in the workstation owning the shared-file server. However, they can be placed in another drive, or even in another workstation, using the path defined through the environment variables LOGPATH for LOG.DAT and LOGPATH2 for LOG2.DAT, using a batch file, or entering the SET command at the command line. Often this results in improved data integrity and increased processing speed.

For LANDP for AIX, the files are placed in the root directory by default. You can redirect the files to a subdirectory but you cannot use remote drives.

In addition, when you use two log files, you should place them in different locations. For LANDP for DOS, OS/2, and Windows NT, this could be separate drives. For LANDP for AIX, you would use separate subdirectories. If one file gets corrupted, your programs can continue working with the other log file.

Examples for LANDP for DOS, OS/2, or Windows NT:

```
SET LOGPATH=D:\
SET LOGPATH2=E:\LOGDIR02
```

These statements place LOG.DAT in the root of drive D: and LOG2.DAT in drive E:, directory LOGDIR02, which could be located in the same or in another workstation.

```
SET LOGPATH=E:\LOGDIR01
SET LOGPATH2=F:\LOGDIR02
```

The previous statements place LOG.DAT in drive E:, directory LOGDIR01, which can be located in the same workstation as the shared-file server owner or another workstation, and LOG2.DAT in drive F:, directory LOGDIR02, which could be located in the same workstation as LOG.DAT or in a different one.

Example for LANDP for AIX:

```
SET LOGPATH=/LOGDIR001
SET LOGPATH2=/LOGDIR002
```

You must specify network drives in UNC format when shared-file is run as a service on Windows NT. For example:

```
SET LOGPATH=\\SERVER\LOGFILES
```

Opening log files

The log files must be open before any function call can be served. Any attempt to use the shared-file server before the log files are open results in a return code X'01004F50' (OP).

The OPENLOG utility can be used to open the log files. Refer to “Opening shared-file server log files—OPENLOG” on page 103 for detailed information about this utility. It should be used as soon as the log files become accessible and the shared-file server is loaded. It returns a message indicating either the log files status or the error message obtained when attempting to open the log files. This program can be called from the AUTOFBSS routine after loading the supervisor and the shared-file server.

Log file management

If you want to use two log files, you can modify the AUTOFBSS file so that it manages both log files. Following is an example of an OS/2 AUTOFBSS file that includes the necessary statements to manage two log files.

```
@ECHO OFF
LOADER SPV.EXE /pc-id
IF ERRORLEVEL 1 GOTO ERRSPV
:LOADSFS
LOADER SHFILE##.EXE [parameters] [/]L:{2|3}
IF ERRORLEVEL 176 GOTO ERRNOFW      REM RC=B0
IF ERRORLEVEL 175 GOTO ERRLOG2     REM RC=AF
IF ERRORLEVEL 174 GOTO ERRLOG1     REM RC=AE
IF ERRORLEVEL 1 GOTO ERRSFS
ECHO Shared-file server loaded.
GOTO END
:ERRNOFW
ECHO Attempting to perform forward recovery on truncated log file(s).
GOTO ERRSFS
:ERRLOG2
ECHO Secondary log file LOG2.DAT is not correct.
ECHO Copying primary log file LOG.DAT over secondary LOG2.DAT ...
COPY %LOGPATH%LOG.DAT %LOGPATH2%LOG2.DAT > NUL
IF ERRORLEVEL 1 GOTO ERRCOPY
ECHO Retrying shared-file server load ...
GOTO LOADSFS
:ERRLOG1
ECHO Primary log file LOG.DAT is not correct.
ECHO Copying secondary log file LOG2.DAT over primary LOG.DAT ...
COPY %LOGPATH2%LOG2.DAT %LOGPATH%LOG.DAT > NUL
IF ERRORLEVEL 1 GOTO ERRCOPY
ECHO Retrying shared-file server load ...
GOTO LOADSFS
:ERRCOPY
ECHO Error copying files.
:ERRSFS
ECHO Unloading supervisor ...
EHCFREE SPV /F
IF ERRORLEVEL 1 ECHO Unexpected error unloading supervisor.
GOTO END
:ERRSPV
ECHO Error loading supervisor.
:END
```

The statements in the example do the following:

1. Load the supervisor.
2. Load the shared-file server with two log files. For information about the loading statements, refer to the *LANDP Installation and Customization* manual.
3. Verify that none of the log files is truncated when attempting to perform a forward recovery. If one of them or both are truncated, AUTOFBSS stops processing.
4. Verify that both log files are correct. If one of them is not correct, the correct one is copied over the incorrect one. Then, AUTOFBSS tries again to load the shared-file server.
5. End the shared-file server load, if an error condition other than those described in the previous steps is detected.

You must set the environment variables LOGPATH and LOGPATH2 before starting AUTOFBSS. Note that in LANDP for DOS you should load the shared-file server first and then the supervisor. For LANDP for AIX, load the supervisor first and then attach the shared file.

If your log files were generated with a previous LANDP version, you must perform the following actions before loading the shared-file server:

1. Run the DBBACKUP utility from that LANDP version. For further information about the DBBACKUP utility, refer to “Backing up shared files—DBBACKUP” on page 103.
2. Delete the log files.
3. Run the current GENLOG utility to generate the new log files. For further information about the GENLOG utility, refer to “Generating shared-file server log files—GENLOG” on page 97.

Special considerations for LANDP for DOS



You can also use the shared DOS directory services for your log file:

1. If you want sharing protection security, load the IBM DOS SHARE program in the workstation where the shared DOS directory server is loaded.
2. Use the SHRSHARE command in the workstation where the shared DOS directory server is located to share the directory where the log file will be stored.
3. Use the following FILESYS command in the workstation where the shared-file server is located:

```
FILESYS d: SHRDIRDD shortname
```
4. Perform OPENLOG in order to open the log files in the shared-file server workstation.

Security is not affected.

shared-file server log files, OS/2

If you unload the shared-file server and want to load it again, you have to detach the logical drive that contains the log file with the following FILESYS command:

```
FILESYS d: /D
```

To attach the logical drive again, after the shared-file server is loaded, repeat step 3 on page 85 from the previous set of steps.

To regain access to the log files, repeat step 4 on page 85 from the previous set of steps.

Refer to “Shared DOS directory services” on page 137 for additional information on the use of the shared DOS directory services.

If you want to use the OS/2 LAN server for remote log support, you must modify the DOSLAN.INI file of the DOS LAN requester in the shared-file server workstation. You should set digits “a” and “b” in the /WRK parameter to zero (default is 1). You should modify the IBMLAN.INI file in the OS/2 LAN server workstation where you want to place a log file as explained in “Special considerations for LANDP for OS/2.”

Notes:

1. The shared-file server for DOS does not load if the log file has ever been used, on OS/2 or Windows NT, by the shared file server running in external log replicator (XLR) mode.
2. When migrating to LANDP for DOS in LANDP Version 5, it is advisable to shut down the shared-file server cleanly. The log contents have changed between Versions 4 and 5, therefore you must run the GENLOG command to create a new log file.

Special considerations for LANDP for OS/2



You must set the LOGPATH and LOGPATH2 variables in the same OS/2 session as that in which you intend to run LANDP for OS/2. If you are running several shared-file server utilities in more than one OS/2 session, you must set the same LOGPATH and LOGPATH2 variables in all these OS/2 sessions. In order to avoid repeating the previous steps, you can modify your CONFIG.SYS file instead, by adding the following statements:

```
SET LOGPATH=D:\
SET LOGPATH2=E:\LOGDIR02
```

If you want to use the OS/2 LAN server for remote log support, you have to modify the IBMLAN.INI file before using the OS/2 LAN server for remote log management for the first time. You must modify the IBMLAN.INI file in the OS/2 LAN server workstation where you want to place the log file and in the shared-file server workstation. The WRKHEURISTICS digit 32 has to be set to zero (default is 2). It is also recommended to set the WRKHEURISTICS digits 2, 3, 9, 20, and 23 to zero. For further information, see *OS/2 LAN Server Network Administrator Reference Volume 2: Performance Tuning*.

If the shared-file server is installed under a shared-file distributor server or a shared-file replicator server, an ECHSFYK1.IX1 file is created for internal use. By default this file is located in the C:\ directory. However, this file can be located to another directory using the EXTPATH environment variable.

For example:

```
SET EXTPATH=D:\
```

Special considerations for LANDP for Windows NT



You must set the LOGPATH and LOGPATH2 variables in the same Windows NT session as that in which you intend to run LANDP for Windows NT. If you are running several shared-file server utilities in more than one Windows NT session, you must set the same LOGPATH and LOGPATH2 variables in all these Windows NT sessions.

Remote logging on Windows NT: If remote logging is to be used and the shared-file server is to run as a service, the service must be given authorization to access the network drive. This is done using the *Services* applet in the control panel:

- Locate and select the LANDP SHFILE## server
- Click on *Startup*
- Click on *This account* and fill in the userid and password of a user that has access to the network drive

Note: The user ID need administrator access to the machine on which the shared-file server is running.

- Click *OK*

Remote logging on Windows 2000: The authorizing mechanism for access to remote drives is essentially the same as on Windows NT. The method for accessing the correct panel is slightly different:

- Open the administrative tools folder in the control panel and select *Services*.
- Locate and select LANDP SHFILE## server.
- Select *Properties* and then the *LogOn* tab.
- Fill in the user ID and password as for Windows NT.
- Click *OK*.

After authorization: On Windows NT or 2000, the above procedure must be repeated if the SHFILE## service is reinstalled or if the password changes.

After authorization is granted, you can set LOGPATH (or LOGPATH2) to point to the directory containing LOG.DAT (or LOG2.DAT) using the universal naming convention (UNC) format. For example:

```
SET LOGPATH=\\SERVER\DISK\SFSLLOG
```

Special considerations for LANDP for AIX



Although you can redirect the location of the log files to separate subdirectories, you cannot use remote drives.

Segmented keys



The shared file server supports locating records using segmented keys. A segmented key is a key that is split inside different locations in the record. Each key segment has five attributes:

- Kind of segment:
 - Data
 - Indicator
 - Locally nullable
 - Globally nullable
- Length: the length of the segment
- Offset: the offset inside the record where the segment starts (0 based)
- Collate: identifies the collate table to be used
- Null value: the null value used only if the segment is nullable

The maximum length of a key is 255 bytes. Therefore, the sum of the lengths of the segments cannot be greater than 255 bytes. Up to 32 segments can form a key.

To provide for national language support (NLS), the shared-file server can use up to 16 user-defined collate tables to determine the index sequence.

Two kinds of collate tables can be used:

- User-defined collate tables created during LANDP customization (up to 16)
- The default collate table provided by OS/2 based on the code page installed in the system

Shared-file server lock management



The shared-file server manages the concurrent access of several applications to the same records or data files by means of two index files, EHCLOCKS.IX1 and EHCLOCKS.IX2. These files are placed, by default, in the root directory of drive C: in the workstation owning the shared-file server. However, you can place these files in a virtual disk by setting the environment variable LOCKSPATH the same way you set LOGPATH and LOGPATH2 (see “Log files” on page 82). You cannot locate these files in a remote workstation.

The minimum disk space required for the index files is 4KB. However, you must consider that the shared-file server uses 60 bytes per lock.

DBCS mode restrictions



The shared-file server manages DBCS information in user records only. Therefore, you must take into account the following restrictions:

- Environment variables LOGPATH, LOGPATH2, and LOCKSPATH must contain path names consisting of SBCS characters only.
- The names of the files and profiles generated by the customization program and any others handled by the shared-file server must consist of SBCS characters only. If these files contain information about path names, file names, record names, DBD names, PCB names, key names, and workstation names, this information must consist of SBCS character strings only.

External Logging Replicator (XLR)



Note: XLR cannot be used in the same system as either the shared-file replicator or the distributor.

With the External Logging Replicator (XLR) facility, the shared-file server provides high data availability with low performance impact.

As described in “Log files” on page 82, the shared-file server uses LOG.DAT (and optionally LOG2.DAT) to record all updates to a shared-file database. When using XLR, the server also sends update log data to a backup shared-file server running on another workstation. The backup shared-file server writes the log data it receives to its own LOG.DAT and applies the update to a mirror set of data files.

If a hardware or software failure occurs on the active shared-file workstation, the backup automatically takes over as the active by:

1. Backing out (rolling back) any inflight transactions (that is, any uncommitted transactions that were running at the time of the failure)
2. Rebuilding its index files
3. Restarting in active mode

This happens without manual intervention.

After a successful takeover, the former active server **must** be brought up in backup mode as quickly as possible. If the active shared-file server logs any updates while the backup is not available, the log records are retransmitted to the backup when it restarts. This process is called synchronization. A backup cannot take over while it is in synchronization mode.

Servers in an XLR configuration

In an XLR configuration, the active shared-file server is registered as SHFILE01 and the backup as BKFILE01. The supervisor is always registered as SPV. An example XLR configuration comprises the servers in Figure 7.

Workstation AA (Active)	Workstation BB (Backup)
SPV	SPV
LAN	LAN
EHCSAM	EHCSAM
SHFILE01	BKFILE01

Figure 7. External logging replicator, servers

SHFILE01 and BKFILE01 are both running SHFILE##.EXE. During shared-file server initialization, the workstations interrogate the Service Availability Manager (EHCSAM) to determine which workstation was last run as the active. This workstation registers itself as SHFILE01 and the backup registers as BKFILE01.

Notes:

1. A workstation can contain only one shared-file server.
2. One shared-file server can run under Windows NT and the other under OS/2. Either can be the active server.
3. XLR cannot be used in conjunction with either the Shared-File Replicator or the Distributor.

Service Availability Manager (EHCSAM)

The Service Availability Manager (EHCSAM) must run on every workstation that contains an XLR server. EHCSAM manages state data for servers using XLR. When an XLR server is started or a takeover occurs, EHCSAM dynamically reconfigures LANDP routing tables on all workstations in the workgroup.

The routing tables correlate servers and workstations. For example, in Figure 7 on page 91, if BB becomes the active, EHCSAM switches the registrations of SHFILE01 and BKFILE01 between the two servers. After a takeover, and when workstation AA is recovered, the configuration looks like Figure 8:

Workstation AA	Workstation BB
(Backup)	(Active)
SPV	SPV
LAN	LAN
EHCSAM	EHCSAM
BKFILE01	SHFILE01

Figure 8. External logging replicator, servers after takeover

On each workstation in which it runs, EHCSAM maintains state data in a file called EHCSAM.CFG. This file should be backed up with other important files. EHCSAM.CFG is renamed by the SAMCFG procedure from EHCSAM.CFN, which is distributed by the GETTING procedure. After LANDP installation and customization, NEWCFG and GETTING are used to distribute run-time files to all the workstations in the LAN. Take care to maintain the correct state information when GETTING is run for an existing configuration. For further information about NEWCFG and GETTING, see *LANDP Installation and Customization*.

Messages

On OS/2, use EHCPMPRT to display messages from XLR servers.

On Windows NT, use the Windows NT Event Viewer to view XLR server messages, which are logged as events.

Effect of XLR takeover on applications

During a takeover, while inflight transactions are backed out and the indexes are rebuilt, the shared-file server is not available to process application requests. As a result, an application may receive:

- Router errors (Lx return codes) or
- Return codes caused by the application's session being broken, for example NO (Not Open)

Applications should be coded to handle these situations, typically by a short wait and retry.

Log files

The log files are critical to the correct operation of XLR. Never delete or manipulate the logs using anything other than the supplied utilities.

The log files on the active and backup should be kept in step by regenerating them at the same time using GENLOG. (See page 97.) GENLOG uses information from the old log while creating a new log.

If the old log on either machine is not available or becomes unusable, use the following procedure:

1. If LOG.DAT is available on the active, start Shared-file to ensure that all backouts and forward recovery have been done. Stop Shared-file.
2. Copy all data files (*.DAT) from the active to the backup.
3. Delete all index files on both machines (*.IX?)
4. Delete LOG.DAT (and LOG2.DAT if dual logging is used) on both machines.
5. Run GENLOG on both machines.
6. Start active and backup.

Relationship between components

Figure 9 on page 95 shows the relationship between the principal elements in a LANDP system using a shared file server with the replicator feature.

Figure 9 shows three machines, named at the right-hand side as **Active Server**, **Backup Server**, and **Applications**. This is the minimum realistic configuration. The servers must be in different machines.

There can be up to 245 shared databases, and up to 243 applications. Figure 9 shows a single database and application. This is sufficient to describe the processing.

Figure 9 is accompanied by detailed notes on the facing page.

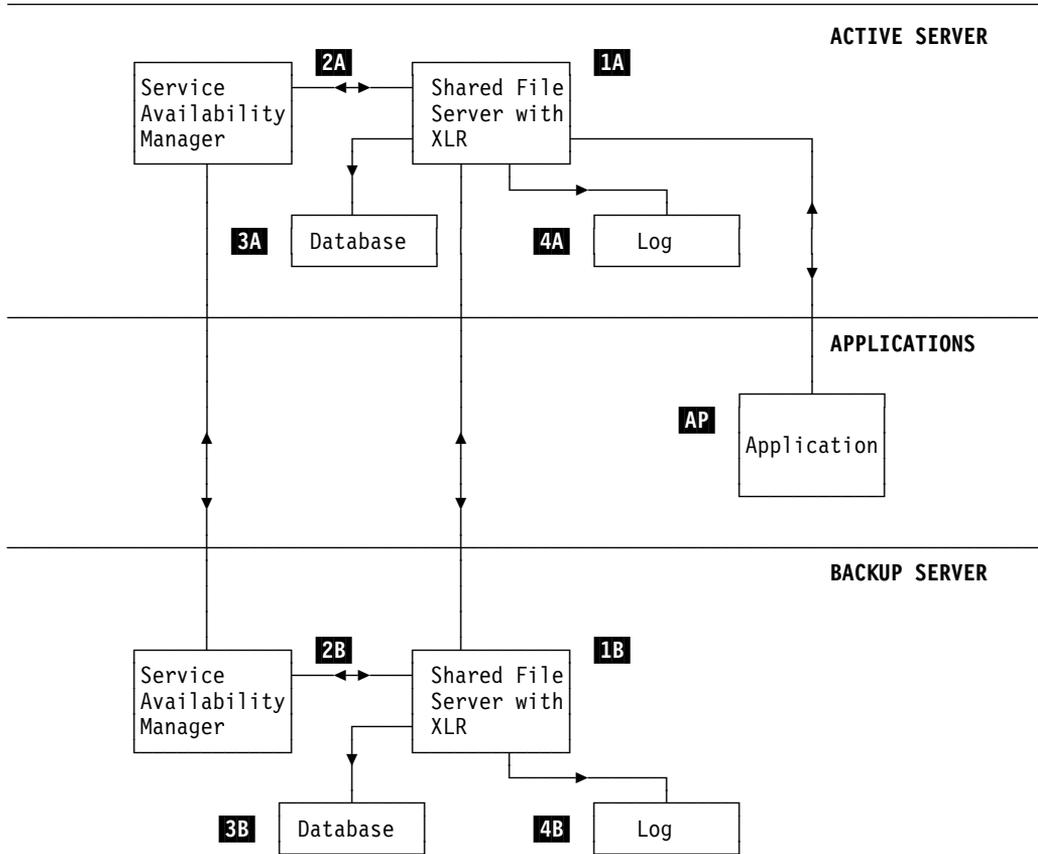


Figure 9. Shared file server with external log replicator (XLR)

Normal processing:

1. The application **AP** sends a database request to the active shared file server with XLR **1A**.
2. **1A** accesses the active database **3A** (updating the database for a write request).
3. **1A** returns the response to **AP**.
4. For an update (write) request
 - a. **1A** writes a log record to the active log **4A**.
 - b. **1A** sends the request to the backup shared file server with replicator **1B**.
The remaining processing is in the backup server and asynchronous with active server processing of the next application request
 - c. **1B** updates the backup database **3B**.
 - d. **1B** writes a log record to the backup log **4B**.

The result of the above processing is:

The backup database **3B** is an exact copy of the active database **3A**.

The backup log **4B** is an exact copy of the active log **4A**.

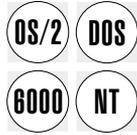
The only impact on active server processing is a single message to the backup server for each update request.

Emergency processing: Service availability managers (**2A** and **2B**), called SAMs, run in every machine that is running the shared file server with replicator. The SAMs are in constant contact. If the active shared file server fails for any reason, the backup SAM **2B** updates the routing tables in all application machines that were being served by the former active shared file server **1A**. As a result:

- The backup shared file server **1B** becomes the active shared file server.
- The backup database and log (**3B** and **4B**) become the active database and log respectively.
- Application processing continues almost seamlessly. See “Effect of XLR takeover on applications” on page 93.

As soon as the failed machine is again serviceable, the SAMs initiate a handshake process between the SFSs, which compare their two logs. During its wait time while supporting application processing, the active SFS sends log records to the backup SFS, which updates its log and database for each record received. When the two logs and databases are again synchronized, normal processing resumes with the original roles of the two servers reversed. At this stage, the administrator can optionally request that the original active server reverts to that status. If this request is made, the service availability manager (SAM) updates the routing tables accordingly.

Shared-file server utilities



The shared-file utilities are provided to support complementary functions for managing some shared-file server features. The shared-file utilities enable you to:

- Generate shared-file server log files; see page 97.
- List shared-file server log files; see page 98.
- Open shared-file server log files; see page 103.
- Back up shared files; see page 103.
- Restore shared files; see page 104.
- List shared-file server statistics; see page 105.
- List shared-file server profile definition; see page 107.
- Update a shared-file server profile; see page 109.

When calling these utilities, any required additional information is specified by using parameters. You can specify parameters in one of the following ways:

```

/P
(-P for LANDP for AIX)
/P:nn
(-P:nn for LANDP for AIX)
  
```

where:

P Is a parameter identifier
nn Is a required value.

You can specify parameters in either lowercase or uppercase.

These utilities can display messages on the screen. The text of these messages can be redirected to a file through the operating system output redirection facility.

Unless otherwise indicated, the return codes can take the following values:

- 0 The utility has completed operation successfully.
- 1 The utility has not completed operation successfully.

Generating shared-file server log files—GENLOG

The GENLOG.EXE utility is provided to generate one or two log files. If no log file, LOG.DAT or LOG2.DAT, exists when you invoke this program, it generates the log files. If the log files already exist, the program checks the log files for any open transactions and data files that are not in a committed state. If the program finds any open transactions or data files that are not in a committed state, it does not generate the log files.

Otherwise, the program renames LOG.DAT to LOG.OLD and LOG2.DAT to LOG2.OLD and creates new files, LOG.DAT and LOG2.DAT, which have a header record containing log general information (see “Listing shared-file server log files—TYPELOG” on page 98). The new log files are created where indicated by the LOGPATH and LOGPATH2 environment variables. If the LOGPATH and LOGPATH2 variables are not

shared-file server utilities, TYPELOG, OS/2 DOS AIX Windows NT

defined, the new log files are created in C:\ directory or in the root directory for LANDP for AIX.

To run the GENLOG utility, enter:

```
GENLOG [parm1 [parm2]] [/D [/F]]
```

where:

parm1 Is a positive integer value that specifies the number of 4096-byte blocks for the log file when it is created. The default value is 1.

parm2 Is a positive integer value that specifies the number of 4096-byte blocks by which the log file has to be increased when it becomes full. The default value is 1, if *parm1* is not specified. If *parm1* is specified, the default value is the *parm1* value.

/D Indicates that two log files have to be generated.

/F Indicates that two new log files have to be generated when the existing log files do not match.

Example

```
GENLOG 20 5
```

This specifies that 20 4096-byte blocks are required when the log file is created, with an additional 5 4096-byte blocks being used when the log file is full.

Listing shared-file server log files—TYPELOG

The TYPELOG utility is provided to display the information contained in the shared-file log files.

To run the TYPELOG utility, enter:

```
TYPELOG [/W:parm1 [/R:parm2 | /0] | /H] [/D]
```

where:

parm1 Specifies the name of the shared-file server client workstation you wish to display information about.

parm2 Is a positive integer value that specifies a record number in the log file. The information held from that record to the end of the log file is displayed.

If /0 is specified, the program displays information about all the open transactions. If the parameter is omitted, the program displays information about all the transactions.

If /H is specified, the program displays information only about the header; that is, it displays information only about the log record with record number 0.

If /D is specified, the program displays information about both log files, LOG.DAT and LOG2.DAT.

shared-file server utilities, TYPELOG, OS/2 DOS AIX Windows NT

You can specify each parameter alone or combine them as follows:

`/W:parm1 /R:parm2`

Displays records corresponding to workstation *parm1* starting from record *parm2*.

`/W:parm1 /0`

Displays records corresponding to open transactions for workstation *parm1*.

`/R:parm2 /0`

Displays records corresponding to open transactions starting from record *parm2*.

If you specify the `/D` parameter with any of these combinations, the TYPELOG utility displays the above information about both log files.

If you invoke the program without any parameters, all log file information is displayed.

The TYPELOG utility displays two kinds of records:

- The header record (record number 0) that contains general information about the log files
- Records that contain information about the functions performed to the data files

Following is an example of the information contained in these records:

LOG RECORD 0	[LOG.DAT	LOG2.DAT]
Length	: 1024	1024
Date of generation	: 20/7/1994	20/7/1994
Time of generation	: 18:2:39	18:2:39
Date of last close	: 20/7/1994	20/7/1994
Time of last close	: 18:5:23	18:5:23
Records initially formatted	: 64	64
Records to be added	: 64	64
Configuration name	: DBSCONF1	DBSCONF1
First byte free	: 1812	1812
First log record of workstation DD	: 0	0
First log record of workstation GG	: 0	0
First log record of session 0	: 0	0
First log record of session 1	: 0	0
First log record of session 2	: 0	0
.		
.		
.		
.		
First log record of session 241	: 0	0
First log record of session 242	: 0	0
Check	: 1024	1024

Figure 10. Shared-file server log file header record

Length Is the record length in bytes.

shared-file server utilities, TYPELOG, OS/2 DOS AIX Windows NT

Date and time of generation

Is the date and time the log file was generated.

Date and time of last close

Is the last time the shared-file server was closed.

Records initially formatted

Shows how many records were initially created for the log file. This number is calculated based on the first parameter specified with the GENLOG utility.

Records to be added

Shows how many records are added when the log file becomes full. This number is calculated based on the second parameter specified with the GENLOG utility.

Configuration name

Is the profile file that the shared-file server is using. This field is set to blanks in a newly generated log file.

First byte free Is the offset of the first free byte in the log file.

First log record of workstation DD

Is the offset of the first log record written for the workstation DD current transaction. If this value is different from 0, a transaction is opened for the main session of that workstation.

First log record of session X

Is the offset of the first log record written for the additional session current transaction. If this value is different from 0, there is an open transaction for this session. If this log record is not 0 for a workstation or for a session at shared-file server loading time, an automatic backout takes place.

Check

Contains the length of the record to allow the shared-file server to check it with the one written in the first field. If the two fields match, the record is valid. If not, the log is damaged.

```

LOG RECORD 1 BELONGING TO LOG.DAT AND LOG2.DAT
Date           : 20/7/94
Time          : 11:10:22
RBA           : 1500
Length        : 281
Rollback indicator : 0
RBA of preceding record: : 1300
Session number      :
Workstation identification : CC
Shared-file record number : 1
Shared-file name    : DPSD256
Operation performed : IS
Transaction number  : 1
Operation number    : 2
Record             : CUSTOMER IDENTIFICATION
                   : 0000000001CUSTOMER ADDRESS AND
                   : J. Q. CUSTOMER      TRTRFEJ
Check            : 281
    
```

Figure 11. Shared-file server log file detail record

- Log record x** Is the number of the logical record in the log file.
- Date** Is the date when the shared-file server function was performed.
- Time** Is the time when the shared-file server function was performed.
- RBA** Is the offset in the log file where the record starts.
- Length** Is the logical record length.
- Rollback indicator**
 Is set to 0 if no rollback was performed. The field is set to 1 if the record has been restored by the rollback function. It is set to 2 if it is the last record in the rolled back transaction. It is set to 3 if the record belongs to a transaction prepared to commit when installed under a shared-file distributor server or a shared-file replicator server.
- RBA of preceding record**
 Is a file pointer to the address of the previous record corresponding to the same transaction. This field is set to 0 for the first record written for a transaction.
- Session number** Is the number of the additional session, if used by the performed operation. If the transaction was performed in the main session of the workstation, the session number is not displayed.
- Workstation identification**
 Is the workstation ID of the workstation that has performed the operation.

shared-file server utilities, TYPELOG, OS/2 DOS AIX Windows NT

Shared-file record number

Is the number of the record in the shared file affected by this operation.

Shared-file name

Is the name of the shared file affected by this operation.

Operation performed

Is the operation that the workstation has performed. Not all the shared-file server functions produce log entries; only the following ones do:

- Delete record (DL) function
- Erase shared-file data (ZD) function
- Insert record (IS) function
- Replace record (RP) function.

If the shared-file server is installed under a shared-file replicator server or a shared-file distributor server, the following server functions also produce log entries:

- End Transaction (ET) function
- Checkpoint (CP) function
- Close batch (CB) function
- Rollback (RB) function.

For information on shared-file server function calls, refer to the *LANDP Programming Reference*.

Transaction number

Is a number used by the shared-file replicator or distributor servers to identify the shared-file server function

Operation number

Is a number used by the shared-file replicator or distributor servers to identify the shared-file server function

Record

Contains the record. Depending on the server function, this record will be:

- A record deleted, for the DL function
- A record inserted, for the IS function
- An old record followed by the corresponding new record, for the RP function

If you do not specify any record split size, the whole record is displayed. If you specify a record split size and the DBD profile is in the shared-file server current path, every physical record is displayed in a separate line. If you specify a record split size but the DBD profile is not in the shared-file server current path, the whole record is displayed with some internal information among the physical records.

Check

Contains a copy of the record length.

Example

```
TYPELOG /W:01 /0
```

This displays shared-file log file information for workstation 01, showing all open transactions.

Opening shared-file server log files—OPENLOG

The shared-file server looks for the log files at loading time, either in the path defined with the LOGPATH and LOGPATH2 environment variables or in the default path, and tries to open them. If the log files are not accessible the shared-file server accepts only open session (OS), test status (TS), and statistics request (SR) function calls before the log files are opened.

A log file is not accessible when it does not exist or when it resides in a workstation that is not accessible. If the log files do not exist, they can be generated using the GENLOG utility. If the log files reside in another workstation, they must be made accessible.

The OPENLOG utility should be used to open the log files. If the utility opens the log files, requests can be sent to the shared-file server. However, if the utility cannot open the log files, messages showing shared-file server return codes will be displayed.

To run the OPENLOG utility, enter:

```
OPENLOG
```

Backing up shared files—DBBACKUP

You should back up data in the shared files at regular intervals to prevent data from being lost in case of any failure. You should back up the following types of files:

.DBD, .PCB, and .SEQ (if there are collated keys) files

These are the shared-file server profiles defined during customization. These files do not change, therefore, you have to back them up only once.

.DAT files

These contain shared-file data. You must back up these files every time you perform a backup procedure.

.IX? files

These are the index files. Their backup is optional. Nevertheless, the restore process varies depending on whether you have backed them up.

The log files

These are used by the restore process to do some checking. However, you can perform the restore process without the log files. It is highly recommended to back up the log files every time you perform the backup process (see Restoring shared files—DBRESTOR).

You can use the operating system BACKUP command to back up files. First, you need to check that the files are in a committed state and the previous working session ended properly.

shared-file server utilities, DBRESTOR, OS/2 DOS AIX Windows NT

The DBBACKUP utility is provided to do this checking.

To run the DBBACKUP utility, enter:

```
DBBACKUP [/D]
```

If you specify the /D parameter, both log files, LOG.DAT and LOG2.DAT, will be checked. However, once you have run DBBACKUP successfully, it is recommended to back up LOG.DAT only.

If DBBACKUP runs with a 0 return code, you can back up all the files. If it runs with a nonzero return code, you should stop the backup process. When you finish the backup process, you may generate a new log file. This file would contain all changes done since the backup.

An example of the backup procedure is provided for each LANDP program:

- BCK.BAT for LANDP for DOS
- BCK.CMD for LANDP for OS/2
- BCK.BAT for LANDP for Windows NT
- dczybck for LANDP for AIX

The shared-file server should not be loaded during the backup process.

Restoring shared files—DBRESTOR

If data in the shared files is lost, it is possible to recover it by means of a backup copy and the log files that contain all the changes since that backup.

The restore process varies depending on the files that were backed up:

- If LOG.DAT was backed up, it should be restored as LOG.OLD in the path where the current log file is located.
- If the index files, .IX? were not backed up, they should be deleted before starting the shared-file server again.

You can restore files using the operating system RESTORE command. Then, some checkings are necessary before you start the shared-file server. The DBRESTOR utility is provided to do these checkings. It checks that the last closing time of the restored log file, LOG.OLD, is the same as the creation time of the current log file, LOG.DAT. If both times match, the utility marks the current log file so that the shared-file server performs a forward recovery. If both times do not match or LOG.OLD does not exist, DBRESTOR returns a nonzero return code and does not mark the log file.

To run the DBRESTOR utility, enter:

```
DBRESTOR[ /F] [/D]
```

If you specify the /F parameter, you can force the restore process to take place under any circumstances. No log validations take place and the log file is marked to perform a forward recovery.

If you specify the /D parameter, both log files, LOG.DAT and LOG2.DAT, will be checked. Once the log file is marked and the shared-file server is loaded, the forward recovery takes place.

If the DBRESTOR return code is 0, the log files chain and a forward recovery will take place.

An example of the restore procedure is provided for each LANDP program:

- RST.BAT for LANDP for DOS
- RST.CMD for LANDP for OS/2
- RST.BAT for LANDP for Windows NT
- dczyrst for LANDP for AIX

Listing shared-file server statistics—TYPESTAT

The TYPESTAT utility displays the statistic file that is generated by the shared-file server, shared-file distributor server, or the shared-file replicator server when they are loaded with the /E flag. The statistic file can only be read when all the shared-files are closed.

To run the TYPESTAT utility, enter:

```
TYPESTAT [/P:parm1]
```

where:

parm1 Specifies the path and the name of the statistic file to list.

There are three different file extensions:

- .EST** For the shared-file server
- .ESD** For the shared-file distributor server
- .ESR** For the shared-file replicator server

If the parameter is omitted, the last statistic file of the shared-file server in the current path is listed. If only the path is specified, the last statistic file of the shared-file server in that path is listed. If only the name is specified, only the statistic file for the shared-file server is searched in the current path.

To specify a statistic file other than the last one, the name of the file must have the following format:

```
[MMddhhmm].XXX
```

where:

- MM* Is the month when the file was generated
- dd* Is the day when the file was generated
- hh* Is the hour when the file was generated
- mm* Is the minute when the file was generated
- XXX* Is the file extension:

- .EST** For the shared-file server
- .ESD** For the shared-file distributor server
- .ESR** For the shared-file replicator server

Example

```
TYPESTAT /P:02211415.EST
```

This command displays a specific statistics file for the shared file server.

The information is presented in tables of the following format:

STATISTICS LISTING OF ACTIVE WORKSTATIONS IN FILE *[file path+.file name]*

WORKSTATION (POOL SESSION	CC X)		
FUNCT		ERR	OK
BT		0	3
IS		1	11
DL		0	4
RP		2	12
RB		0	2
GET/HOLD/KEEP/FETCH	14		65

Figure 12. Information from shared-file server statistics file

- **CC** is the workstation ID. All workstations that have interacted with the shared-file server are listed.
- **X** is a session number.
- The first column shows the shared-file server function called. For information on shared-file server function calls, refer to the *LANDP Programming Reference*.
- The second column shows the total number of function calls that ended with a nonzero return code.
- The third column shows the total number of function calls that returned 0K after processing.

Note: Each statistic refers either to a workstation or to an additional session. Statistics for workstations show the workstation ID and statistics for additional sessions show the corresponding session number.

Listing shared-file server profile definition—TYPECONF

The TYPECONF utility is provided to display the profiles that define the database descriptions (DBD) and the program control blocks (PCB). To run the TYPECONF utility, enter:

```
TYPECONF [/C:parm1 [/D:parm2]]
```

where:

parm1 Specifies the profile file name. The default value is CONFIGUR. Note that the profile is searched in the current directory.

parm2 Specifies the DBD name. If specified, *only* the information related to that DBD is displayed. If the parameter is omitted, the whole profile is displayed.

The information is presented as follows:

DBD INFORMATION

DBD name : F1DBD
Record name : FIREC
Files path and name : D:\BASEDB1\F1
Data record length : 80
Record split size : 50
Number of keys : 2

Key name : F1P
Key length : 30
Key class : 0
Segment type : 2
Null value for nullable segment : 0x20
Segment length : 15
Segment offset in data record (0 based) : 20
Collated segment : SE2
Segment type : 1
Segment length : 15
Segment offset in data record (0 based) : 5
Collated segment : 1
Segment type : 0
Segment length : 5
Segment offset in data record (0 based) : 0
Collated segment : 0
Segment type : 0
Segment length : 5
Segment offset in data record (0 based) : 35
Collated segment : 0
Segment type : 3
Null value for nullable segment : 0x00
Segment length : 5
Segment offset in data record (0 based) : 40
Collated segment : 0

Key name : F1S
Key length : 15
Key class : 1
Key offset in data record (0 based) : 5
Collated key : SED

PCB INFORMATION

PCB name : PCBF1P
DBD name : F1DBD
Key name : F1P

PCB name : PCBF1S
DBD name : F1DBD
Key name : F1S

Figure 13. Shared-file server profile definition

In this example, the DBD F1DBD has 2 keys.

- The F1P key is 30 bytes long, is non-duplicatable and non-modifiable, and is composed of 5 segments.

- The first segment is to use collate table PROFILE.SE2. It is locally nullable, with the nullable value 0x20.
- The second segment is to use PROFILE.SEQ. It is an indicator segment.
- The remaining three segments are not to be collated. The last segment is globally nullable with the nullable value 0x00. The other segments are data segments.
- The F1S key is duplicatable modifiable and will be collated using PROFILE.SED table.

The command that produced the output above was:

```
TYPECONF /C:SHFLPRF1
```

Updating a shared-file server profile—UPDTCONF

You can update existing shared-file server profiles with the UPDTCONF.EXE program. To run the program, enter:

```
UPDTCONF parm1 {/ext:dbdname | parm2 {/exta11 | /add:dbdname | /adda11}}
```

where:

parm1 Specifies the file name of the profile file to be updated.

/ext: Extracts a DBD and its related PCBs from profile file *parm1*.

dbdname Is the name of the DBD to be added to or extracted from profile file *parm1*.

parm2 Specifies the file name of the profile file used as a reference to update *parm1*. It must be different from *parm1*.

/exta11 Extracts from *parm1* every *parm2* DBD and its related PCBs.

/add: Adds to *parm1* a *parm2* DBD and its related PCBs.

/adda11 Adds to *parm1* every *parm2* DBD and its related PCBs.

If a key in the DBD you are adding has to be collated through a table defined in the existing profile, an error message is returned. If a collated table is defined in the existing profile only, or in the profile to add only, this table is used.

After extracting DBDs from a profile, the resulting unused collating tables are erased.

Note that the UPDTCONF program searches for the profile files in the current directory.

Shared-file distributor server



The shared-file distributor server manages the distribution of disjoint database data among several shared-file servers. This permits the disjoint database data of several shared-file servers to appear as a single database and provides transaction integrity, error recovery, and deadlock management.

Figure 14 shows an example of a shared-file distributor server that manages database data in two shared-file servers. From the client perspective, the database data is seen as a single database when accessed through the shared-file distributor server.

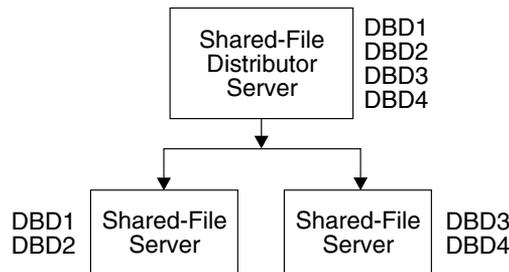


Figure 14. Shared-file distributor server

This server can continue to operate if at least one shared file server is operational. This makes it possible for clients to work on a partial database. If a read or write is attempted on a portion of the database that is not active, an error condition is returned to the application. The client application can decide if a transaction will or will not occur. This feature gives the application designer maximum flexibility when designing applications. It also maintains the integrity of transaction data by not allowing portions of a transaction to be saved.

The interface for the shared-file distributor server is the same as the shared-file server.

The shared-file distributor server provides data integrity at the transaction level. This means that a transaction is either committed or rolled-back for all the involved servers even if the server experiences an abnormal stop.

The shared-file server, shared-file distributor server, and shared-file replicator server gather statistics about the primary functions used by the workstations. All statistics are available to the users of all connected workstations. During customization you can specify that the statistics are to be written in a file when all shared files are closed.

A shared-file distributor server can be installed in the same machine as a shared-file server or shared-file replicator server.

A tree topology is used to define the relationship between distributor servers, replicator servers, and shared file servers.

Three terms are used to describe the server relationships: root node, inner node, and leaf node.

- A root node is a shared-file distributor server that communicates with an application and sends requests to leaf nodes.
- An inner node is a shared-file replicator server that is installed under a root node. It receives requests from an application or shared-file distributor server and sends requests to a leaf node.
- A leaf node is a shared-file server that handles requests but does not make write requests for services.

A shared-file server is always a leaf node and a shared-file distributor server is always a root node.

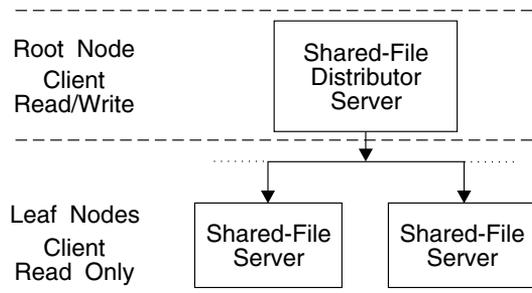


Figure 15. Shared-file distributor server topology

A leaf node cannot be accessed by more than one root node. However, a root node can access many leaf nodes.

The shared-file distributor server supports several shared file servers and/or several shared-file replicator servers each supporting several shared file servers. Only OS/2 shared-file servers are supported.

Although read requests directly to any shared-file server are supported, write requests can only be issued to the root server to avoid uncontrolled modifications to a database.

See the *LANDP Programming Reference* for information about the specific functions associated with the shared-file distributor server.

Shared-file replicator server



Note: For replication and fault tolerance with enhanced performance, see “External Logging Replicator (XLR)” on page 90.

The shared-file replicator server maintains exact copies of database data in different machines that are managed by shared-file servers.

Figure 16 shows an example of a shared-file replicator server that maintains identical database data in two shared-file servers.

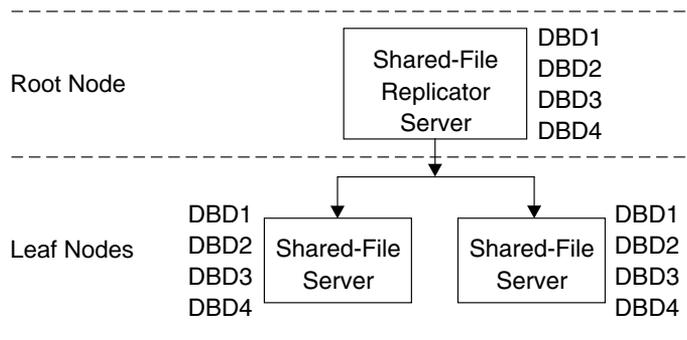


Figure 16. Shared-file replicator server

The shared-file replicator server is able to continue to operate if at least one current database is operational. An error is returned to an application only when all databases are inoperable.

The shared-file server, shared-file distributor server, and shared-file replicator server gather statistics about the primary functions used by the workstations. All statistics are available to the users of all connected workstations. During customization you can specify that the statistics are to be written in a file when all shared files are closed.

Each time a shared-file server is started, the shared-file replicator server determines whether the database is current. If the databases are not synchronized, database synchronization is performed by the shared-file replicator server.

The interface for the shared-file replicator server is the same as the shared-file server. See the *LANDP Programming Reference*.

A tree topology is used to define the relationship between distributor servers, replicator servers, and shared-file servers.

Three terms are used to describe the server relationships: root node, inner node, and leaf node.

- A root node is a shared-file replicator server that communicates with an application and sends requests to leaf nodes.
- An inner node is a shared-file replicator server that is installed under a root node. It receives requests from an application or shared-file distributor server and sends requests to a leaf node.
- A leaf node is a shared-file server that handles requests but does not make write requests for services.

A shared-file server is always a leaf node. However, a shared-file replicator server can be configured as a root node or installed under a shared-file distributor server as an inner node.

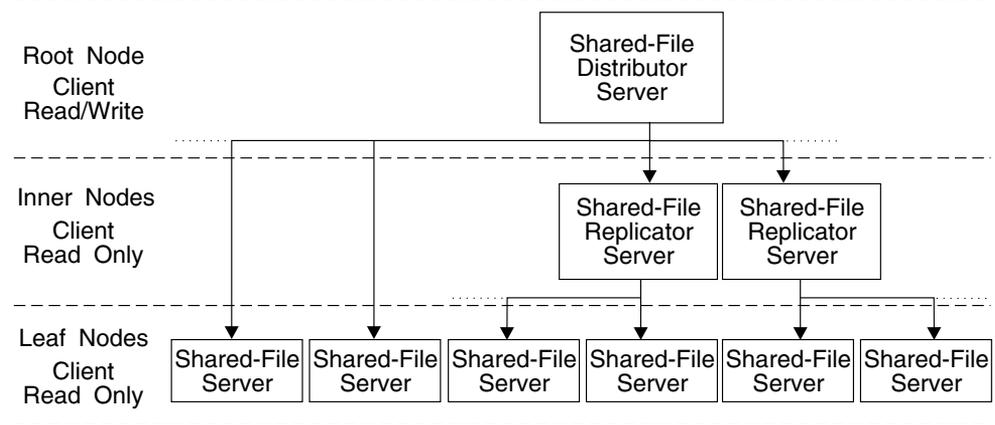


Figure 17. Example of shared-file distributor/shared-file replicator topology

A leaf node cannot be accessed by more than one root node. However, a root node can access many leaf nodes.

No server can provide services for more than one server. Therefore, each node has only one parent. However, any single node can have several children in the tree.

Each shared-file replicator server can support several shared-file servers. Only OS/2 shared-file servers are supported.

Also, a shared-file replicator server can be installed in the same machine as a shared-file server or shared-file distributor server.

Although read requests issued directly to any shared-file server are supported, write requests can only be issued to the root server to avoid uncontrolled modifications to a database.

See the *LANDP Programming Reference* for information about the specific functions associated with the shared-file replicator server.

Query server

 OS/2 6000

The query server provides database services to applications that use structured query language (SQL). A shared-file interface, similar to the shared file server, is also supported.

These servers extend the functions of the associated database management software. See *LANDP Introduction and Planning* for a list of the database manager software supported for each server.

The LANDP for OS/2 query server supports the remote access level of the Distributed Relational Database Architecture™ (DRDA®) which allows LANDP applications to transparently access remote DB2® databases.

Although several query servers can be present in a workgroup, each must be located in a different workstation.

The query server requires the database data to be stored on a hard disk. Access to the files supported by the query server is independent of the programming language used to write the application.

The main features of the query server are:

Data integrity

This is guaranteed by the supported database management software. Transactional integrity and the use of the rollback (or undo) function provide data integrity.

Shared multistation access

Every member of a LANDP workgroup, including LANDP for DOS, OS/2, and AIX workstations, can access the query server.

Compatibility with the shared-file server

The utility programs provided by LANDP can be used to migrate the shared-file data sets to SQL table structures.

Full SQL functions

For new applications, it is possible to issue SQL requests using normal SQL syntax.

Common interface

Applications use the same interface as all the LANDP family servers use. This makes it easier to produce new applications. No other libraries, binding, or pre-compile steps are necessary to link applications. Also, the error-recovery and low-level query-requester procedures need to be developed only once for the servers used with the application.

Forward recovery for applications

This lets the application programmer recover an up-to-date database from a backup copy of the database plus:

- A log file obtained during the normal operation of the applications that access the OS/2 databases. See the description of the utility programs, especially the
- A journal and journal receiver obtained during normal operation of applications that access OS/400 databases.

IBM DB2/2 provides forward recovery. You can choose between using LANDP for OS/2 forward recovery and the IBM DB2/2 standard routines.

Database interactions done without the LANDP for OS/2 interface cannot be recovered by means of the query server forward recovery. Therefore, you should use IBM DB2/2 forward recovery instead. In LANDP for AIX, forward recovery is provided by the IBM AIX RDBMS.

Access to AIX databases using LANDP for AIX query server

The LANDP for AIX query server enables applications to access a variety of supported AIX databases. For a complete list of supported LANDP for AIX RDBMS software, refer to *LANDP Introduction and Planning*.

Operating modes

The query server has three operating modes:

- Not logged in any mode
- Logged in shared-file server interface mode
- Logged in SQL dynamic-interface mode

When the server is in its not-logged mode, functions to enter and leave the logged-in modes and perform other duties can be issued.

When the server is in the shared-file server interface mode the server supports an expanded subset of the shared-file server online functions. For more information about the compatibility of functions when in this mode and the shared-file server functions, see the *LANDP Programming Reference*.

When the server is in the SQL dynamic-interface mode a client is able to send SQL queries and receive SQL responses. Additional functions are provided to perform row updates from a “full row” structure. Normal dynamic SQL coding is possible using this mode. Using the common API, there are several functions available to make requests.

For more information about functions when in this mode, see the *LANDP Programming Reference*. For information about tuning the performance of the query server, see the *LANDP Installation and Customization*.

Query server log file

query server log file, OS/2

The logo consists of the text "OS/2" inside a circle, which is positioned in the top-left corner of a 2x2 grid of squares. The top-right and bottom-right squares are shaded light gray, while the bottom-left square is white.

The query server log file, which is provided for forward recovery, is disabled by default. To use the log file, it must be enabled when loading the server.

The query server log file, EHCSQLOG.DAT, is placed, by default, in the root directory of the C: drive in the workstation where the query server is installed. However, LANDP for OS/2 enables you to place the log file in another drive or even in another workstation, which often results in improved data integrity and increased processing speed.

The log file is placed in a drive other than the default drive by setting an environment variable, LOGPATH, using a .CMD file, or by entering an OS/2 command.

Examples

```
SET LOGPATH=D:\
```

This command causes the log file to be placed in the root directory of drive D:.

```
SET LOGPATH=E:\MYLOGDIR
```

This command places the log file in drive E:, directory MYLOGDIR, which can be located in the same workstation as the query server or in a different one.

You must set the LOGPATH variable in the same OS/2 session as that in which you intend to run LANDP for OS/2. If you are running several query server utilities in more than one OS/2 session, you must set the same LOGPATH variable in all these OS/2 sessions. Instead of following the previous steps, you can modify your CONFIG.SYS file by adding the following statement:

```
SET LOGPATH=D:\
```

If you want to use the OS/2 LAN server for remote log support, you have to modify the IBMLAN.INI file before using the OS/2 LAN server for remote log management for the first time. You must modify the IBMLAN.INI file in the OS/2 LAN server workstation where you want to place the log file and in the query server workstation. The WRKHEURISTICS digit 32 has to be set to zero (default is 2). For further information, see *OS/2 LAN Server Network Administrator Reference Volume 2: Performance Tuning*, S10H-9681.

If you do not use the query-server log file, the forward recovery capabilities provided by IBM DB/2 1.0 can be used by making the required modifications in the Configuration Tool panel of the Database Manager.

Query server utilities



The query server provides utilities to support system design and setup tasks. These utilities enable you to:

- Generate a log file; see below.
- Examine a log file; see page 117.
- Perform a deferred open of a database; see page 118.
- Back up a database; see page 118.
- Restoring a database; see page 118.
- Invoke the query server forward recovery procedures; see page 119.
- Restart a database; see page 119.

Generating a log file—GENLOG2

The GENLOG2 utility is provided to generate a log file for the query server, with special formatting attributes. The initial size and the incremental growing size of the log file can be specified. If there is an existing log file, before creating a new log file, the existing log file is renamed with the same file name and the extension OLD.

To run the GENLOG2 utility, enter:

```
GENLOG2 [initblocks [addblocks]]
```

where:

initblocks Is the number of empty blocks, 4096-byte size, used to create the log file. The default is 1.

addblocks Is the number of empty blocks added when the log file gets full. The default value is 1 if *initblocks* is not specified, otherwise the default is the *initblocks* value.

Example

```
GENLOG2 20 5
```

This specifies that 20 4096-byte blocks are required when the log file is created, with an additional 5 4096-byte blocks being used when the log file is full.

Examining a log file—TYPELOG2

The TYPELOG2 utility is provided to dump a current log file to be examined. It may be useful for problem determination when forward recovery is unsuccessful.

To run the TYPELOG2 utility, enter:

```
TYPELOG2
```

The log file contains all the statements, and the associated data, issued through the query server to the IBM Database Server for OS/2 Warp, Version 4, on a work-unit basis. Note that only those operations issued in *log-habilitated mode* are stored on the log file.

Performing a deferred opening of a database—OPENDB2

The OPENDB2 program is provided to perform a deferred open of the SQL database and log file. It may be used, for example, when these files could not be opened at server load time.

To run the OPENDB2 program, enter:

```
OPENDB2 [servername]
```

where:

servername Is the complete name of the LANDP for OS/2 server to be opened. The default is EHCSQL01. The utility requests a profile/database name and a password for opening.

The OPENDB2 program does the following to open the database:

1. It connects to the database (“START USING”).
2. It reads and verifies the PCB CATALOG (EHCSQLTB table), allocating space for the system file pointers.
3. It opens the forward recovery static log file and verifies its contents.

Backing up a database—BACKUP2

The BACKUP2 program is provided to invoke a DBMS backup utility, in order to save a database.

To run the BACKUP2 program, enter:

```
BACKUP2 [/B: dbname] [/P: password] [/T: xxx] [/D: drive]
```

where:

dbname Is the name of the database to back up.

password
Is the database password, if it exists.

xxx Specifies total backup (*xxx*=A11) or incremental backup (*xxx*=Inc).

drive Is the backup destination drive.

If you do not specify these parameters, the utility requests them.

Restoring a database—RESTORE2

The RESTORE2 program is provided to invoke a system recovery utility, as a first step to using forward recovery.

To run the RESTORE2 program, enter:

```
RESTORE2 [/B: dbname] [/P: password] [/T: destdrive] [/D: origdrive]
```

where:

dbname Is the name of the database to restore.

password
Is the database password, if it exists.

destdrive
Is the target drive where the database is to be restored.

origdrive
Is the source drive where the database is to be restored from.

If you do not specify these parameters, the utility requests them.

Invoking the query server forward recovery procedures—PLAYLOG2

The PLAYLOG2 program is used to invoke forward recovery, which is performed by the query server. Prior to running the PLAYLOG2 program, you must run the RESTORE2 utility.

To run the PLAYLOG2 program, enter:

```
PLAYLOG2 [servername]
```

where:

servername
Is the complete name of the LANDP for OS/2 query server to work with. The default is EHCSQL01.

If you load the query server with logging disabled, you cannot run this utility.

Restarting a database—RECOVER2

The RECOVER2 program is provided to restart a database, if necessary, when the previous session ended abnormally. You do not need to have LANDP for OS/2 loaded to run RECOVER2. It is recommended that you add RECOVER2 to your AUTOFBSS.COM before loading the server. The utility will then perform a logon to database manager, if required, start Database Manager, and restart the database automatically.

To run the RECOVER2 program, enter:

```
RECOVER2 [/B: dbname] [/P: password] [/S]
```

where:

dbname Is the name of the database to restart. If not specified, the following message is displayed:

```
Enter Database name :
```

You can then provide the database name.

query server utilities, OS/2

password

Is the database password, if it exists. If not specified, the following message is displayed:

Enter Password :

You can then provide the password.

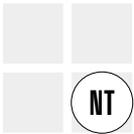
S Is a parameter you have to specify if you do not want to be asked for confirmation of the database recovery. If not specified the following message is displayed:

Press <Enter> to continue or <Esc> to cancel

You can then decide whether to continue the database recovery or cancel it.

If you enabled the auto-restart option for the database with the database manager configuration tool, RECOVER2 does not ask you for confirmation.

ODBC query server



The ODBC query server provides database services to applications that use structured query language (SQL). Unlike the LANDP for OS/2 query server, the shared-file interface is not currently supported.

This ODBC query server supports database manager software using a specification called Open Database Connectivity (ODBC). The ODBC query server interacts with database manager software through an ODBC Driver Manager and the ODBC Device Driver. It is the ODBC Device Driver that interacts with the database manager software. See *LANDP Introduction and Planning* for a list of database manager software supported for this server.

The main features of the ODBC query server are:

Data integrity

This is guaranteed by the supported database management software.

Transactional integrity is provided by the database management software.

Shared multistation access

Every member of a LANDP workgroup, including non-Windows NT workstations can access the ODBC query server.

Full SQL functions

The ODBC query server allows the client to issue normal SQL syntax to communicate with the database management software.

Common interface

Applications use the same interface as for all the LANDP family servers. This makes it easier to produce new applications. No other libraries, binding, or pre-compile steps are necessary to link applications.

Forward recovery

Forward recovery is provided by the use of the database management software logging functions. Forward recovery is done by the database management software itself.

Multidatabase connectivity

If a valid ODBC device driver exists, the ODBC query server can allow applications to access different databases that are held by different database management software.

Operating modes

The ODBC query server has currently two operating modes:

- Not logged in any mode
- Logged in SQL dynamic-interface mode

When the server is in its not-logged mode, functions to enter and leave the logged-in mode and to perform management tasks can be issued.

When the server is in the SQL dynamic-interface mode, a client can send SQL queries and receive SQL responses. Additional functions are provided to perform row updates from a “full row” structure. Normal dynamic SQL coding is possible using this mode. The common API supports several functions to make requests. The server supports two data type interfaces when running in SQL dynamic-interface mode. It supports the original LANDP for OS/2 query server data types, and another interface to support ODBC types. The ODBC data types interface allows the application program to use field type descriptor values as defined by ODBC, that is, it can use values SQL_CHAR, SQL_VARCHAR, SQL_INTEGER, and so on. The returned data format is also different. For example date and time types are returned as ‘C’ structures that are defined by ODBC, for example, SQL_C_TYPE_DATE. And decimals are returned as character strings rather than in packed decimal.

For more information about functions provided by this server, see the *LANDP Programming Reference*.

For information about tuning the performance of the ODBC query server, see *LANDP Installation and Customization*.

ODBC conformance levels

The ODBC query server uses the ODBC specification to communicate with the underlying RDBMS. ODBC contains various levels of conformance, with which an application and ODBC device driver must be compatible. The ODBC query server conforms to level 1. Although a device driver may conform to a conformance level, it does not necessarily support all attributes within that conformance level, and this can affect the behaviour of the ODBC query server.

query server utilities, OS/2

ODBC data source creation

An ODBC data source must be created before the ODBC query server can be used. This data source must be a system data source. For information on how to create a data source, refer to the ODBC administration documentation.

ODBC query server utilities

The ODBC query server provides a utility to support migration from OS/2 to Windows NT.

The LANDP for OS/2 query server needs the client machine to log on to the database. The ODBC query server can not use this mechanism. Instead, the GF (Grant) function (which must be issued once before any workstation can use the ODBC query server) has been extended to handle data source information (data source name, userid and password). Existing applications that issue a GF function still work without a data source connection. Before running applications that use ODBC query server, a workstation can run a utility called OPENODB (see "OPENODB - creating connection criteria") to issue the extended version of GF to include the data source information.

OPENODB - creating connection criteria

OPENODB informs the ODBC query server of the data source, the userid, and password to be used in the connection to the database management software. This utility is provided so that existing applications can work without too many changes. The utility issues a GF function to the ODBC query server with the data source, userid, and password as data. The connection is not created at the time of processing a GF, but the information is stored until the connection is created (that is, when an OQ (Open Query) function is issued).

To run OPENODB utility, enter:

```
$openodb /S:server name /D:data source /U:userid /P:password /I
```

where:

server name

Is the complete name of the LANDP ODBC query server.

data source

Is the name of the ODBC data source that has been set up to be used to access the database.

userid

Is the user ID that you wish to use for the connection.

password

Is the password used to authorise access for the given user ID.

/I

Enables interaction with the user. If a parameter is missing, OPENODB prompts the user for the missing data. In the case of the password, what the user types in will be echoed back as an * for each character.

Electronic journal server



The electronic journal server provides a convenient way of storing the data of all transactions in a LANDP workgroup during a chosen business period (for example, a day or a week).

The electronic journal contains the transaction records. The different record formats to be used are defined during customization.

The electronic journal server lets the application programmer write programs to create, manage, and process data in electronic journal files. For example, files can be created that contain all the transactions performed in one production site.

The electronic journal server offers:

- Access to records stored in the electronic journal using functions for adding, retrieving, updating, and deleting (logically).
- A search mechanism that can be used to search for specific records in electronic journals. See “Search operations” on page 126 for more information.
- Application control of the electronic journal files using functions for creation, access, and reuse of physical disk space.

The electronic journal is a LANDP family server, building on the capabilities of the LANDP for DOS, OS/2, and Windows NT shared-file server and the LANDP for AIX query server used in SQL dynamic-interface mode, for access to the individual data sets which compose the electronic journal. Therefore, electronic journal performance is highly affected by shared-file and query server performance. For more information about performance tuning, refer to the *LANDP Installation and Customization*.

The electronic journal server cannot be installed if the shared-file servers are installed under a shared-file replicator server or shared-file distributor server.

The LANDP for DOS, OS/2, and Windows NT electronic journal server uses the shared-file server with indexed sequential access. Several secondary indexes are available and can be used for efficient retrieval of all transactions originating from, for example, a particular user or workstation. The electronic journal server uses the system manager server to access record format definitions and other system support services.

The LANDP for AIX electronic journal server provides three functions to define, erase, and query record formats.

You can install several electronic journal servers in a LANDP workgroup. A LANDP for DOS, OS/2, and Windows NT electronic journal server must use a shared-file server located in the same workstation.

A LANDP for AIX electronic journal server must use a query server located in the same workstation. Two electronic journal servers cannot use the same shared-file or query

electronic journal files

server. Therefore, it is not possible to have more than one electronic journal server in a workstation. Applications refer to the desired electronic journal server using a unique server name, which is composed of the literal ELECJO##, where ## is the unique server identifier for the client workstation as defined during customization. Two electronic journal servers may have the same identifier, but an application cannot use two electronic journal servers with the same identifier.

Each electronic journal file has a physical journal name. As an option, each electronic journal file can be divided into logical journals using a grouping scheme that matches your needs. For example, logical journals could be defined for user IDs, workstation IDs, or any other grouping scheme you require. Logical or physical journal names can be used to access electronic journal files depending on the environment specified during customization. Up to 99 logical journals can be defined for a physical journal file.

Transactions, henceforth called *records*, are built by the application and sent to the electronic journal server. The electronic journal server builds the required header for the records. In LANDP for DOS, OS/2, and Windows NT, the server segments the record if necessary, and stores the resulting physical records in the data sets. In LANDP for DOS, OS/2, and Windows NT, the maximum record length is 4064 bytes. In LANDP for AIX, the maximum record length is 4048 bytes. If the RDBMS is IBM DB2, the maximum record length is 4005 bytes.

Once the electronic journal server is loaded, the application must wait until this server is initialized and operational by issuing a test initialization function call.

For more information, refer to the *LANDP Programming Reference*.

Electronic journal files

Up to 70 electronic journal files can be defined. The number of files and the physical record length to be used are defined during customization. The files wrap on themselves with the newest entries over-writing the oldest entries.

In LANDP for DOS, OS/2, and Windows NT, each electronic journal file is a data table accessed by the shared-file server using an internal key or one of the user-defined keys.

In LANDP for AIX, you can access the electronic journal records directly in the SQL database. However, you should access records directly only if the electronic journal server is not working properly. The electronic journal uses SQL to communicate with the query server by using the set of data types defined as LANDP for AIX data types in the electronic journal.

Records versus physical records

In LANDP for DOS, OS/2, and Windows NT, a record (or logical record) is the entity to be journaled by the application. The structure of each type of record to be used is defined during customization. A physical record is the entity which is actually stored in the shared files. A record (that is, a logical record) may be stored as several physical

records. Using the information in the record definition, the electronic journal server builds control information that is stored with the records in the files.

LANDP for AIX provides a utility program to define the structure of each type of record. For more information about this utility program, refer to “LANDP for AIX record definition utility program” on page 240. You can also define the structure of each type of record from your application by issuing a function call. For more information about function calls, see the *LANDP Programming Reference*.

The electronic journal server uses the shared-file server in LANDP for DOS, OS/2, and Windows NT, and the query server in LANDP for AIX to add, retrieve, delete, and update the records.

Data integrity for LANDP for DOS, OS/2, and Windows NT electronic journal files



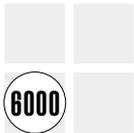
Data integrity of the electronic journal files is largely provided by the shared-file server. The shared-file server contains data integrity functions. Refer to the *LANDP Programming Reference*, for more information about these functions.

There are two data integrity options for electronic journals that you can specify during customization:

- The electronic journal server can be configured to obtain a separate shared-file server session for data access. In this case, the electronic journal server issues the data integrity functions. Opening and closing a separate shared-file session is controlled by a flag in the relevant electronic journal function. Using the separate session option, transactions can only access the electronic journal files.
- If the separate session option is not selected, the application must issue the shared-file server functions. In this case, the electronic journal server does not issue functions. This is the default configuration. Using this option, transactions can access electronic journal files and other shared files.

Note that the system manager is also involved in the data integrity of the electronic journal files, as part of its data checking. For more information see “User profile and application data maintenance” on page 147.

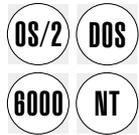
Data integrity for LANDP for AIX electronic journal files



Data integrity of the electronic journal files is largely provided by the query server, which contains data integrity functions.

For more information, refer to the *LANDP Programming Reference*.

Electronic journal environments



During customization, you define whether you are going to use the physical or the logical journal environment.

- **Physical journal environment**

A physical electronic journal is a file. Records are directly added to and accessed in the physical journals.

A physical journal name of your choice is assigned to each electronic journal file using the Allocate Physical Electronic Journal (AL) function.

In the physical journal environment, records are always added to, updated in, deleted from, or retrieved from a specific journal using the physical journal name.

A physical journal can be used to store records for:

- A particular period
- A specific user
- A specific user during a particular period

No logical journals exist in the physical journal environment.

- **Logical journal environment**

Each file (physical electronic journal) is divided into parts. In the logical journal environment, logical journals are accessed by applications using logical names.

The records within a physical journal (file) can be grouped. Each resulting group is called a logical journal and is accessed by an application using the logical journal name. This allows you to split a physical journal into many logical journals. The electronic journal server supports up to 99 logical journals per physical journal.

The maximum number of logical journals per physical journal to be used is defined during customization.

Typical examples of logical journal names used for access are:

- User ID
- Employee number or name
- Workstation ID
- Cash box number

In the logical electronic journal environment the records are always accessed using the logical journal name. As logical journals are part of physical journals, physical journals must be maintained as in the physical journal environment.

For more information about physical and logical journals, see the *LANDP Programming Reference*.

Search operations



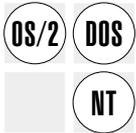
The electronic journal and store-for-forwarding servers provide a method for retrieving records based on search criteria that are built using the data field names, values, comparisons, and logical operators. In LANDP for DOS, OS/2, and Windows NT, the access can be based on one or both of:

- Defined secondary keys
- Values in the data fields

In LANDP for AIX, the standard SQL search conditions apply.

For more information about search operations, see the *LANDP Programming Reference*.

DBCS mode restrictions



The electronic journal server requires that the following information contains SBCS characters only:

- Record format names
- Record field names
- Physical and logical journal names

Electronic journal server utility



The EJUTIL.EXE program is provided to reallocate the electronic journal data sets, when the number of them has been increased. Because the electronic journal data sets are used in a wraparound way, the EJUTIL.EXE program ensures proper access to the new electronic journal data sets.

The number of electronic journal data sets is modified through the customization utility, resulting in a new electronic journal profile.

At the end of the day, after closing the shared-file server, you must run the EJUTIL.EXE program, if you have increased the number of electronic journal data sets. Note that it must be done before unloading the LANDP program. Upon completion, the LANDP program is unloaded. The next day, you will work with the new profile and will be able to store records on the new data sets.

To run the EJUTIL.EXE program in a workstation, the electronic journal server must be installed in the workstation and must provide services to that workstation.

To run the EJUTIL.EXE program:

1. Go to the drive where the utility is located, for example d:.
2. Go to the path where the utility is located.
3. Run the utility by typing:

```
EJUTIL
```

Store-for-forwarding server



The store-for-forwarding server temporarily stores transactions intended for a host system. This feature helps to maintain service if communication between the workgroup and the host is temporarily lost or as part of normal operations.

The forwarding server is also used to transmit the transaction records to the host. One transaction record can be forwarded to the host at the same time as another is being added. You can also develop your own forwarding server. Figure 18 on page 129 shows how the store-for-forwarding operates with the forwarding server, or your own applications or forwarding facility.

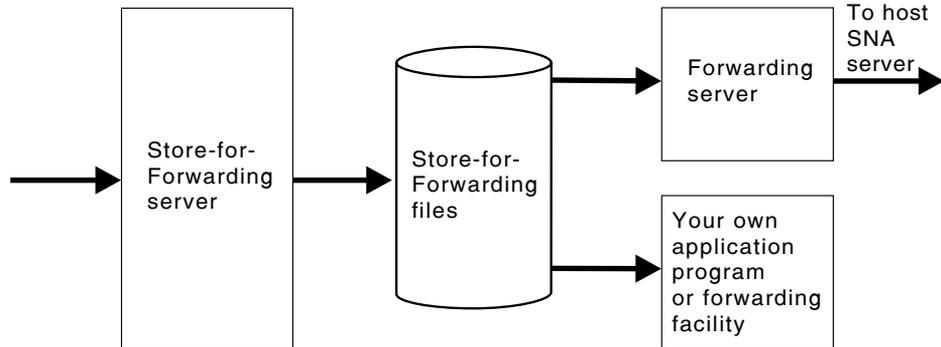


Figure 18. Store-for-forwarding Operation

A user-developed application uses the store-for-forwarding server to store records in one or more files (A). Several functions are available in the store-for-forwarding server. For example, stored records can be retrieved or updated.

The LANDP forwarding server retrieves the records, sends the records to the host using the SNA server, and physically deletes the records after the host receives them (B).

You can also develop your own application to process the stored records in any way that suits your operation, or you can develop your own forwarding server (C).

Whenever an application assumes responsibility for a record, the LANDP forwarding server will not attempt to forward such a record to the host. It remains a responsibility of the application to ensure that the respective records are processed and deleted from the file.

The LANDP for DOS, OS/2, and Windows NT store-for-forwarding server uses the shared-file server, which provides data integrity and recovery. Indexed sequential access is used. The keys are defined during customization.

The LANDP for AIX store-for-forwarding server uses the query server in SQL dynamic-interface mode, which provides data integrity and recovery.

Store-for-forwarding performance is highly affected by shared-file and query server performance. For more information about performance tuning, see the *LANDP Installation and Customization*.

Once the store-for-forwarding server is loaded, the application must wait until this server is initialized and operational. The application must issue a test initialization function call to verify that the server is initialized and operational. For more information, refer to the *LANDP Programming Reference*.

A customization option allows you to specify automatic transmission or application control of starting and stopping transmission.

Store-for-forwarding files

Each LANDP for DOS, OS/2, or Windows NT store-for-forwarding file is defined as a shared file with indexed sequential access. An internal key is used by the store-for-forwarding server to number the stored records. Other secondary keys, up to 14, are defined during customization.

In LANDP for AIX, you can access the store-for-forwarding records directly in the SQL database. However, you should access records directly only if the store-for-forwarding server is not working properly. The store-for-forwarding server uses SQL to communicate with the query server by using the set of data types defined as LANDP for AIX data types in the query server.

During operation the server maintains a file control table that reflects the current status of the store-for-forwarding files and their respective forwarding status.

Search capabilities are provided for records stored in the store-for-forwarding files. For further details see “Search operations” on page 126.

The store-for-forwarding server uses a single shared-file server with up to 64 files, with one or more indexes. The maximum number of files is defined during customization.

During customization a host application, to which stored records are forwarded, can be associated with each file.

Storing records

Functions are available to:

- Store (add) records
- Disable storing
- Enable storing

Initially all files are, by default, enabled for storing. The disable storing function can be used, for example, to allow the forwarding server to empty a file before new records are added.

Records versus physical records

Records to be stored are built by the application and sent to the store-for-forwarding server. The server adds the required header to the records. In LANDP for DOS, OS/2, and Windows NT, the server segments the record if necessary, and stores the resulting physical records.

In LANDP for DOS, OS/2, and Windows NT, the structure of each type of record to be used is defined during customization.

LANDP for AIX provides a utility program to define the structure of each type of record. For more information about the LANDP for AIX record definition utility program, see “LANDP for AIX record definition utility program” on page 240. You can also define the structure of each type of record from your application by issuing a function call. See

the *LANDP Programming Reference* for more information about the define record format function.

Data integrity for LANDP for DOS, OS/2, and Windows NT store-for-forwarding files



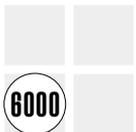
Data integrity of the store-for-forwarding files is largely provided by the shared-file server. The shared-file server contains data integrity functions to begin and end transactions and perform record operations. For more information about these functions, refer to the *LANDP Programming Reference*.

There are two data integrity options for store-for-forwarding files that you can specify during customization:

- The store-for-forward server can be configured to obtain a separate shared-file server session for data access. In this case, the store-for-forward server issues the data integrity functions. Opening and closing a separate shared-file session is controlled by a flag in the relevant store-for-forward function. Using the separate session option, transactions can only access the store-for-forward files.
- If the separate session option is not selected, the application must issue the shared-file server functions. In this case, the store-for-forward server does not issue functions. This is the default configuration. Using this option, transactions can access store-for-forward files and other shared files.

Note that the system manager is also involved in the data integrity of the store-for-forwarding files, as part of its data checking. For more information see “User profile and application data maintenance” on page 147.

Data integrity for LANDP for AIX store-for-forwarding files



Data integrity of the store-for-forwarding files is largely provided by the query server. The query server contains data integrity functions.

For more information, refer to the *LANDP Programming Reference*.

Forwarding stored records



The forwarding server sends the stored records to the host, and deletes the records from the store-for-forwarding file.

Depending on a customization option forwarding is either:

- Automatic, when at least one record is available for transmission and the communication link is up.
- Controlled by the application, using the functions start and stop transmission of stored records.

Forwarding sessions

There are three forwarding sessions that have predefined names:

SESSION1
SESSION2
SESSION3

During customization these sessions must be associated to appropriate IMS or CICS sessions.

Forwarding priority

To control the forwarding process, a store-for-forwarding file priority scheme is implemented, based on options selected during customization. Forwarding can be requested to:

- Alternate (or not) between forwarding sessions (LANDP for DOS, OS/2, and Windows NT only)
- Alternate (or not) between files

In LANDP for AIX, the three SNA sessions can be used simultaneously because LANDP for AIX uses three parallel processes for forwarding.

Host applications

Host applications, to which stored records should be forwarded, have to be associated during customization with the file on which they are stored for forwarding. The host application has to extract the records from the transmission record, analyze the record status, and take appropriate actions. For example, if the record status indicates that the record might be a duplicate (record status P), that is, it has been retransmitted, the record should be ignored if already processed.

For detailed information about forwarding see “Forwarding server” on page 134.

Search operations



The search operations of the store-for-forwarding server are identical to those introduced for the electronic journal server. See “Search operations” on page 126.



DBCS mode restrictions



The store-for-forwarding server requires that the following information contains SBCS characters only:

- Record format names
- Record field names

Forwarding server



The forwarding server complements the store-for-forwarding server and ensures correct transmission of records to the host computer. The forwarding server calls the store-for-forwarding server to retrieve data that is to be sent to the host computer. It also informs the store-for-forwarding server about the current state of the host communication sessions.

In LANDP for DOS, OS/2, and Windows NT, the forwarding server requests services from the SNA server, the shared-file server, and the system manager server. In LANDP for AIX, it requests services from the SNA server and the query server. Also, the LANDP for DOS forwarding server requests services from the ASCII-EBCDIC translation server when operating in DBCS mode.

Each file stored to be forwarded to the host computer is associated with a host computer application to which the file records are sent. Up to 64 files and up to three forwarding sessions can be defined.

Records in different files can be forwarded using the same session.

Transmission

The forwarding server supports up to three simultaneous communication sessions with one or more host computers running CICS/VS or IMS/VS. The session number and session types are defined at customization.

The communication session is always established from the workstation, regardless of the options defined during customization. When the forwarding server is reloaded and detects an existing session, the session is terminated and then reestablished. The BIND image is checked against the values specified during customization.

The forwarding server establishes LU-LU sessions with the host computer applications. The forwarding server uses LU_0 session IDs 37, 38, and 39 for the three reserved sessions. The records are transmitted to the host computer with *single message protocol*. This means that the forwarding server does not expect a host computer application response. That transmission protocol involves use of response protocol, bracket protocol, and chaining. Unsolicited host computer messages sent to one of the reserved forwarding server sessions are discarded.

Record transmission is concurrent with other communication activity. The forwarding server handles session state transitions, and it analyzes the forwarding process state, when a session is interrupted. In addition, the store-for-forwarding server is informed about the communication session status, indicating whether communication with the host computer is available or interrupted.

Communication for all sessions is monitored by the forwarding server, but it is implemented by the SNA server. The forwarding server requests services from the SNA server. For information on the SNA server function calls, refer to the *LANDP Programming Reference*.

If communication is interrupted, the forwarding server automatically resumes operation whenever the link is reestablished. Between the attempts to reopen a host computer session, the forwarding server waits an interval of time that is defined at customization.

The LANDP for AIX forwarding server stops processing only when LANDP is unloaded or when a SIGQUIT signal is received.

During the forwarding process, new records can be stored. Special communication events, such as abnormal transitions, are sent to the system manager server for logging.

Modes of transmission

At customization, the mode in which the forwarding server will perform the record transmission is defined. Two transmission modes are available:

Automatic transmission

Any records found in a store-for-forwarding file are sent to the associated host computer application.

Program initiated transmission

The workstation application explicitly starts and stops the transmission. For those purposes the application must call functions to begin and stop transmission. For information on the store-for-forwarding server function calls, refer to the *LANDP Programming Reference*.

Block transmission

To improve forwarding performance, records to be sent to the host computer can be grouped in blocks. A block can contain records from only one store-for-forwarding file.

At customization, you specify how the records are to be grouped in blocks. Then, if block transmission is chosen, the forwarding server automatically sets the data according to the specifications.

If messages longer than the maximum chain size defined at customization are to be transmitted, they are sent as SNA chained messages.

The block factor is specified during customization. For more information, refer to the *LANDP Installation and Customization*.

Signon feature

Once the host computer session is established and before sending any record to the host computer, the forwarding server can check the signon. The signon message and the expected response are stored in an ASCII text file. For more information, refer to the *LANDP Installation and Customization*.

Transmission priority

To control the order of transmission, some specifications must be provided at customization. The file priority and alternating session transmission can be selected.

forwarding server, OS/2 DOS Windows NT AIX

LANDP for AIX always uses the alternate session option, because there are three parallel processes for forwarding.

Host computer considerations

The forwarding server supports two host computer environments:

- CICS/VS
- IMS/VS

VTAM applications are supported within the SNA protocols supported by these host computer environments.

CICS/VS environment

The characteristics of the CICS/VS environment are as follows:

Workstation type

Workstations must be defined as 3601 LUs.

Transaction mode

CICS input only transactions are supported; conversational transaction processing is not supported.

CICS BMS, Pipeline LU, Inbound FMH, and CICS sign-on Tx

These features are not supported.

IMS/VS environment

The characteristics of the IMS/VS environment are as follows:

Workstation type

Workstations must be defined as UNITYPE=SLUTYPEP.

IMS/VS MFS

IMS basic editing is supported; MFS and MFS DPM are not supported.

Response mode

Terminal response mode is not supported for the records to be forwarded.

Fast path

Fast path transactions are not supported; normal IMS transactions are supported.

Acknowledgement

Optional acknowledgement of recoverable input records (OPTACK option) is supported.

Transaction mode

IMS non-conversational transaction processing is supported; conversational transaction processing is not supported.

Inbound FMH

This feature is not supported.

At customization, you can select data field translation. Then, data fields are translated from ASCII to EBCDIC, using the host computer communication translation tables.

Host computer definitions

Only one-way messages (those that do not result in an application response) can be sent to the host computer by the forwarding server. The host computer application that receives the message must comply with this rule. The CICS/VS and IMS/VS definition for the TRANSACT sent by the forwarding server must correspond to non-response transactions.

DBCS mode restrictions



The forwarding server requires that the following information contains SBCS characters only:

- Signon and Signon completed strings
- CICS/VS and IMS/VS transaction codes (they are always translated without SI/SO)
- Record format names
- Record field names

Shared DOS directory services



The shared DOS directory services enable workstations in the LANDP workgroup to share programs, data files, and applications. For example, by storing an application in a shared DOS directory, only one copy has to be installed in the LANDP workgroup.

The workstations can share data files, such as table files and format files. The workstations can also share fixed disk space. Thus, a workstation with a fixed disk can provide space to other workstations in the LANDP workgroup that have no fixed disk or need more space.

The following restrictions apply when working with the shared DOS directory services:

- The IBM PC LAN Program, the IBM DOS LAN Requester, or the IBM Network DOS Requester cannot be used on the shared DOS directory client workstations.
- Some DOS commands are not available. For information on DOS commands that are not available when working on a network, refer to the appropriate DOS manuals.
- An application running in a LANDP for OS/2 workstation cannot request services from the shared DOS directory. In a mixed environment, LANDP for DOS and OS/2 workstations keep shared data and programs on LANDP for OS/2 workstations. These can then be accessed through the OS/2 LAN requester/server, PC LAN program requester functions, and the DOS LAN Requester.

shared DOS directory services, SHRSHARE utility

Creating a shared DOS directory

To create a shared DOS directory:

1. Make a directory.
2. Place the information to be shared in that directory.
3. Use the SHRSHARE utility to enable that directory to be shared by the workstations in the LANDP workgroup.

For information on how to use the SHRSHARE utility, refer to “SHRSHARE utility.”

Utilities

The shared DOS directory services provide two utility programs:

SHRSHARE utility program

Defines which server disks or directories are to be shared and assigns them a name. This utility must be run on the server workstation.

FILESYS utility program

Defines the drive letter used to access the shared DOS directory services and the name for the disk or directory to be accessed. This utility must be run on the client workstation.

To provide security for the shared DOS directory services, the system manager server can be used to control access to these services. For more information see “System security” on page 141.

Notational conventions

In the command formats below, the following notational conventions apply:

- [] is used to indicate optional parameters
- | is used to indicate choice between two options
- Relevant DOS parameters are only briefly explained.

Refer to the appropriate DOS manuals for detailed information.

SHRSHARE utility

This utility must be used at the server workstation. It defines which server disks or directories will be shared and gives a short name to them. The format of the command used to invoke this utility is:

`SHRSHARE funct parms`

where:

funct Is the utility function that is called. There are six utility functions:

START
STOP
LOGON

shared DOS directory services, SHRSHARE utility

LOGOFF
PAUSE
CONTINUE

parms Are the specific parameters passed for the called function.

Note: If the SHRSHARE command is issued without any function or parameter, it displays a list of the LANDP workgroup directories the server workstation shares with other workstations in the LANDP workgroup.

START function: This function begins the sharing of directories. The format of the command is:

```
SHRSHARE START shortname=d:[\][path] /R|/W computer|* userid|*
```

where:

shortname

Is an equivalent name (alias) for the disk or directory to be shared. It can be up to six characters. The short name of a disk or directory must be known by the user who wants to access that disk or directory. Note that PSWDIR is reserved.

d: Is the drive specifier.

path Is the DOS path to the directory to be shared.

/R Specifies *read only* access.

/W Specifies *read and write* access.

computer

Is the name of the workstation in the LANDP workgroup that is to share the disk or directory of the server workstation.

* Specifies that any workstation can use the disk or directory of the server workstation.

userid Is the name of the user of the workstation that is to share the disk or directory of the server workstation. Note that this parameter is valid only if system security was selected at customization.

* Specifies that any workstation user can use the disk or directory of the server workstation.

Two examples of SHRSHARE START follow:

Example 1:

```
SHRSHARE START dir01=c:\prog\w BB TELLER1
```

This command states that:

- The subdirectory prog in the drive c: can be shared and its short name is dir01.
- The shared DOS directory can be accessed with reading and writing capability (w parameter).

shared DOS directory services, SHRSHARE utility

- The shared DOS directory can be used from the workstation BB.
- The user TELLER1 can use this shared DOS directory.

Example 2:

```
SHRSHARE START diskc=c:\r * *
```

This command states that:

- The drive c: can be shared and its short name is diskc.
- The shared drive can be accessed with read only capability (r parameter).
- The shared drive can be used from all the workstations (first *).
- The shared drive can be used by all the users (second *).

STOP function: This function ends the sharing of directories. Note that it takes effect after SHRDIR has been reloaded. The format of the command is:

```
SHRSHARE STOP shortname
```

where *shortname* is an equivalent name (alias) for the drive or directory which will no longer be shared.

LOGON function: This function enables the user to sign on. Use this function if you chose at customization to have system manager server control. Note that it prompts the user to enter a user ID and a password. The user ID and the password must consist of SBCS characters only. The format of the command is:

```
SHRSHARE LOGON
```

LOGOFF function: This function enables you to sign off from the system manager server. The format of the command is:

```
SHRSHARE LOGOFF
```

PAUSE function: This function temporarily suspends the sharing of directories. The format of the command is:

```
SHRSHARE PAUSE [/f]
```

where *f* is an optional force parameter that causes all open files to be closed without any prior message. If this parameter is not specified and there are open files, on issuing the command, an attention message is displayed and the open files are not closed. In addition, a return code is sent that can be interrogated by the batch subcommands IF and ERRORLEVEL 1.

CONTINUE function: This function resumes the sharing of directories, after this has been temporarily suspended due to a PAUSE command. The format of the command is:

```
SHRSHARE CONTINUE
```

FILESYS utility

This utility must be used at the client workstation. It defines the drive letter used to access the shared DOS directory services and the short name for the disk or directory to be accessed. It is also used to detach a directory.

The format of the command used to invoke this utility is:

```
FILESYS [d:] shrdird shortname | [d:] /D
```

where:

d: Is the drive to be attached. Note that X: is reserved.

shrdird

Is the shared DOS directory client module.

shortname

Is the equivalent name (alias) for the disk or directory to be shared. The short name of a disk or directory must be known by the user who wants to access that disk or directory. This short name is the one defined with the SHRSHARE command.

/D Breaks the attachment of drive d.

Two examples of FILESYS follow:

Example 1:

```
FILESYS G: shrdird DISKC
```

This command states that:

- The user specifies drive G to access the shared DOS directory.
- The user specifies short name DISKC to access a disk or directory.

Example 2:

```
FILESYS G: /D
```

This command states that:

- The drive G: is to be detached.

Protecting your shared files

For file sharing protection purposes, you must load the IBM DOS SHARE program before working with the shared DOS directory services. To load the SHARE program, at the DOS prompt enter:

```
SHARE
```

System security

The system manager server can control the access to the shared DOS directory services. This access control works based upon an authorization level scheme. The access control and its required authorization levels are selected during customization of

shared DOS directory services, FILESYS utility

the system manager server profile. For further information on customizing the system manager server profile, refer to the *LANDP Installation and Customization* manual.

System security for the shared DOS directory services involves:

- Security in the FILESYS utility
- Security in the SHRSHARE utility.

For information on the command formats, refer to “Utilities” on page 138.

SHRSHARE utility security

If you selected system manager server access control, the authorization level required to use the SHRSHARE utility is security level **M** or **A** where:

M Allows a user to access the system manager from a user-written application rather than the system manager operator

A Is the authorization level for a system manager administrator.

If a LOGON function has not been issued, or the user authorization level is neither **M** nor **A**, when invoking the SHRSHARE utility, the following message is displayed on the workstation screen:

```
Access denied
```

FILESYS utility security

If you selected system manager server access control, the authorization level required to use the FILESYS utility is security level **O** or **R**. Thus, only users with authorization level either **O** or **R** can access the shared DOS directories.

If a LOGON function has not been issued, or the user authorization level is neither **O** nor **R**, when entering a FILESYS attaching command, the following message is displayed on the workstation screen:

```
Access denied
```

Unloading the shared DOS directory services at a client workstation

If you want to stop sharing directories from a client workstation, you can unload the shared DOS directory services client portion at that workstation. To do that, enter the following command:

```
SHRDIRDD /U
```

Chapter 7. System management servers

This chapter describes the servers and services that the LANDP family provides to help manage a LANDP system. Included is information about the following servers and services:

System manager server

Provides management and maintenance support for the workgroup.

LANDP for DOS and OS/2 system manager operator

Provides access to the system manager server data and lets one or more users using DOS or OS/2 workstations use the workgroup facilities.

Operator interface

Provides an interface for a DOS workstation operator to interact with a printer manager server, 3270 emulator, or 3287 printer emulator.

Local resource manager server

Lets applications use the LANDP common API to interact with specific LANDP for DOS components installed in the same workstation as the local resource manager.

System manager server



The system manager server provides the following services for all OS/2, DOS, and Windows NT workstations in a LANDP workgroup:

- User identification and control functions
- User session maintenance and control
- Network SignON Coordinator/2 exit routine feature (DOS and OS/2 only)
- User profile and application data maintenance
- Date and time synchronization
- LANDP workgroup common data maintenance
- Retrieval of defined record structures
- Data validation with defined record structures
- Alerts management
- System and user log management
- Operator messages support
- Back-up global data

Only one system manager server can be installed in each LANDP workgroup.

The LANDP common API provides functions that can be used by applications to make requests to the system manager server.

As well as being accessible from OS/2, DOS, and Windows NT, the system manager server can also be accessed through a LANDP for AIX workstation. However, in the case of a LANDP for AIX workstation, this functionality and access control is generally provided by the AIX user maintenance functions. For example, system administrators

system management server, OS/2 DOS Windows NT

can use AIX maintenance functions to define a group of users who have access to applications that use LANDP services.

The system manager server can also be configured to back-up global data during installation and customization. See *LANDP Installation and Customization* for additional information.

User identification and control functions

The system manager server provides functions to control access to different parts of the LANDP components.

This control is provided by signon and signoff functions using a password. Up to six levels of system authorization can be selected to control the access to the system services. Also, up to ten application authorization levels can be selected to control access to the selected functions.

If security is defined, you must sign on before being able to use most of the system manager server functions.

In OS/2, DOS, and Windows NT a user can be logged on to only one workstation at a time. Once a user is logged on to a workstation, no other user can log on to that same workstation.

In the case of AIX, although several users can be logged on to one LANDP for AIX system (which only counts as one workstation in a LANDP workgroup), only one user can be signed on to each LANDP for AIX terminal at a time.

Every new application a user starts can:

- Perform a signon function with the user ID and password supplied by the user. This requires that the user has the appropriate authorization level for the tasks they want to perform.
- Omit the signon function if another application in the same workstation has already performed a signon. The system manager server keeps track of the signed on users and their authorization levels for all the workstations in the LANDP workgroup.

During customization, you can choose the levels of authorization that will be used. If you do not select any access authorization level, no control takes place.

More than one system and application authorization level can be assigned to individual users. With an access authorization scheme, your application and the system manager control all access to the LANDP and application resources.

See *LANDP Installation and Customization* for more information about defining authorization levels.

User session maintenance and control

The system manager includes functions that are used to:

- Get the PC ID and authorization level for a specified userid
- Get a list of the defined users
- Get a list of signed-on users
- Get a list of users that have locked global LAN data, system data, or application data for themselves or for another user
- Send messages to a NetView operator
- Send resolutions to NetView

See *LANDP Programming Reference* for additional information about these functions.

Network SignON Coordinator/2 exit routine feature (OS/2 and DOS only)

The LANDP security system, managed by the system manager server, can use the IBM Network SignON Coordinator/2 exit routine feature to perform multiple signons and signoffs and change passwords on different systems (OS/2 User Profile Management, LAN server, and host). For more information about the exit routines, refer to the online *IBM Network SignON Coordinator/2 User's Guide*.

The exit routine feature must be related to one of the systems defined by the Network SignON Coordinator/2, so that access is not limited to LANDP only. LANDP for DOS and OS/2 provide the following exit routines:

```
..\EHC0500\SMGREXIT.EXE (LANDP for DOS)  
..\EHC0500\SMGREXIT.EXE (LANDP for OS/2)
```

To perform signon, you must be a defined LANDP user with the appropriate authorization levels. The LANDP client can then perform the functions allowed by these levels.

The exit routines and the Network SignON Coordinator/2 must be located on the same workstation. The system manager server can be located on any OS/2, DOS, or Windows NT workstation in the workgroup.

User profile and application data maintenance

Each user has a user profile, which controls access to LANDP services. Also, up to 1KB of retrievable and updatable application data information can be defined for each user. If a record format for application data is defined during customization (using the record definition facility), the system manager server checks that the data supplied in the application data function matches the defined record format. The system manager server provides functions to maintain of the user profile and application data. The initial content of both items is defined during customization. See *LANDP Programming Reference* for more information about these functions. Authorized users can retrieve:

- Data for update from only one user at a time
- The user profile and the application data of the same user

Authorized users can update:

system management server, OS/2 DOS Windows NT

- Their own user profile and application data
- Other user profiles and application data

Date and time synchronization

Synchronization of date and time is available for all workstations in a LANDP workgroup. The date and time is maintained by the operating system on the workstation where the system manager is loaded by updating the operating system date and time in each computer of the LANDP workgroup. The date and time can be synchronized:

- Using a function in any workstation, at any time, to update the operating system date and time from the application.
- Automatically. If this mode is selected during customization, synchronization occurs whenever:
 - The workstation that contains the system manager is started.
 - The date and time has been changed on the workstation where the system manager server is loaded by:
 - The system manager operator
 - The system manager server set date and time function.
 - A LANDP workgroup session is established between the workstation containing the system manager server and another workstation.

Note: If the date and time are changed on the workstation with the system manager server using operating system functions, no automatic synchronization takes place.

For more information about the functions and restrictions, see *LANDP Programming Reference*.

LANDP workgroup common data maintenance

The system manager server provides functions to retrieve and update the system status area. This area contains the data common to all workstations in a LANDP workgroup including:

- Current LANDP workgroup ID
- Log status (active, deactivated, or not defined)
- Indicator of log in wraparound mode
- Alerts status (active, deactivated, or not defined)
- Indicator of alerts file full
- Status of the PU-SSCP session used to send alerts
- Operator message status (active, deactivated, or not defined)
- User ID who receives the operator messages
- Indication whether the system date and time have been set after IPL

The user must be authorized to update system data.

The system manager server also provides functions to retrieve and update the LANDP workgroup common data. Common data is an area of up to 1KB that is shared among all the workstations in a LANDP workgroup. This area contains data defined during customization and updated at run time. If record format for LAN common data was defined during customization, the system manager server checks that the data specified in the update LANDP LAN common data (UD) function matches the structure of FBSSGLUS.

For more information about the functions used to retrieve and update LANDP common data, see *LANDP Programming Reference*.

Retrieving defined record structures

The system manager server provides a function to retrieve record structures defined at customization. The characteristics of record structures are described in *LANDP Installation and Customization*.

The system manager server retrieve function has two options:

- Retrieve only the format attributes of a record definition
- Retrieve a record definition with description of the fields

In both cases the request to the system manager server contains the record name defined during customization.

For more information about retrieving defined record structures, see *LANDP Programming Reference*.

Data validation with defined record structures

The system manager server provides a function to validate data with a defined record structure. You define this record structure during customization.

For more information about validating data with a defined record structure, see *LANDP Programming Reference*.

Alerts management

The system manager server provides functions for problem notification using the SNA generic alerts architecture. A Tivoli NetView licensed program in the host acts as the alert receiver. Every LANDP software component and application in the LANDP workgroup can provide useful problem notification to the network operator in the same way as SNA nodes.

An *alert* is an unsolicited record sent to the network operator by a network component that has detected a problem. The alert provides the following information:

- Identity of the sender
- Problem severity indicator and problem description
- List of probable causes — user causes, install causes, and failure causes, if any
- List of recommended actions for the operator

system management server, OS/2 DOS Windows NT

A *resolution* is an unsolicited record sent to the network operator by a network component that has detected that a problem has been resolved.

A resolution record provides the following information:

- Identity of the resolution sender
- Resolution type and description
- List of actual causes — user causes, install causes, and failure causes, if any
- List of actual actions performed.

The system manager server uses the SNA server to forward the network management data to the management services focal point in the network. This focal point provides centralized network management support for all workstations in a LANDP workgroup.

The system manager supports a user exit to filter or modify generic alerts and resolutions. This can be used to determine which alerts and resolutions are forwarded to NetView, depending on rules chosen by the user.

If required, the alerts are queued for delayed transmission. The alerts can be generated by:

- The supervisor
- LANDP servers
- Servers you have developed
- Applications

See *LANDP Programming Reference* and *LANDP Problem Determination* for additional information.

System and user log management

The system manager server provides functions to store and retrieve the system and user log.

Logging uses standard DOS, OS/2, or Windows NT files. All fields of the record header are ASCII character strings. The format of the other data is not standardized. It need not necessarily be ASCII characters.

The drive type and the file size influence the performance of the log operations. A fixed disk is highly recommended. The minimum file size is 10KB and the maximum 500KB. The default is 100KB.

If the log file does not exist it is created automatically, at loading time, using the size specified during customization. If there is not enough free space in the specified drive, the file is not created and an error code is returned. If the log file already exists, the size and the record length of the actual file is used and customization specifications are ignored.

The log file is created and maintained by the system manager server through the functions provided.

Attention: Do not, under any circumstances, modify the log file directly.

Deleting the log file is allowed only when your LANDP programs are not running.

For more information about these functions, see *LANDP Programming Reference*.

Operator messages support

An operator message is a log file record written with operator message type. For more information on message types, refer to *LANDP Programming Reference*.

The system manager server process the messages received for the operator of messages and informs the system manager operator of their existence. Only the user specified during customization or the one assigned using an update message operator receiver function can read the operator messages. For more information, refer to “System manager operator.”

DBCS mode restrictions

The system manager server requires that the following information contains SBCS characters only:

- User IDs
- Passwords

System manager operator



The system manager operator (SMOP) provides access to system manager server data and provides a way to manage the workgroup facilities.

The system manager operator can be loaded in the same workstation as the system manager server or in any other workstation in the LANDP workgroup.

If a system manager server is installed with the log management option, install the system manager operator to display messages for the operator.

Different kinds of users can use the system manager operator to perform various tasks. For example, adding new users is an administrator’s task while transmission of alerts to the host computer can be controlled by an operator.

An access authorization scheme can control the access to the different functions. User authorization levels are required for working with the system manager operator. Note that users can be assigned more than one authorization level. User authorization levels are assigned when customizing the system manager server or by using the options contained in the system manager operator panels.

To manage the workgroup resources, more than one workstation can have the system manager operator installed, and more than one system manager operator can be used at the same time. The system manager server controls concurrent access to the shared workgroup information.

system manager operator, DOS

Only one of the operators using the system manager operator can receive messages that are received and stored by the system manager server. This operator is defined when customizing the system manager server, but can be changed using the system manager operator. See Customizing the system manager operator.

The LANDP for DOS system manager operator does not accept DBCS data input. However, the LANDP for OS/2 system manager operator accepts DBCS data as input if it has been specified during customization.

Customizing the system manager operator

The following features can be defined when customizing the system manager server:

- A hotkey to invoke the system manager operator (LANDP for DOS only).
- An audio signal and a video signal. The signal informs the operator that there is a message to be displayed. The signal can be audible (beep), visible (special symbol on the bottom right corner of the screen), or both.
- The LAN common data and application data area record definitions. If these definitions are provided, the system manager operator can display formatted information. Otherwise, the information is displayed in hexadecimal format.
- The user who receives the messages for the operator. If only one user is assigned the operator authorization level, this user is the one who receives the messages. If more than one user is assigned the operator authorization level, one of them must be identified as the user who receives the messages. At run time, the authorized operator can change the user identification using the system manager operator.

For information on the customization procedures, refer to *LANDP Installation and Customization*.

Working with the LANDP for DOS system manager operator



The LANDP for DOS system manager operator can be invoked by:

- A hotkey defined at customization
- An application, simulating a soft hotkey.

Attention: When the BIOS video mode differs from the alpha mode numbers 0, 1, 2, 3, or 7, the hotkey call is cancelled and the alarm sounds to alert the workstation operator. The contents of the panel may be lost, if the video mode is 0 or 1 and the page number is different from zero.

When the system manager operator is invoked, the authorization level is verified as follows:

- If no user is signed on, the panel below is displayed, asking for user identification:

```
LAN DISTRIBUTED PLATFORM/DOS
System Manager Operator User Identification
-----

User identification.....:

Password.....:

-----
Enter=Process           F1=Help           Esc/F3=Exit
```

Figure 19. LANDP for DOS SMOP Logon panel

The entered data is sent to the system manager server for verification.

- If a user is already signed on, it is verified that this user has the authorization level **O**, **A**, or **S**, where:
 - O** Is the authorization level for the system manager operator.
 - A** Is the authorization level for a system manager administrator.
 - S** Is the authorization level for access to debug and trace tools (DDT) and log files.

If LANDP for DOS has been customized without an authorization level, the system manager operator logon panel is displayed as well. Then, you have two options:

- Bypass the panel. You must press the Enter key, with no information typed in the **User identification** and **Password** fields.
- Enter a user ID and password. Note that to perform tasks that require user identification, such as displaying messages for the operator, you must enter the user ID and password corresponding to a user assigned at customization.

Access is revoked if you attempt to use the system manager operator without proper authorization, or enter more than three invalid passwords. The unsuccessful user is not

system manager operator, DOS

allowed to enter the system manager operator until an authorized administrator updates the user profile of the rejected user by resetting the **Number of unsuccessful signons** field. The User Profile panels for DOS and OS/2 are shown on pages 159 and 173. For more information see *LANDP Programming Reference*.

Only one user can be signed on per workstation. That user must sign off to enable another user access to the system manager server.

A user can be signed on at only one workstation at a time. They must sign off before being able to sign on at another workstation.

After verification of the access authorization, the system manager operator main menu is displayed:

```

                                LAN DISTRIBUTED PLATFORM/DOS
                                System Manager Operator Options
-----
Select option by entering appropriate number .....:

1. Display system status and enter commands

2. Access LAN common data

3. Access user profile data

4. Browse the system log file

5. Display messages for the operator

-----
F1=Help                               Esc/F3=Return
```

Figure 20. LANDP for DOS SMOP main menu

The main menu enables you to select various system manager operator options. After you select an option, a further panel is displayed, with each panel corresponding to a different option. These panels are described in the following sections.

To exit the main menu, press the Esc or the F3 function key. If you entered the system manager operator via the logon panel, a signoff takes place automatically.

Displaying system status and entering commands

Only users with authorization level **O** can use this option. Enter 1 on the main menu to display the following panel:

```

                                LAN DISTRIBUTED PLATFORM/DOS
                                System Status and Commands
                                -----
LAN identification ...: xxxxxxxx                                PENDING MESSAGES
Operator messages ID ...: xxxxxxxx
Date ...: mm/dd/yy      Time ...: hh/mm/ss      Automatic sync. ...: YES
                                -----
System status ...:
Alerts transmission .....                                On   Off  Not def.
Logging recording .....                                x
Messages for the operator.....                            x
Video signal .....                                    x
Audio signal .....                                      x

Command ...: xx                                Parameter ...:

                                -----
Enter=Enter command      Enter=Refresh status      F1=Help
Esc/F3=Main menu
    
```

Figure 21. LANDP for DOS SMOP System Status and Commands panel

This panel shows the current status of those activities controlled by the system manager server.

The **Operator Messages ID** entry defines the user ID that can receive and view the operator messages. The date format depends on the country.

If a message for the operator is pending, a highlighted reminder appears on the upper right-hand corner. There is also an indicator for automatic synchronization selection; note that the synchronization mode is selected at customization.

The video signal and the audio signal are set during customization; they cannot be modified by issuing a command here. A command field is provided to change system status and data.

Note: Commands *must* be entered in upper case.

The following function keys are available:

Enter Refreshes status, provided no command is typed in the command field. If a command is typed there, the system enters the command and refreshes the status.

system manager operator, DOS

Esc/F3 Returns to the main menu.

F1 Provides help. The help panels show the available commands and their parameters. These commands are shown in the table below with their required parameters, where applicable.

Table 29 (Page 1 of 2). Commands for Changing System Status and Data

Command	Meaning	Action sequence
AA	Activate alerts	<ul style="list-style-type: none"> This command starts the transmission of alerts to the host computer.
AL	Activate logging	<ul style="list-style-type: none"> This command enables recording of messages that are sent to be logged.
AM	Activate messages to workstation operator	<ul style="list-style-type: none"> This command enables recording of messages for the workstation operator.
SA	Stop alerts	<ul style="list-style-type: none"> This command stops the transmission of alerts to the host computer. After issuing an SA command, and before issuing an AA command, alerts are discarded.
SL	Stop logging	<ul style="list-style-type: none"> This command stops recording messages in the log file, including messages to the workstation operator. After issuing an SL command, and before issuing an AL command, messages for the log file are discarded.
SM	Stop messages to workstation operator	<ul style="list-style-type: none"> This command stops operator notification, when a message for the workstation operator is received. After issuing an SM command, and before issuing an AM command, the workstation operator is not notified of messages received for the operator.
UD	Update date and time	<ul style="list-style-type: none"> This command changes the <i>date and time</i> of the system, which are the required parameters. If automatic synchronization was selected during customization, the system manager server sends the new date and time to all workstations in the LANDP workgroup.
UI	Update LAN identification	<ul style="list-style-type: none"> This command changes the <i>LAN identification</i>, which is the required parameter. The LAN identification is used by the LANDP NMVTs as the service point and identifies the LAN in the host computer. To identify in the host computer where the alerts come from, it is recommended that you assign the LANDP SNA PU name to the LAN ID in the host computer configuration.
UO	Update message operator identification	<ul style="list-style-type: none"> This command changes the <i>message operator identification</i>, which is the required parameter. It must be issued by the message operator user.

Table 29 (Page 2 of 2). Commands for Changing System Status and Data		
Command	Meaning	Action sequence
UT	Update time	<ul style="list-style-type: none"> This command changes the <i>time</i> of the system, which is the required parameter. If automatic synchronization was selected during customization, the system manager server sends the new time to all workstations in the LANDP workgroup.

Accessing LAN common data

Only users with authorization level **O** can use this option. Enter 2 on the main menu to display a LAN Common Data panel, of which the following is an example:

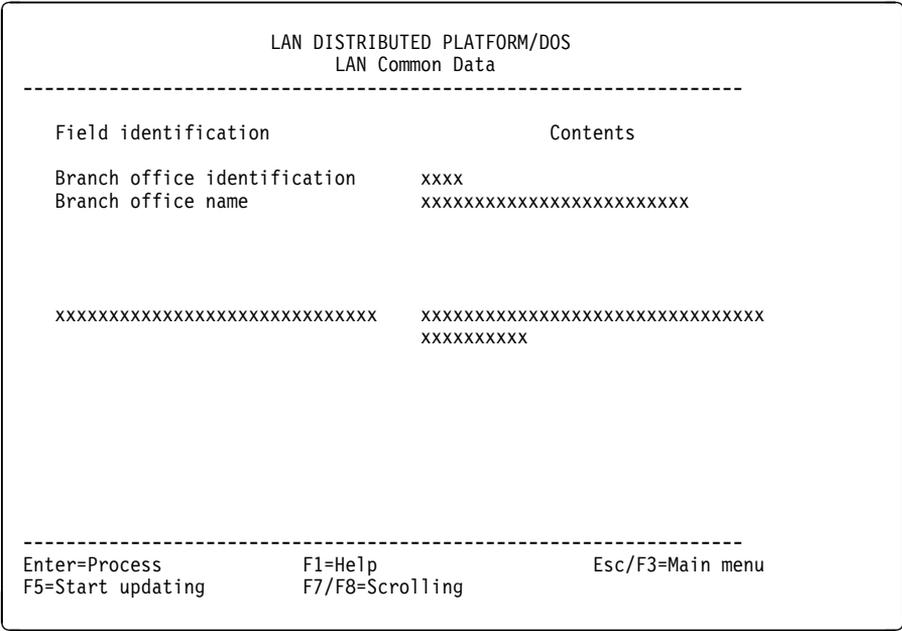


Figure 22. LANDP for DOS SMOP LAN Common Data panel

This panel shows the current common data maintained by the system manager server and used by applications.

The LAN common data area contains information that can be shared by all applications installed in the LANDP workgroup. That data is formatted according to the record format FBSSGLUS, defined at customization. If the record format is not defined, the data is shown in hexadecimal format. The size reserved for this area is up to 1024 bytes.

system manager operator, DOS

The **Field identification** items, which precede the **Contents** information, correspond to the field descriptions entered for the record format definitions, at customization. Note that the **Field identification** contents cannot be changed by the workstation operator on this panel. Numeric and alphanumeric fields are shown divided into as many groups of 40 bytes per line as their length requires. Hexadecimal fields are presented in as many half-bytes as their record format length needs, two characters per byte from X'00' to X'FF'.

The following function keys are available:

Enter	Refreshes or updates panel modifications after F5 has been pressed.
F1	Provides help.
Esc/F3	Returns to the main menu.
F5	Starts data updating process.
F7/F8	Scrolls the user information.

To *display* the LAN common data, only the F7 function key, to scroll backwards, and the F8 function key, to scroll forwards, are required.

To *update* the LAN common data, you should follow the steps below.

1. Scroll to the data to be updated.
2. Press F5 to enable data updating.
3. Modify data on the panel.
4. Scroll to other data to be updated, and modify it.
5. Press the Enter key to perform the update, when there is no more data to be modified.

The data is validated according to the record format defined at customization. If a function key other than Enter is pressed, no change is made to the data.

Accessing user profile data

Authorization level **A** is required to display, update, add, or delete users. System authorization levels can be modified, but there must be at least one user with authorization level **A**.

Enter 3 on the main menu to display the following panel:

```

LAN DISTRIBUTED PLATFORM/DOS
User Profile : System Data
-----

Field Identification                Contents

User identification                xxxxxxxx
Password
User name                          xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Last password change on:          yy/mm/dd  hh:mm:ss
Language identification            xx
System authorization level         xxxxx
Appl. authorization level         xxxxx
Number of unsuccessful signons    xxxxx
Date/time of last signon          yy/mm/dd  hh:mm:ss
Date/time of last signoff        yy/mm/dd  hh:mm:ss
Type of signoff

-----
Enter=Process F1=Help F2=Search user Esc/F3=Main menu F4=Add user
F5=Start updating F6=Delete user F7/F8=Scrolling users F9=Prev. panel
    
```

Figure 23. LANDP for DOS SMOP User Profile: System Data panel

This panel shows the user profile data. The first panel displayed corresponds to the first user recorded. There can be more than one panel per user. To display the next panel, press the Enter key. To perform the functions available for the user profile data, see page 161.

The following function keys are available:

- Enter** Updates panel modifications after pressing F5, adds a new user, and shows the next panel of the same user.
- F1** Provides help.
- F2** Searches for a user.
- Esc/F3** Returns to the main menu.
- F4** Adds a new user.
- F5** Updates a user profile.
- F6** Deletes a user.
- F7/F8** Scrolls through information for all users defined in the profile.
- F9** Returns to previous panel.

system manager operator, DOS

Once the last **User Profile : System Data** panel is displayed, pressing the Enter key displays a panel, of which the following is an example:

LAN DISTRIBUTED PLATFORM/DOS	
User Profile : Application Data	
Field Identification	Contents
Operation counter	xxxxx
Total savings	xxxxxxxxx
Cash box amount	xxxxxxxxx

Enter=Process F1=Help F2=Search user Esc/F3=Main menu F4=Add user
F5=Start updating F6=Delete user F7/F8=Scrolling users F9=Prev. panel

Figure 24. LANDP for DOS SMOP User Profile: Application Data panel

This panel shows information associated with each user. The data is formatted according to the record format FBSSUSPR, defined at customization. If the record format is not defined, the data is shown in hexadecimal format. The size of this area is up to 1024 bytes.

The **Field identification** items, which precede the **Contents** information, correspond to the field descriptions entered for the record format definitions, at customization. Note that the **Field identification** contents cannot be changed by the workstation operator on this panel. Numeric and alphanumeric fields are shown divided in as many groups of 40 bytes per line as their length requires. Hexadecimal fields are presented into as many half-bytes as their record format length needs, two characters per byte from X'00' to X'FF'.

Only the contents of the fields can be changed. Before changing information, note the following:

- An administrator can add as many new users as required. The same user can be assigned more than one authorization level.
- If the user is logged on, only the administrator's data in the profile record can be updated. If that data has been updated, the administrator must first log off, then log on again to refresh data for the servers.

- The user profile data of a logged-on user cannot be deleted. Hence, the user profile data of the active administrator cannot be deleted.

The following functions are available for both the user profile and application profile data.

Search

1. Press F2 to enable data entry.
2. Enter the user identification required.
3. Press the Enter key to access the user profile data.

Add

1. Press F4 to access the panel to add a new user.
2. Enter the user information.
3. Press the Enter key to get the next panel for the user.

When the last panel is reached, press Enter to add the user and validate the data. If you press any other function key, the adding process is not completed and an attention message appears.

To add another new user, press F4 again and repeat the process. The administrator can add as many new users as necessary.

Update

1. Scroll to the user profile data to be updated, or access it directly using the Search function (F2).
2. Press F5 to start data update.
3. Modify data on the panel.
4. Press the Enter key to continue updating a user's information on the next panel. When the last panel of a particular user is reached, a message appears on the screen, telling you that you have reached the last panel for this particular user. If you press Enter at this point, all changes are validated and recorded on the system. If any other function key is pressed, no data is changed.
5. Repeat the previous two steps as many times as necessary.

Except for the administrator's data, no information can be updated in a profile record while the user is logged on. If the administrator is logged on, a signoff and then a signon is necessary to refresh the user data.

Delete

1. Scroll to the user to be deleted, or access it directly using the Search function (F2).
2. Press F6 twice to delete the user.

The user you are deleting must be signed off when the deletion is performed. An active administrator cannot be deleted.

Browsing the system log file

Only users with authorization level **O** or **S** can select this option. Enter 4 on the main menu to display the following panel:

```
LAN DISTRIBUTED PLATFORM/DOS
System Log File : Search Criteria
-----

Display order ..... x

Search Fields                               Values / Range

Sequence number .....                    xxxx
Date .....                               yy/mm/dd   yy/mm/dd
Time .....                               hh/mm/ss   hh/mm/ss
Bucket .....                             x
PC identification .....                  xx
Message type .....                       x
Service .....                            xxxxxxxx
Device identifier .....                   xxx

-----
Enter=Process           F1=Help           Esc/F3=Main menu
```

Figure 25. LANDP for DOS SMOP System Log File: Search Criteria panel

This panel displays the contents of the log file. If the log file contains DBCS data, it is not displayed when working in SBCS mode.

Two values are available for the **Display Order**:

- 1 From the oldest record to the newest one.
- 2 From the newest record to the oldest one.

Most fields in the log record heading can be used as a search argument. Simple AND operations are available to combine some of these fields. If no argument is entered, all the log records are shown.

The following function keys are available:

- Enter** Validates and processes search criteria.
- F1** Provides help.
- Esc/F3** Returns to the main menu.

The records are displayed as shown in the following panel:

```

LAN DISTRIBUTED PLATFORM/DOS
System Log File : Browse
-----
Num.  Date  Time  Buc.  Type  WS  Requester  Device  Event  Retcode
nnnn yymmdd hhmss  x   x   xx  xxxxxxxx   xxx   xx   xxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxx

nnnn yymmdd hhmss  x   x   xx  xxxxxxxx   xxx   xx   xxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Search sequence number ..... xxxx

-----
Enter=Process          F1=Help              Esc/F3=Previous panel
F7/F8=Scrolling       F9=Hex/ASCII

```

Figure 26. LANDP for DOS SMOP System Log File: Browse panel

The formatted header and the message appear as they are stored in the log file. Pressing F9 enables you to display messages in hexadecimal.

The number of lines per record depends on the length defined at customization. The maximum length is 1024 bytes.

You can enter a record number for the **Search sequence number**. The system manager operator searches for that record in the log file and, if it exists, displays it first.

The following function keys are available:

- Enter** Displays data related to the search sequence number.
- F1** Provides help.
- Esc/F3** Returns to the previous panel.
- F7** Displays the previous record in the file according to the search criteria and display order.
- F8** Displays the next record in the file according to the search criteria and display order.
- F9** Displays the message alternately in hexadecimal or ASCII.

The following function keys are available:

- F1** Provides help.
- Esc/F3** Returns to the main menu.
- F7/F8** Scrolls the operator messages.

Once a message is deleted from the screen, it cannot be displayed again. However, the message is written in the log file, and you can read it by using the **Browse the system log file** option in the main menu.

Working with the LANDP for OS/2 system manager operator



The LANDP for OS/2 system manager operator can be invoked by:

- An application. You can use the techniques provided by IBM OS/2 to run an application from another application, because the system manager operator can be regarded as an application that can be called from another application. Because the IBM OS/2 process identifications (PIDs) of the applications are different, the system manager server can distinguish between them, and both applications can request services from the system manager server in parallel.
- A command issued from an IBM OS/2 full screen.

When the system manager operator is invoked, the authorization level is verified as follows:

- If no user is signed on, the window below is displayed, asking for user identification:

Figure 28. LANDP for OS/2 SMOP Logon window

Three pushbuttons are available:

- Enter** Processes identification data.
- Cancel** Leaves the system manager operator.
- Help** Provides help. A help window is displayed to explain the fields on the Logon window.

system manager operator, OS/2

Any data you enter is sent to the system manager server for verification.

- If a user is already signed on, it is verified that this user has the authorization level **O**, **A**, or **S** where:
 - O** Is the authorization level for the system manager operator.
 - A** Is the authorization level for a system manager administrator.
 - S** Is the authorization level for access to debug and trace tools (DDT) and log files.

If LANDP for OS/2 has been customized without authorization level, the System Manager Operator Logon window is displayed as above. Then, you have two options:

- Bypass the window. You must click on the **Enter** pushbutton, with no information typed in the **User ID** and **Password** fields.
- Enter a user ID and password. Note that to perform tasks that require user identification, such as displaying messages for the operator, you must enter the user ID and password corresponding to a user assigned at customization.

Access is revoked if you attempt to use the system manager operator without proper authorization, or enter more than three invalid passwords. The unsuccessful user is not allowed to access the system manager operator until an authorized administrator updates the user profile of the rejected user, resetting the **Number of unsuccessful signon** field.

Only one user can be signed on at one workstation. They must sign off to enable other users access to the system manager server.

A user can be signed on at only one workstation at the same time. It must be signed off to enable its signon at another workstation.

After verification of the access authorization the main System Manager Operator window is displayed:

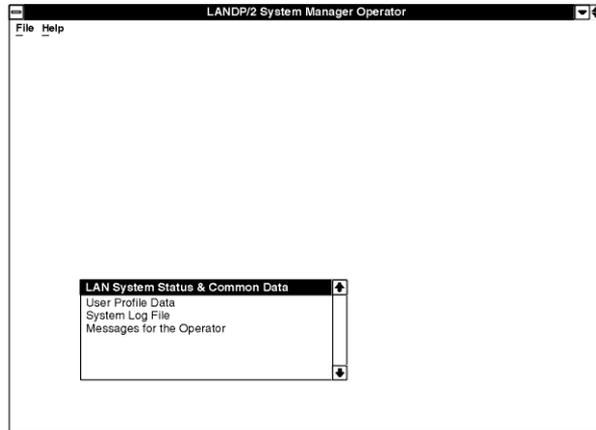


Figure 29. LANDP for OS/2 SMOP main window

From the list box on the main window, you can select four system manager operator options, each of which displays a further window. More than one window can be displayed at the same time, but only one can be active. The windows corresponding to the different choices are described in the following sections.

To exit from the main window, select **File** on the action bar to display a pull-down, then select **Exit** from the pull-down. You can also select the **Close** action in the system menu pull-down. When you leave the main menu, all the displayed windows are removed. If you entered the system manager operator through the Logon window, a signoff takes place automatically.

LAN system status and common data

Only users with authorization level **O** can make this selection. When you select **LAN System Status & Common Data**, the following window is displayed:

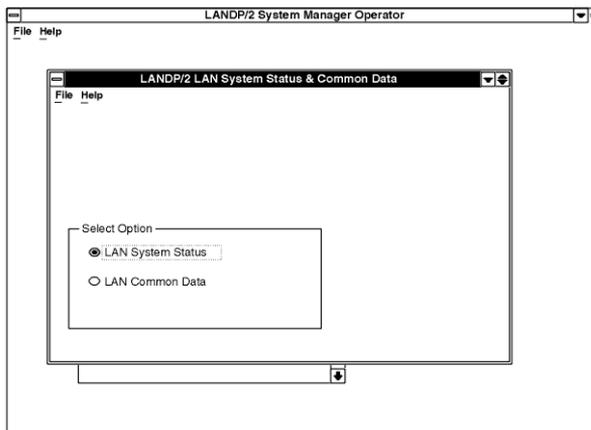


Figure 30. LANDP for OS/2 SMOP LAN System Status and Common Data window

After selecting **LAN System Status** or **LAN Common Data** from the group box, select **File** in the action bar to display a pull-down:

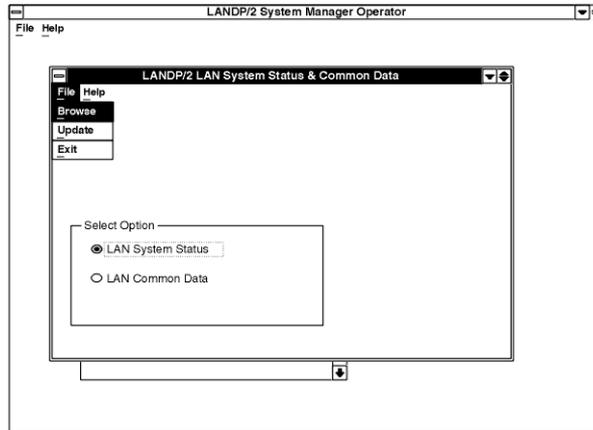


Figure 31. LANDP for OS/2 SMOP LAN System Status and Common Data, File pull-down

The pull-down contains the following actions:

- Browse** Displays information, but does not permit change.
- Update** Displays information and permits change.
- Exit** Leaves the LAN System Status & Common Data window.

system manager operator, OS/2

If you select **LAN System Status** in the group box, and either **Browse** or **Update** from the pull-down, the following window is displayed:

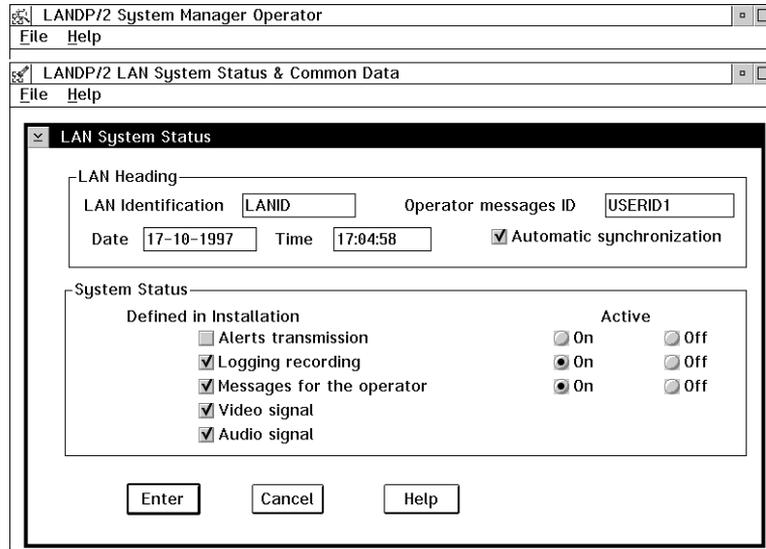


Figure 32. LANDP for OS/2 SMOP LAN System Status window

The **LAN Heading** states the user who receives the messages, in the **Operator Messages ID** field. This specification was made at customization, but it can be changed using this window. The change must be made by the current message operator.

The **System Status** items are selected during customization. You can activate and deactivate the first three items by selecting **On** and **Off**.

User profile data

Authorization level **A** is required to display, update, add, or delete users. The system authorization level can be modified, but at least one user with authorization level **A** must remain.

When you select **User Profile Data** from the list box on the main System Operator Manager window, the following window is displayed:

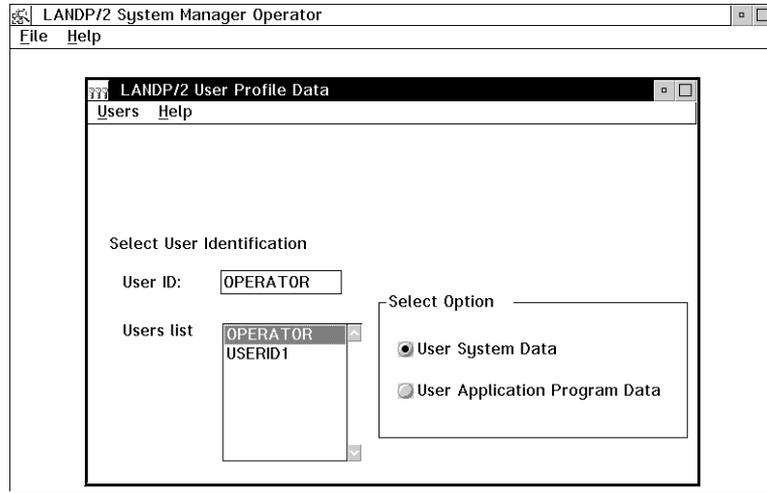


Figure 34. LANDP for OS/2 SMOP User Profile Data window

A group box containing two options, **User System Data** and **User Application Program Data** is displayed.

After selecting one of them, select **Users** on the action bar to display a pull-down:

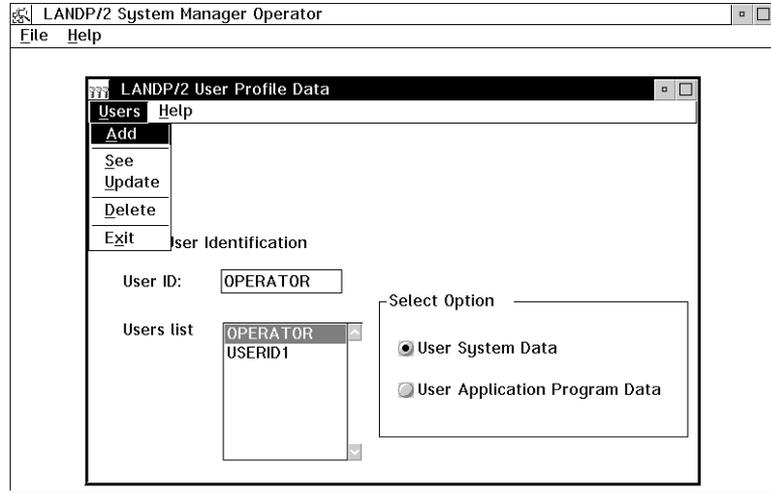


Figure 35. LANDP for OS/2 SMOP User Profile Data, Users pull-down

If you select **User System Data** in the group box, and either **Add**, **See**, or **Update**, from the pull-down, the following window is displayed:

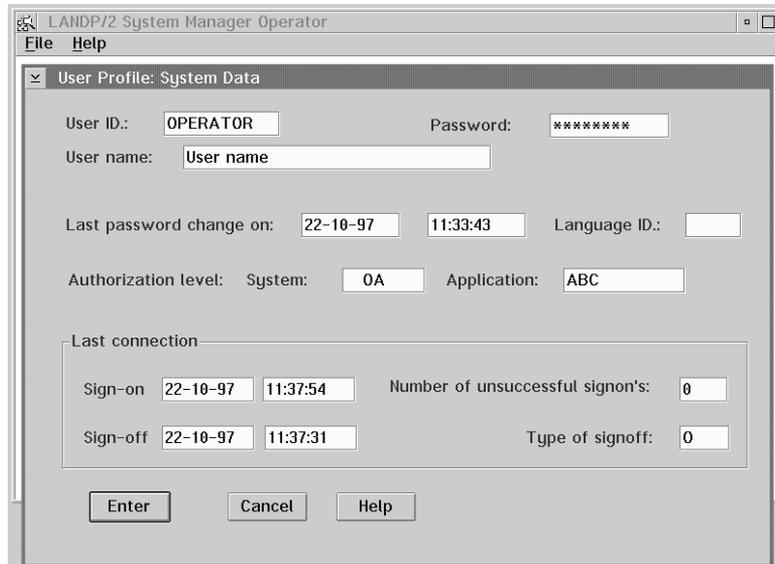


Figure 36. LANDP for OS/2 SMOP User Profile: System Data window

Although the window is similar for the three actions, the information presented depends on the action selected:

system manager operator, OS/2

- Add** Displays blank entry fields so that the administrator can enter a new user.
- See** Shows information about the user selected in the LANDP User Profile Data panel.
- Update** Allows information to be changed by the administrator.

The system data format is the same for all users. Some system data, although automatically recorded by the system manager server, can be changed by an authorized operator, for example, the **Number of unsuccessful signons** field when a user has been revoked due to the number of unsuccessful retries.

Before changing information, note the following:

- An administrator can add as many new users as required. The same user can be assigned more than one authorization level.
- Only the administrator data in the profile record can be updated, if the owner is currently signed on. The updated data takes effect after next signon.
- If the user is logged on, the data cannot be deleted. The user must be logged off at deletion moment; thus, the active administrator cannot be deleted.

If you select **User System Data** in the group box and **Delete** from the pull-down, the following pop-up window is displayed asking for confirmation:

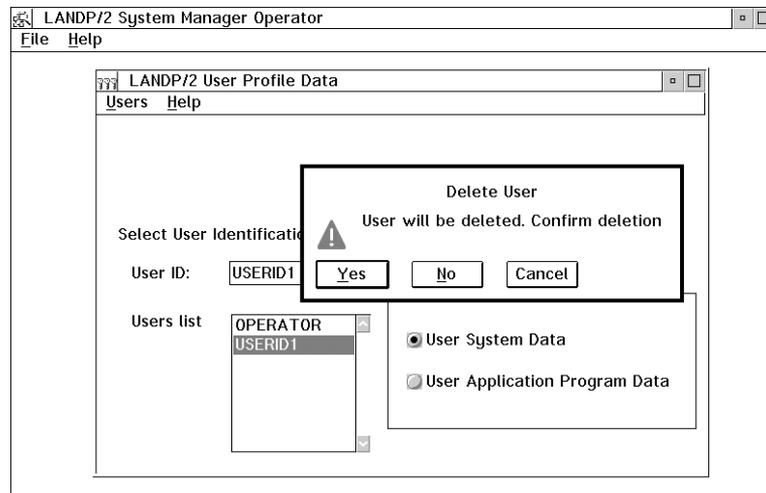


Figure 37. LANDP for OS/2 SMOP Delete User pop-up window

If you select the **Yes** pushbutton, the user is deleted.

If you select **User Application Program Data** in the group box, and either **Add**, **See**, or **Update** from the pull-down, the following window is displayed:

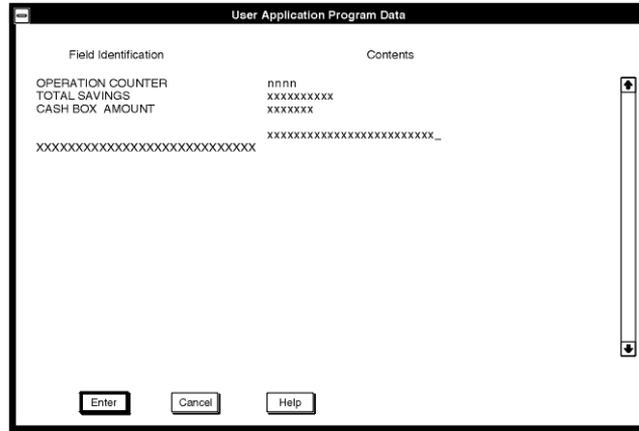


Figure 38. LANDP for OS/2 SMOP User Application Program Data window

The application data depends on the user. The data is formatted according to the record format FBSSUSPR, defined at customization. If the record format is not defined, the data is shown in hexadecimal format. The size reserved for this area is 1024 bytes.

The information under **Field Identification** corresponds to information defined at customization. It cannot be changed here by the operator.

Numeric and alphanumeric fields are shown divided into as many lines as their length requires. To display more information, you use the scroll bars. Hexadecimal fields are presented in as many half-bytes as their record format length needs, that is, two characters per byte from X'00' to X'FF'.

System log file

Only users with authorization levels of **O** or **S** can select this option, which displays the contents of the log file. If the log file contains DBCS data, it will not be displayed when working in SBCS mode.

When you select **System Log File** from the list box on the main System Manager Operator window, the following window is displayed:

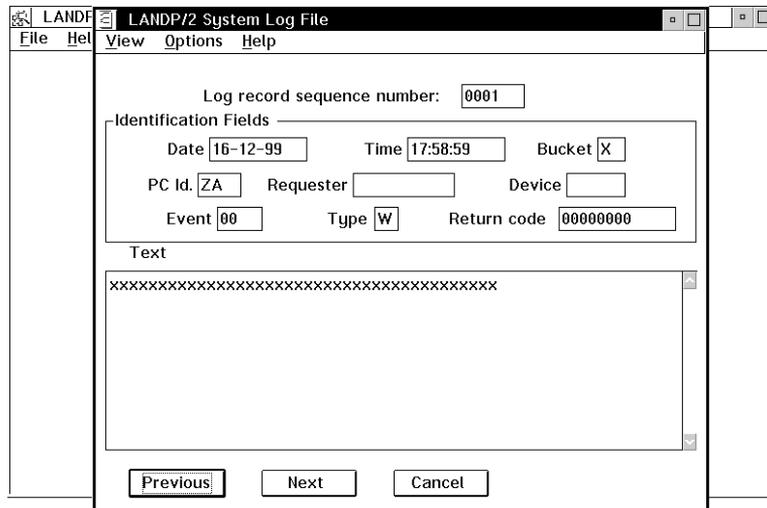


Figure 39. LANDP for OS/2 SMOP System Log File window

The display order of the records is determined by the sequence number that is part of the record, and that is assigned by the system manager server. Two pushbuttons are available to choose the display order:

Next From the oldest record to the newest one.

Previous From the newest record to the oldest one.

system manager operator, OS/2

If you select **Search item** from the pull-down, an entry window is displayed in which you can specify the sequence number of a particular record:

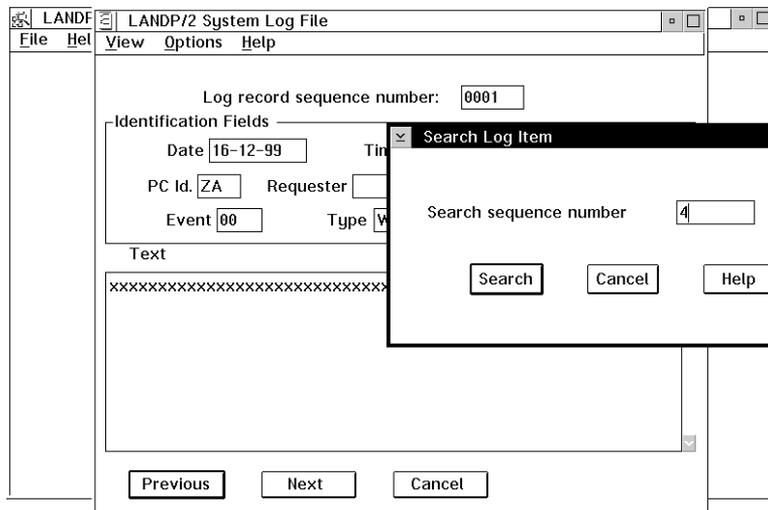


Figure 41. LANDP for OS/2 SMOP Search Log Item entry window

If you specify 0000, the oldest record is searched for, regardless of the sequence number of that record. In the same way, if you specify FFFF, the newest record is searched for.

If you select **Range criteria** from the pull-down, an entry window is displayed in which you can specify the criteria for searching in the log file:

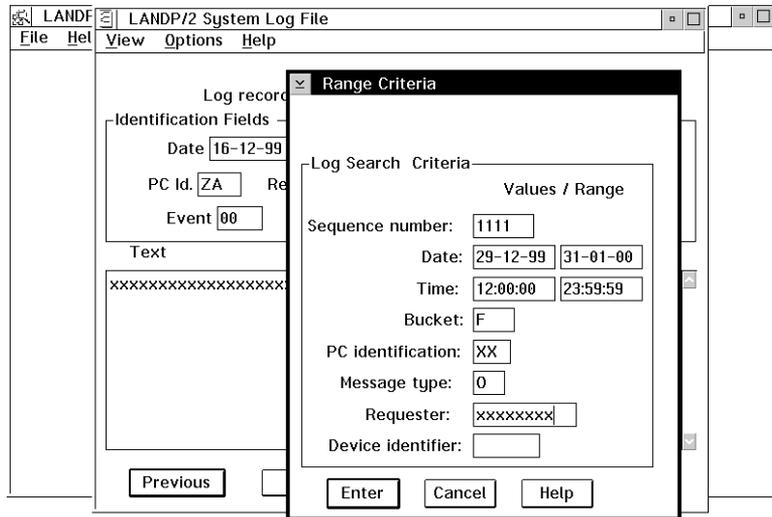


Figure 42. LANDP for OS/2 SMOP Log Search Range Criteria entry window

Most fields in the log record heading can be used as a search argument. Simple AND operations are available to combine some of those fields. If no argument is entered, all log records are shown. When a criterion is entered, only those records that match it are shown when the **Next** and the **Previous** pushbuttons are selected.

Operator interface



The operator interface is an optional component of LANDP for DOS that allows the workstation operator to interact with the following LANDP for DOS components:

- Printer manager server
- 3270 emulator
- 3287 printer emulator

When the IBM Financial Branch System Integrator (FBSI) is installed in the workstation, the operator interface can interact with the FBSI banking printer program (BPP).

The operator interface provides the same services as the local resource manager server. Only one of these two LANDP for DOS components can be loaded in any single workstation. The decision of which to use is made during customization.

The operator interface is used to display information and to issue commands. If selected, define the following during customization:

- A hotkey to enter the operator interface
- Host computer or workstation application IDs that allow the workstation operator to recognize the different emulator sessions.

The operator interface does not accept DBCS data input.

Activating the operator interface

The operator interface is started by the client/server mechanism when:

- The application issues a simulate hotkey (EE) supervisor function call
or
- The workstation operator presses the hotkey defined at customization for that purpose.

The workstation operator ends the operator interface by pressing the F3 function key. Control is then returned to the task that was being performed when the operator interface was started.

Attention: When the BIOS video mode differs from the alpha mode numbers 0, 1, 2, 3, or 7, the hotkey call is cancelled and the alarm sounds to alert the workstation operator. The contents of the panel may be lost if the video mode is 0 or 1 and the page number is different from zero.

Messages and return codes

The operator interface provides panels where you can enter commands and their corresponding parameters. The available commands are listed and explained in “Command set” on page 185.

If an error occurs after you issued a command, one of the following messages is displayed:

- Status not compatible with the command.
- Emulator service is not defined. Check the configuration record.
- Emulator service is not loaded. Check the initialization procedure.
- Nonrecoverable local error. Restart the workstation.
- Program error. Invalid length or format.
- Program error. Invalid command.

In addition, two return codes are supported:

yyyyyyyy.(zz) Service error. (zz) is a return code from the component yyyyyyyy.

(xx) Program error. It means program malfunction in the operator interface. Please, notify your IBM representative about the return code.

Presentation panels

The operator interface provides two kinds of panels:

- Status and Command
- Help

On the *Status and Command* panels, information messages are displayed relating to the status of the features involved. Which features are involved depends on the component accessed through the operator interface. A status message list is presented in “Status messages.”

The panels also provide input fields to enter commands. The available commands, along with their resulting action sequences, are shown in “Command set” on page 185.

At the bottom of the panels, the available function keys are displayed. For more information see “Function keys” on page 189.

To see examples of sample Status and Command panels, see “Sample panels” on page 190.

On the *help* panels, information on the available commands and their parameters is provided.

Status messages

The status messages which appear on the status and message panels are listed below. Note that on the emulator panels the operator interface displays the status of the emulators in the following order:

1. All 3270 emulators
2. 3287 printer emulator
3. All banking printer programs (BPP)

operator interface, DOS

On the printer manager server panels the operator interface displays the status of the printers in the following order:

1. First parallel port
2. Second parallel port
3. Third parallel port

The status messages list follows:

- **Mode:**
 - SUSPEND
 - SUSPEND FORCE
 - LISTING HELD
- **Communication status:**
 - Host not connected
 - Host connected
 - In session
 - In session processing
 - Session ended automatically (X.25)
 - Session lost
 - Session ended
 - Session forced to end
- **Printer status:**
 - Remote
 - Remote force
 - Local
 - Local force
 - First request

 - Attending this service
 - Not attending this service

 - Available
 - Adapter not installed
 - Printer not installed
 - Printer powered off
 - Printer not on line
 - Printer out of paper
 - Only known when used by this service
 - Printer busy

 - Alternate use by
 - Exclusive use by
 - Exclusive mode for pending
- **Current listing options:**
 - Operation intervention active
 - End of listing is under application control
 - Waiting for the application to end the listing

Command set

The operator interface provides commands which are issued through the status and command panels. The commands are usually performed immediately; if this is not possible, an explanatory message is displayed. When a command is pending, *force* commands (those with an **F** in parentheses) and the Enter key to refresh status (that is, with no command entered) are accepted.

The table below shows the set of available commands, along with their meanings and their resulting action sequences.

Note that not all the commands apply to all the LANDP for DOS components or FBSI program components that are accessible through the operator interface:

- The commands **SA** and **SE** can be used for the printer manager server
- The command **EC** can be used exclusively for the 3270 emulator
- The commands **IS**, **ES**, **AS**, **SS**, **EL**, **AI**, **EI**, **CP**, **CA**, **CM**, and **CC** can be used for the 3287 printer emulator
- The commands **HP**, **LP**, and **CC** can be used for the BPP, when FBSI is installed in the workstation

Table 30 (Page 1 of 5). Operator interface commands

Command	Meaning	Action sequence
AI	Activate operator intervention.	<ul style="list-style-type: none"> • The workstation operator is prompted to enable listing printing. When a listing should be started, the message LISTING HELD appears in the heading of the operator interface panel. • To start the listing, the AS command is required. • This command is allowed in any status.
AS	Activate service.	<ul style="list-style-type: none"> • LU operation resumes after a previous SS, SS(F), or AI command. • This command can be used if the status is: SUSPEND SUSPEND FORCE LISTING HELD
CA n	Change the printer assignment of the LU in the operator interface panel. The value of n can be 1, 2, or 3.	<ul style="list-style-type: none"> • The printer assigned to the LU is changed. The new printer must have the adapter installed. • If the LU is In session processing status, this command is not available.
CA(P) n	Fix a new printer assignment. The value of n can be 1, 2, or 3.	<ul style="list-style-type: none"> • A new printer assignment is updated in the operator interface configuration file. The OPE.CFG file must be available when this command is entered. If not available, an error occurs. If LANDP for DOS is installed on a fixed disk, no error should occur. • This command has the same requirements as the CA command.
CC	Cancel pending commands.	<ul style="list-style-type: none"> • Pending ES, SS, HP, or LP commands are canceled. CC command must be issued before end of processing.

operator interface, DOS

Table 30 (Page 2 of 5). Operator interface commands

Command	Meaning	Action sequence
CM n	Change the LU print density mode. The value of n can be N (normal), M (medium), or C (condensed).	<ul style="list-style-type: none"> The LU print density mode changes. The available modes are: normal mode (10 cpi), medium mode (12 cpi), and condensed mode (17 cpi). The print density mode is updated in the operator interface configuration file. The OPE.CFG file must be available when this command is typed. If not available, an error occurs. If LANDP for DOS is installed on a fixed disk, no error should occur.
CP n	Change LU priority. The value of n can be 2, 3, 4, or 5.	<ul style="list-style-type: none"> The LU priorities in the printer queues are changed. The LU that is In session processing status currently, has the highest priority, priority 1, which cannot be changed. Priorities for inactive LUs, priority 0, cannot be changed, either. All priorities between the old and the new priority are shifted. The new priority cannot exceed the number of LUs assigned to the printer, and the number of LUs with listings to be printed. Printers in SUSPEND status have no priority.
EC	End connection.	<ul style="list-style-type: none"> Connection with the host computer ends. It is especially useful for X.25 networks. A new connection is established when the operator presses the emulator hotkey. This command can be used if the status is: Host connected
EI	End operator intervention.	<ul style="list-style-type: none"> Operator intervention condition is deactivated.
EL	End listing.	<ul style="list-style-type: none"> A listing is ended, provided the status is Waiting for the application end the listing. This means that the 3287 printer emulator is in one of the following status: <ul style="list-style-type: none"> The first line in the listing contains <i>LIST=nnnnnnnn</i>, indicating that the application will send <i>ENDL=nnnnnnnn</i> as the last sentence An <i>end bracket</i> was received The normal end of listing timer elapsed. This command results in simulating an <i>end of listing</i> condition, thus allowing another LU to access the same printer.

Table 30 (Page 3 of 5). Operator interface commands

Command	Meaning	Action sequence
ES	End session after the end of the current listing.	<ul style="list-style-type: none"> The host computer ends the session because of a <i>power off</i> and UNBIND request reception, when the end of the current listing is reached. If no printing process is active, the request receives immediate attention. This command can be used if the status is: <ul style="list-style-type: none"> In session In session processing Session lost <p>Note that, if the status is <i>Session lost</i>, nothing is sent to the host computer.</p>
ES(F)	End session immediately.	<ul style="list-style-type: none"> The host computer ends the session immediately because of a <i>power off</i> and UNBIND request reception. This command can be used if the status is: <ul style="list-style-type: none"> In session In session processing Session lost <p>Note that, if the status is <i>Session lost</i>, nothing is sent to the host computer.</p> <ul style="list-style-type: none"> It can also be used if an ES command is pending.
HP	Change the printer to remote mode when the current listing ends.	<ul style="list-style-type: none"> The printer is set to remote mode. If the printer is in local mode, the command takes effect when the current transaction ends. This command can be used in any communication status. This command can be used if the printer status is: <ul style="list-style-type: none"> Local Local force First request
HP(F)	Change the printer to remote mode immediately.	<ul style="list-style-type: none"> The printer is set to remote mode. If the printer is in local mode, the command takes effect immediately and any transaction in process is interrupted. This command can be used in any communication status. This command can be used if the printer status is: <ul style="list-style-type: none"> Local Local force First request <ul style="list-style-type: none"> It can also be used if an HP command is pending.

operator interface, DOS

Table 30 (Page 4 of 5). Operator interface commands

Command	Meaning	Action sequence
IS	Initiate session.	<ul style="list-style-type: none"> The host computer reestablishes the session because of a <i>notify power on</i> request reception. This command must be used after an ES or an ES(F) command. This command can be used if the status is: <ul style="list-style-type: none"> Host connected Session ended automatically (X.25) Session lost Session ended Session forced to end
LP	Change the printer to local mode when the current listing ends.	<ul style="list-style-type: none"> The printer is set to local mode. If the communication status is In session processing and the mode is not SUSPEND, the current listing is finished. This command allows for local printing until the next HP or HP(F) command takes effect. This command can be used if the printer status is: <ul style="list-style-type: none"> Remote Remote force First request
LP(F)	Change the printer to local mode immediately.	<ul style="list-style-type: none"> The printer is set to local mode. If the communication status is In session processing and the mode is not SUSPEND, the current listing is finished. This command allows for local printing until the next HP or HP(F) command takes effect. This command can be used if the printer status is: <ul style="list-style-type: none"> Remote Remote force First request <p>That is, the same action sequences and conditions as for the LP command. Besides, note the following:</p> <ul style="list-style-type: none"> Any host computer-controlled listing is interrupted The command is also available when LP is pending.
SA	Set alternate mode.	<ul style="list-style-type: none"> Printer manager server operation mode is set to alternate.
SE n	Set exclusive mode. The value of n can be 1, 2, 3, 4, 5, or 6.	<ul style="list-style-type: none"> Printer manager server operation mode is set to exclusive.
SS	Suspend service.	<ul style="list-style-type: none"> LU operation stops after printing the current listing. If the LU is not In session processing status, it remains inactive, even though a host computer request for printing is received. The activity can be resumed using the AS command. This command may be used in any communication status.

Table 30 (Page 5 of 5). Operator interface commands

Command	Meaning	Action sequence
SS(F)	Suspend service immediately.	<ul style="list-style-type: none"> • LU operation stops immediately. • The activity can be resumed using the AS command. • This command may be used in any communication status and when an SS command is pending.

Function keys

The bottom part of the status and command panels contains the available function keys. For the Status and Command panels, these are:

Enter	Refreshes status, provided no command is entered. If a command is entered, it performs the command and refreshes status.
F1	Provides help.
F3	Leaves the operator interface.
F7	Scrolls up, showing previous emulator status.
F8	Scrolls down, showing next emulator status.
F9	Switches from emulator panels to printer panels, and vice versa.
Tab forward	Goes to the next command field.
Tab backward	Goes to the previous command field.
→	Jumps a character forwards in the command field.
←	Jumps a character backwards in the command field.

On the help panels, two function keys are defined:

Enter	Goes to the next help panel. If you are in the last help panel, it returns you to the Status and Command panel from which help was requested.
F3	Goes to the previous help panel. If you are in the first help panel, it returns you to the Status and Command panel from which help was requested.

Sample panels

The following figure shows a sample Status and Command panel for a 3287 printer emulator:

```
LAN DISTRIBUTED PLATFORM/DOS
Operation: aaaa aaaaa aaaaaa
-----
Host application ID ...: bbbbbbbb
Printer ID .....: c   Mode ...: dddddddd   Priority ....: e
-----

Communication status .....:
Host connected

Printer status .....:
Attending this service

Current listing options .....:
Operation intervention active
End of listing is under application control

Command ...: xx           Parameter ...:           Subcode ...:

-----
Enter=Enter command      Enter=Refresh status      F1=Help
Esc/F3=Return            F7/F8=Scrolling Emulators  F9=Printers
```

Figure 45. Operator Interface Status and Command panel: 3287 Printer Emulator

The panel is divided into three areas:

- The upper part of the panel contains the LANDP identification, the emulator type, the host computer application identification, the assigned-printer identification, and the printer priority.
- The middle part of the panel contains the communication status, the printer status, and the current listing options. Three input fields are provided to enter a command, its parameters, and subcodes.
- The bottom part of the panel contains the available function keys.

When the F1 key is pressed, the help panels are displayed to show the available commands and their parameters.

The following figure shows a sample Status and Command panel for the printer manager server:

```

                                LAN DISTRIBUTED PLATFORM/DOS
                                Operation : Parallel Printer Management
-----
Printer ID .....: c
-----

Printer status .....:
    Printer not online
    Alternate use

Services .....:
    1 3287 emulator
    2 47X2 printer
    3 User SERVER1
    4 User SERVER2
    5
    6 Others

Command ....: xx           Parameter ....:

-----
Enter=Enter command      Enter=Refresh status      F1=Help
Esc/F3=Return           F7/F8=Scrolling Ports    F9=Emulators

```

Figure 46. Operator Interface Status and Command panel: Printer Manager

The panel is divided into three areas:

- The upper part of the panel contains the LANDP identification and the printer identification.
- The middle part of the panel contains the printer status, and a list of the possible clients. Two input fields are provided to enter a command and its parameters.
- The bottom part of the panel contains the available function keys.

When the F1 key is pressed, the help panels are displayed to show the available commands and their parameters.

Local resource manager server



The local resource manager server enables applications to use LANDP common API functions to interact with specific LANDP for DOS components, if the applications and components are installed in the same workstation as the local resource manager server.

The local resource manager server provides services to:

- Get communication and printer status from the emulators
- Control emulator communication with the host computer
- Control printing of data
 - On parallel-attached printers
 - On serial-attached printers

The local resource manager server provides the same services as the operator interface. Only one of these two LANDP for DOS components can be loaded in a workstation. The decision of which to use is made during customization. The host computer or workstation application IDs can also be configured customization.

If you want your application to use this identifier to refer to an LU session, you must provide this information at customization. Since each identifier stands for an emulator session, you must use a different identifier for each emulator session. If you do not use this identifier, you may refer to the emulators by their LU session number identifier.

The local resource manager server provides functions to control the operation of the following LANDP for DOS local resources:

- Printer manager server
- 3270 emulator
- 3287 printer emulator

If IBM Financial Branch System Integrator (FBSI) is installed in the same workstation and it is used with IBM 4700 Finance Communication System, the local resource manager server controls the operation of the banking printer program (BPP).

Maximum number of LU sessions and printers

Depending on the LANDP for DOS components selected during customization, the local resource manager server can interact with a maximum of:

- Five 3270 emulators. Each has an LU_2 session assigned.
- Five LU_1 sessions through a 3287 printer emulator. Each session has a priority number assigned. The 3287 printer emulator can provide service to a maximum of three supported printers attached to parallel ports.
- One printer manager server that manages up to three parallel-attached printers.
- Four banking printer programs (BPPs). Each BPP has an LU_0 session assigned (only if FBSI is installed).

Service availability manager (SAM)



The System Availability Manager (EHCSAM) monitors all active servers and workstations in a LANDP workgroup. Used by Shared-File External Logging Replicator (XLR), EHCSAM supports fault-tolerant data access with minimal user intervention.

The primary functions of EHCSAM are:

- Maintaining state information for XLR servers.
- Dynamic reconfiguration of XLR servers.
- Message broadcast facility.

Maintaining state information for XLR servers

State information is data that, for example, identifies which workstations are running the active and backup XLR servers. To ensure that this information is up-to-date, all SAM servers in a workgroup communicate with each other and use a voting system to decide which information is current.

When a SAM server starts up all active SAM servers exchange state information for all configured active and backup XLR servers. This is effectively voting for the active server. For example:

One SAM server No vote possible. See note.

Another SAM server comes up Take vote. If two votes agree, the chosen XLR Server is made active.

If votes differ, wait for another SAM server to start up. Take another vote, which should yield a majority.

Note: When only one SAM server is available or there is no majority vote, messages are issued and user intervention is required. *LANDP Installation and Customization* describes the messages and the required actions in the description of the /O parameter of the loading statement for the shared-file server. This description is in the chapters on preparing OS/2 and Windows NT workstations.

Because of the above, run EHCSAM on **three** or more workstations if possible.

Dynamic reconfiguration of XLR servers.

When an XLR server starts or changes role from backup to active, it uses SAM to update the LANDP routing tables to indicate which workstation is running the service. SAM notifies all current workstations (and any workstations that subsequently join the workgroup) of the new location of the server. EHCSAM runs only on OS/2 and Windows NT, but it supports DOS and AIX workstations for these routing changes.

Message broadcast facility.

service availability manager, OS/2 Windows NT

XLR servers and EHCSAM broadcast messages to all workstations running EHCSAM. XLR servers use this broadcast facility to notify the operator of service startup, role change, termination and error situations.

OS/2 uses EHCPMPRT (a simple GUI application) to display the messages.

Windows NT uses the Application section of the Event Viewer to display the messages. Authorized users can view the messages from any Windows NT workstation on the LAN.

Chapter 8. Application integration servers

This chapter describes the servers provided by the LANDP family to enable access to non-LANDP environments and facilitate resource and data sharing with non-LANDP programs. Information about the following servers and services is included:

CICS interface server

Enables LANDP applications to access CICS facilities and data located in the same or another workstation in the same LANDP workgroup.

DDE access server

Enables LANDP applications to exchange data with applications that can use the clipboard or dynamic data exchange (DDE) conventions.

Batch machine facility

Enables jobs to be processed in batch mode.

MAIL program

Used to send messages and files to other members in the workgroup or to a batch machine, or to receive messages and files.

Object Post Box Server

Provides the means for a client application to store messages in queues that can be accessed by the owner of the message queue.

Batch machine operator

Provides a graphical interface to the batch machine for LANDP for OS/2 workstations.

LANDP-DCE interoperability

Enables DCE clients to access LANDP servers and LANDP clients to access DCE servers.

CICS interface server



The CICS interface server enables LANDP applications to access CICS facilities and data located in the same or another workstation in the same LANDP workgroup or on a remote host.

The application that requests services from the CICS interface server can run in any workstation within the LANDP workgroup. However, the CICS interface server must be installed in the same LANDP for OS/2 workstation as the CICS system that provides the CICS external call interface (ECI) it uses.

More than one CICS interface server can be installed in a LANDP workgroup. This enables an application to request services from more than one CICS interface server concurrently.

application integration servers

Under LANDP for DOS, the client can be a Windows 3.1 or non-Windows application. Under LANDP for OS/2 and Windows NT the client can be a GUI (graphical user interface) or non-GUI application.

The CICS interface server provides:

- An interface and protocol to send and receive data by requesting multiple transactions belonging to a single “logical unit of work,” using a reduced set of functions.
- Full transaction and data integrity.
- Support for synchronous and asynchronous parallel calls.

See *LANDP Programming Reference* for more information about the LANDP CICS interface server functions.

DDE access server



The DDE access server provides an interface between LANDP and dynamic data exchange (DDE) compliant applications. The interface uses specific Presentation Manager® methods for exchanging data using the clipboard or DDE conventions. This server extends these methods for remote access and makes the location of these services transparent to the user application.

The DDE protocol defines a naming convention used to specify data that can be exchanged between DDE-compliant applications. A DDE client application requests data from a DDE server application using the DDE protocol to identify the specific data being requested.

The DDE access server acts as a DDE client for LANDP clients to get data from DDE Server Applications (DSA).

A DSA can be a Presentation Manager program that is OS/2 DDE-compliant or a Windows 3.1 DDE-compliant program (running in a WIN-OS/2® session).

The DDE access server:

- Simplifies the usage of DDE functions through a set of high-level functions that enable data exchange with a single function call.
- Provides access to the DDE environment for non-Presentation Manager applications.
- Provides remote access to the DDE environment. LANDP application can run in a different workstation from the one where the DDE access server and DDE application are running.
- Can access OS/2 and Windows 3.1 DDE applications.
- Provides a function for starting the DDE server application, if it is not active when your LANDP application attempts to establish a conversation.

- Provides functions for locking a DDE server application topic. If you lock a topic, no other LANDP application can access this topic until it is released.

Using the clipboard services

The DDE access server also provides clipboard services to:

- Copy data in text format to the clipboard
- Get data in text format from the clipboard
- Get data in link format from the clipboard

If data from a DDE application is copied to the clipboard using a **copy** or **cut** command, the DDE server application inserts the application, topic, and item names of the selected item into the clipboard in link format. A client application can get these application, topic, and item names and use them to establish a DDE conversation with the DDE server application.

By using the DDE access server clipboard services, a remote application can access the clipboard in the workstation where the DDE access server is installed.

See *LANDP Programming Reference* for more information about the functions associated with the DDE access server.

Batch machine facility



The batch machine facility is used to process jobs in batch mode. This facility is intended for environments with the following requirements:

- Integrating applications that deal with input and output files.
- Reducing the workload in the client workstations by moving work into more powerful server workstations.
- Processing jobs asynchronously in unattended mode at a later time.
- Concentrating jobs in a single workstation, thus reducing the need for multiple copies of the same program.
- Sending jobs for later processing to workstations that are not online when you send these jobs.

The batch machine facility consists of the object post box server (OPBS) and the batch machine loader server.

The OPBS program manages the message traffic among the users defined for the system manager server and the programs written for batch processing.

The OPBS program requires that the shared-file server be installed in the same workstation, and that a system manager server be installed in the same workgroup.

MAIL program, OS/2 DOS

The batch machine loads the batch programs into memory, where they stay resident, and informs OPBS when the batch programs are ready to receive messages. Only one BMLS program can be loaded in a workstation; however, any number of LANDP for OS/2 workstations in the workgroup can load this server.

Accessing the batch machine

The batch machine is accessed using the LANDP MAIL program. For more information about the MAIL program, refer to "MAIL program." It can also be accessed using the batch machine operator installed in LANDP for OS/2 workstations. For more information about the batch machine operator, refer to "Batch machine operator" on page 204.

Processing jobs in batch mode

To process jobs in batch mode, write a batch or executable program that performs the unattended processing. This program must be in an accessible path as specified when loading the BMLS program. For information about the server loading statement, refer to *LANDP Installation and Customization*. See the *LANDP Programming Reference* for more information about writing a batch or executable program that runs unattended.

Batch machine log file management

The workstation that provides the shared-file services, as defined at customization, contains the log files or inactive queues for messages processed by the batch machine users. From time to time, you should erase the contents of these log files in order to free some disk space. The log files name is EHCMLOG. They are placed in the directory selected at customization for the BMLS files.

MAIL program



The LANDP MAIL program is used to send messages and files to another user in the same workgroup or to a batch machine, or to receive these messages and files. The MAIL program offers the same features as the batch machine operator.

The MAIL program can be invoked without any parameters. It will enter a *prompt mode*, from which MAIL commands can be entered. The MAIL program leaves the prompt mode when the Enter key is pressed without any command (or when an error occurs).

The MAIL program can also be invoked using the commands it provides as parameters.

Mail program commands

The MAIL program provides the following commands, which you invoke from the command line:

MAIL HELP

Used to display information about the other MAIL commands.

MAIL LOGON (*userid(password)*)

where:

userid Is the user ID of the system manager server.

password Is the password, if required.

Used to sign on to the system manager server. This command is required before sending any message to the batch machine. The unattended batch processing programs do not need to issue this command.

MAIL LOGOFF

Used to sign off the previously signed-on user ID from the system manager server. The unattended batch processing programs do not need to issue this command.

MAIL SEND *pcfilespec userlist (/ACK) (/comment)*

where:

pcfilespec Is a disk file, specified as follows:

(*drive:*) (*path\filename.ext*)

userlist Is a list consisting of one or more system manager server user IDs or *bm1names* specified at customization, where *bm1names* are the public user IDs used by the object post box server.

/ACK Is a request for a low level acknowledgement, so that a data message will be generated and placed in the issuer's input queue when the original message is processed by the recipient.

/comment Is any additional data included in the data message header.

Places the specified file in the issuer's output queue and in the recipient's input queue.

MAIL MESSAGE *userlist (/ACK) (/comment)*

where:

userlist Is a list consisting of one or more system manager server user IDs or *bm1names* specified at customization, where *bm1names* are the public user IDs used by the object post box server.

/comment Is any additional data included in the data message header.

Used to send a message header and optional comments to the list of system manager server user IDs or *bm1names* specified through the *userlist* parameter.

MAIL program, OS/2 DOS

MAIL DISCARD *mailfilespec* | *spoolid*

where:

mailfilespec Is a queue file, specified as follows:

(*userid*\\)*filename*(.ext)

spoolid Is the number assigned to a queue message.

Used to erase the specified data message from the issuer output queue and from the recipient input queue.

MAIL RECEIVE *mailfilespec* | *spoolid* (*pcfilespec*) (/O)

where:

mailfilespec Is a queue file, specified as follows:

(*userid*\\)*filename*(.ext)

spoolid Is the number assigned to a queue message.

pcfilespec Is a disk file, specified as follows:

(*drive*:) (*path*\\)*filename*(.ext)

Used to transfer data to the specified recipient's local file and erase the data transferred from the queues in the same way as the MAIL DISCARD command does. If you specify the /O parameter, any existing copies of the same file are overwritten.

MAIL PEEK *mailfilespec* | *spoolid* (*pcfilespec*) (/O)

where:

mailfilespec Is a queue file, specified as follows:

(*userid*\\)*filename*(.ext)

spoolid Is the number assigned to a queue message.

pcfilespec Is a disk file, specified as follows:

(*drive*:) (*path*\\)*filename*(.ext)

Used to transfer data to the specified recipient's local file in the same way as the MAIL RECEIVE command does without erasing the data from the queues.

MAIL LIST *mailfilespec* (OUTPUT) (LOG)

where:

mailfilespec Is a queue file, specified as follows:

(*userid*\\)*filename*(.ext)

Used to show the contents of the input or output message queues for the issuer of this command. By default, it shows the active input queue. If you specify the OUTPUT parameter, this command shows the active output queue. If you specify the LOG parameter, the command shows the inactive queues, that is, the historical list of messages processed by the destination users. This command displays the following fields:

SpoolID	Identification number for the message. This number is dynamically assigned by the OPBS program. You can use this number in the RECEIVE, DISCARD, and PEEK commands.
Name	Name associated to this message (file name). If the message is an acknowledgement, it will read: [Acknowledge].
UserID	User identification defined for the system manager server or batch machine name.
NodeID	Workgroup name, as defined during the system manager server profile customization. Usually, this will be a fixed value after you have finished customizing. The default is LanID.
Size	Size (in bytes) of the message. It does not include the message header data or the <i>/comment</i> parameters.
Date	Date when the message was sent.
Time	Time when the message was sent.
Ref	Additional data found in the data message header (<i>/comment</i>).

You can also specify commands by typing the significant letters of the command name. The significant letters for the MAIL program commands are the following:

H	Help
LO	Logon
LOGOFF	Logoff
S	Send
M	Message
D	Discard
R	Receive
P	Peek
L	List

You can redirect the input and output of this program to or from a file or device. The ERRORLEVEL set by the MAIL program at exit can be:

- 0 Command completed successfully.
- 1 Syntax errors when invoking the program. Issue the command again.
- 2 The command has a correct format. It failed due to temporary causes (for instance: disk full or spool file not found). Retry the operation later on.
- 3 Unexpected error or error due to system initialization causes.

Examples:

1. Logging on to the system manager server:

```
MAIL LOGON
```

2. Receiving all files sent by user ID JOHN:

```
MAIL RECEIVE JOHN\*.*
```

batch machine operator, OS/2

3. Sending all .EXE files to user IDs JOHN and PETER with acknowledgement:

```
MAIL SEND *.EXE JOHN PETER /ACK
```

Object post box server



The object post box server provides the means for a client application to store messages in queues that can be accessed by the owner of the message queue.

The messages are compressed and stored in a shared-file server. This provides inherent data integrity and recoverability.

The messages are organized as information queues that are owned by a single user. The user could be a workstation, object command server, or a user defined by the system manager server. Only the owner of a queue can read the stored data. However, any client can send a message to any known queue.

The messages are composed of two parts:

- Message header:
 - Origin and destination user identifiers
 - Origin and destination LAN identifiers
 - Timestamp
 - Data size
 - Hold status
 - Type
 - Additional data

- Message data:

As much user provided data as desired.

The object post box server must be accessible to all workstations that are going to use it. Because the object post box server uses the shared-file server and the system manager server, it has to be defined as a user of these servers.

Batch machine operator



The batch machine operator provides a graphical interface for accessing the batch machine. It offers the same features as the MAIL program. The batch machine operator requires that a system manager server user be logged on at the workstation where you intend to use the batch machine operator. If no user is logged on, the batch machine operator displays a signon panel. Leaving the batch machine operator will then cause a signoff to occur. If a user was already signed on, no signon or signoff is performed.

If you want to send jobs to the batch machine, you have to define these jobs by supplying the following data:

- The *destination batch machine* to which the job will be sent. This is required information.
- The *command and parameters* that will be processed by your batch program.
- A *list of files* that will be sent to the destination batch machine before processing the command.

Note: At least one command or one file in the list must be present in each job definition.

You can store job definitions on your local disk and retrieve them later on for processing.

When you have defined a new job or retrieved a previously saved job definition, you can send the job to its destination batch machine. The batch machine operator will notify you when the job has finished processing. You can stop the sending process while it is in progress. The batch machine operator performs the required backout operations if a file was partly sent. All files already sent remain at their destination queue.

You can manage your *input queue*, which contains the files sent back by the batch machines as entries in a list with the following information about each file: spool ID, origin, timestamp, file size, file name, and command. You can select multiple files and perform the following actions on them:

- Store the files on the local disk
- View the files without storing them
- Delete the files from the queue.

You can stop any of these operations while they are in progress. The batch machine performs the required backout operations for the file with which you are working.

You can also perform the following tasks:

- View your *output queue*. It is the list of files that were successfully sent and not yet processed.
- View your *output action log*. It is a historical list of files that were successfully sent and processed.
- View your *input action log*. It is a historical list of files received or deleted from the input queue.

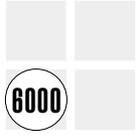
The batch machine operator provides online help to assist you in performing the described tasks.

DBCS mode restrictions

The batch machine operator requires that the following information contains SBCS characters only:

- Commands
- Parameters

LANDP-DCE interoperability



LANDP and DCE are development platforms used to create distributed applications. These platforms follow the client/server model in heterogeneous hardware and software environments.

The Open Systems Foundation (OSF) adopted the Distributed Computing Environment (DCE) technology to develop and use distributed applications on heterogeneous hardware platforms and operating systems.

LANDP for AIX provides interoperability with IBM AIX DCE running under IBM AIX 3.2 or later, for RISC System/6000.

The interoperability provided by LANDP for AIX enables DCE clients to access LANDP servers and LANDP clients to access DCE servers. These LANDP servers and clients can be installed in any workstation in the workgroup, regardless of the underlying LANDP program installed. A DCE client can access LANDP servers residing in different LANDP workgroups. LANDP clients from several LANDP workgroups can access the same DCE server.

DCE clients that conform to the LANDP-DCE interface can access LANDP servers. LANDP clients that conform to the LANDP common API can access DCE servers.

DCE servers must conform to the LANDP-DCE interface to get requests from LANDP clients. However, DCE servers without a LANDP-DCE interface may also be accessible to LANDP clients through a DCE server conforming to the LANDP-DCE interface.

A single DCE server conforming to the LANDP-DCE interface can enable access to several DCE servers that do not conform to this interface.

Figure 47 on page 207 shows the interoperability between the LANDP and the DCE environments.

LANDP for AIX and, at least, IBM AIX DCE Base Services/6000, must be present in the same IBM RISC System/6000 to attain LANDP-DCE interoperability.

Some sample DCE client and server programs are provided with LANDP for AIX. See the *LANDP Programming Reference* for more information.

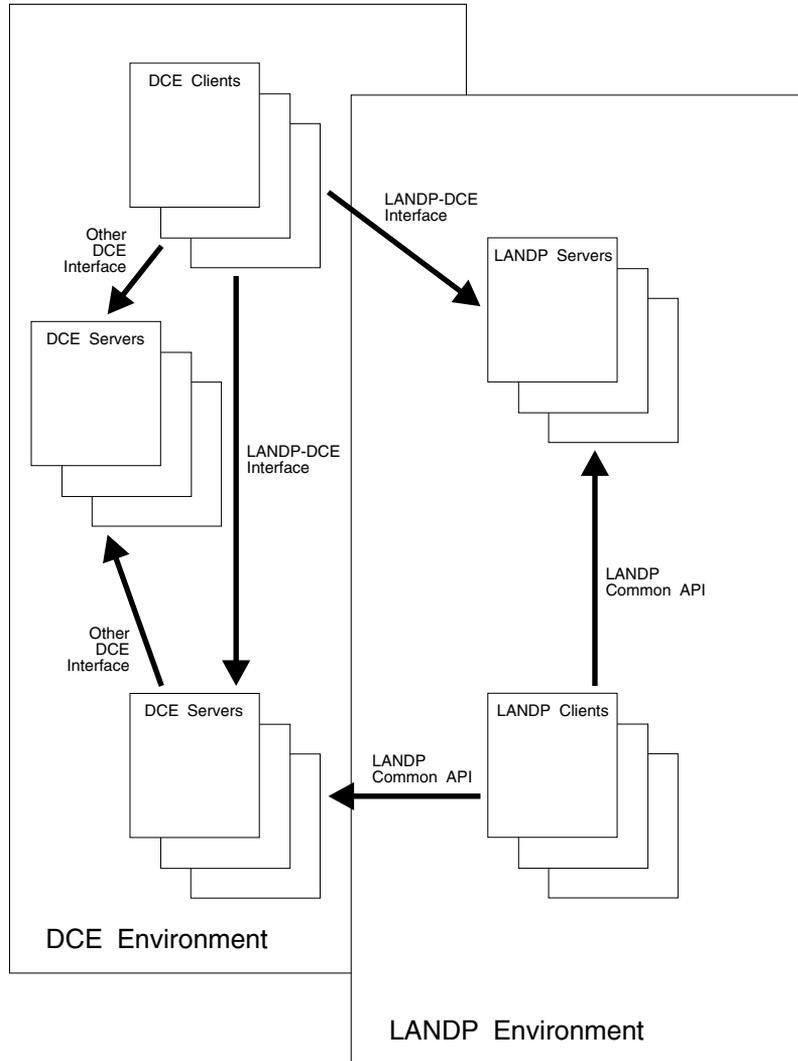


Figure 47. LANDP-DCE Interoperability

LANDP-DCE interoperability characteristics

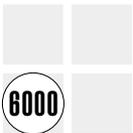
The LANDP-DCE interoperability provides the following capabilities:

- Support in both environments for using the remote procedure call (RPC). It enables LANDP and DCE clients to issue remote requests to DCE and LANDP servers.
- Support for DCE clients and servers to inter-operate with IBM LANDP, using Naming Services, the same way as with other DCE programs in a pure DCE environment.
- DCE login. DCE clients must be logged in to access LANDP services before starting LANDP-DCE interoperability. See “Performing a `dce_login`” for more information.

The LANDP-DCE interoperability does not support the following DCE features:

- The DCE Authenticated Remote Procedure Call (RPC). DCE clients cannot issue requests to LANDP servers using Authenticated RPC nor can LANDP clients issue requests to DCE servers using the Authenticated RPC.
- Access Control List (ACL) Managers that control the access of DCE clients to LANDP servers and vice versa.
- Distributed Time Services (DTS).

Performing a `dce_login`



If you plan to use LANDP-DCE interoperability, you must perform a `dce_login` for the DCE clients that will access LANDP services. You have to do this before starting LANDP-DCE interoperability. Enter:

```
dce_login account_name account_password
```

where:

`account_name` Is the name of the account created for the LANDPDCE principal.

`account_password` Is a password assigned to this account by the DCE system administrator.

LANDP clients can access DCE servers that have performed `dce_login` to an account that provides privileges different from those provided by the account created for the LANDPDCE principal. To enable access, you must also perform `dce_login` to an account that provides those other privileges before loading LANDP for AIX.

When you start LANDP for AIX, although you have enabled LANDP-DCE interoperability, the procedure does not start the DCE servers to be accessed by LANDP clients. Ensure that you have started those DCE servers before using them.

Chapter 9. LANDP facilities and utilities

This chapter describes the facilities and utilities provided by the LANDP family. Included is information for:

File transfer facility

Used by a workstation to send files to and receive files from a host computer. It does not support DBCS data.

Transmission is only supported from screens that are 24 lines deep by 80 columns wide.

3270 send/receive facility

An improved facility for transferring files between a workstation and a host computer. In addition to the function offered by the file transfer facility, it supports DBCS data under DOS, and can be used with any supported screen size.

4707 monochrome display support

Information about how to use the various modes of the IBM 4707 monochrome display.

Multiple virtual DOS machine relay

Used in LANDP for OS/2 and Windows NT workstations to support LANDP for DOS emulators, and multiple WIN-OS2 (on OS/2 only) and DOS applications.

Microsoft Windows 3.1 support

Information about how to run LANDP for DOS under Microsoft Windows 3.1 or Windows for Workgroups 3.11, and requesting LANDP for DOS services from Windows applications.

Utility programs

Various utility programs provided with the LANDP family.

File transfer facility



The file transfer facility is used to send files to and receive files from a host computer.

The file transfer facility provides a SEND command and a RECEIVE command. The SEND command transfers a copy of a file on the workstation hard disk or diskette to the host computer. The RECEIVE command transfers a copy of a file in the host computer to the workstation hard disk or diskette.

These are disk-resident programs, which the workstation operator can invoke from the DOS command line or a batch (.BAT) file. They can also be invoked from a program using the LANDP high-level language application programming interface (HLLAPI).

file transfer facility, OS/2 DOS Windows NT

The IBM Financial Branch System Integrator (FBSI) also provides a file transfer facility to send files from the workstation to the 4700 processor and to receive files from the 4700 processor at the workstation. For information on the FBSI, refer to *Financial Branch System Integrator Programmer's Reference*, GA19-5452.

Workstation requirements

The file transfer facility has the following requirements:

- The file transfer message file must be available in the current directory on the default disk or diskette drive.
- The LANDP 3270 emulator must be installed.
- The SEND and RECEIVE programs must reside in the current directory.

The LANDP for OS/2 and Windows NT file transfer facility can be used only in a virtual DOS machine (VDM).

Host computer requirements

To use the file transfer facility, the host computer must have received a logon from the LANDP workgroup. One of the following 3270 file transfer programs must be installed in the host computer:

System	IBM Program Required	Program Number
VM/CMS	IBM 3270-PC FTP	5664-281
MVS with TSO	IBM 3270-PC FTP	5665-311
MVS with CICS	CICS/VS 3270-PC FTP	5798-DQH

VSE/SP™ and VSE/ESA host systems do not need any host computer file transfer programs. The host computer file transfer is supported by the operating systems.

For information on how to use the file transfer facility, see “Using the file transfer facility commands.”

Compatibility with other emulators

Because of syntax differences between LANDP emulators and other emulators, problems can arise when invoking the file transfer programs through another program.

To solve these problems, LANDP provides the capability of passing all the command options, without checking, to the host computer session command line. Then the host computer file transfer program is invoked with those options unchanged.

Using the file transfer facility commands

The file transfer facility provides two commands, SEND and RECEIVE. This section explains both commands, their parameters and options by taking into account the different host computer operating systems supported.

Notational conventions

The following notational conventions are used in this section:

- [] indicates optional parameters
- Relevant DOS parameters are only briefly explained.

Refer to the appropriate PC DOS manuals for detailed information.

MVS with CICS SEND command

The format of the command is:

```
[accpath]SEND [d:][path]filename[.ext] cicsfile /H:C[n][options]
```

where:

- accpath* Is the workstation drive and path where the disk or diskette with the file transfer programs reside. Entering *accpath* is required only if it is not the default drive or directory, or if the path is not defined.
- SEND Is the required command name.
- d:* Is the workstation drive where the disk or diskette with the DOS file to be sent resides. If not specified, the system assumes the default drive.
- path* Is an optional DOS directory specifier for the directory where the DOS file to be sent resides.
- filename* Is the required name of the DOS file to be sent.
- .ext* Is the DOS file extension. This is required if the file to be sent has an extension.
- cicsfile* Is the required name of the CICS file.
- /H:C[*n*] Is a HOST CICS parameter that specifies file transfer from the workstation to the CICS host computer. *n* stands for the 3270 emulator short name and ranges from 1 to 5; the default value is 1.

The following *options* are available:

- /C The CRLF option specifies whether carriage return and line feed characters are recognized as record separators, and deleted before storing to the host computer storage medium.
- /T The TRANSLATE option is used when the DOS file is stored in ASCII format and is to be translated to EBCDIC by the host computer file transfer program. If specified, this option causes the CICS file transfer program to translate records from ASCII to EBCDIC before writing them to the data set.
- /I:*n* The INTERVAL option specifies the number of 10-second intervals to be used as the time-out value during file transfer; *n* stands for the number of 10-second intervals. The minimum value (/I:3, meaning 30 seconds) is the default time-out value. The maximum value is /I:30, meaning 300 seconds (5 minutes). This option is useful when you are transferring files via a heavily loaded or slow communication line.

file transfer facility, OS/2 DOS Windows NT

Example: The example below shows the use of the MVS with CICS SEND command, assuming the following:

- The SEND.COM program resides on drive C of the workstation.
- The source DOS file named FIL.ASC is on drive A of the workstation.
- The destination CICS file name is NEWFILE.
- The DOS file is stored in ASCII format and is to be converted to EBCDIC by the CICS file transfer program.
- Carriage return and line feed characters are to be removed before storing the CICS file.
- The second emulator is used.

At the DOS prompt, the workstation operator should enter the following:

```
C:SEND A:FIL.ASC NEWFILE /H:C2/T/C
```

MVS with TSO SEND command

The format of the command is:

```
[accpath]SEND [d:][path]filename[.ext] dsname[(mem)] [/passw] /H:T[n] [options]
```

where:

accpath Is the workstation drive and path where the disk or diskette with the file transfer programs reside. This is required only if it is not the default drive or directory, or if the path is not defined.

SEND Is the required command name.

d: Is the workstation drive where the disk or diskette with the DOS file to be sent resides. If not specified, the system assumes the default drive.

path Is an optional DOS directory specifier for the directory where the DOS file to be sent resides.

filename Is the required name of the DOS file to be sent.

.ext Is a DOS file extension. This is required if the file to be sent has an extension.

dsname Is the required name of the MVS/TSO data set. If a host computer data set already exists with the data set name specified in the command, the new data set replaces the old data set unless the /A option (see below) is specified.

(*mem*) Is an optional name of a member in a partitioned data set.

/passw Is the password if the data set is password-protected.

/H:T[*n*] Is a HOST MVS/TSO parameter that specifies file transfer from the workstation to the MVS/TSO host computer. *n* stands for the 3270 emulator short name and ranges from 1 to 5; the default value is 1.

There should be no space between data set name, member name, and password. Single quotes can be typed around the combined data set name, member name, and password to indicate that the user ID is not prefixed.

The following *options* are available:

- /A The APPEND option specifies that a workstation file is to be attached to the end of the host computer data set. The default is REPLACE. This option cannot be specified for members of a partitioned data set.
- /B:*n* The BLKSIZE option specifies the block size of the host computer data set. *n* stands for the length of a data block, in bytes. The default value is LRECL for new files. If appending or replacing, /B is ignored.
- /C The CRLF option specifies whether carriage return and line feed characters are recognized as record separators, and deleted before storing to the host computer storage medium.
- /L:*n* The LRECL option specifies the logical record length of the host computer data set records. *n* stands for the number of characters in each record. In RECFM V, this is the maximum record size expected. The default value is 80 for new data sets.

/R:*format*

The RECFM option specifies the record format and characteristics of the host computer data set.

format is one of the following values:

- F Fixed length
- V Variable length
- U Undefined

If CRLF is specified, for new data sets, the default is V. If CRLF is not specified, for new data sets, the default is F.

- /S:*n*[,*m*] The SPACE option specifies the amount of space to be allocated for a new data set. The parameters *n* and *m* are measured in the same units as the SPACE UNITS mentioned below, with *n* the amount of space, and *m* an increment. If the SPACE option is specified, the /U option (see below) can also be specified.
- /T The TRANSLATE option is used when the DOS file is stored in ASCII format and is to be translated to EBCDIC by the host computer file transfer program. This option causes the MVS file transfer program to translate records from ASCII to EBCDIC before writing them to the data set.
- /U:*units* The SPACE UNITS option specifies the units used for SPACE quantity and increment. This option can be specified only when the SPACE option is specified.

units is one of the following values:

- A(*n*) Avblock (value)
- C Cylinders
- T Tracks

file transfer facility, OS/2 DOS Windows NT

/I:n The INTERVAL option specifies the number of 10-second intervals to be used as the time-out value during file transfer; *n* stands for the number of 10-second intervals. The minimum value (*/I:3*, meaning 30 seconds) is the default time-out value. The maximum value is */I:30*, meaning 300 seconds (5 minutes). This option is useful when you are transferring files via a heavily loaded or slow communication line.

Example: The example below shows the use of the MVS/TSO SEND command, assuming the following:

- The SEND.COM program resides on drive C of the workstation.
- The source DOS file named FIL.ASC is on drive A of the workstation.
- The destination TSO data set name is NEWFILE.
- NEWFILE is a partitioned data set, and NEWMBR is a member.
- NEWFILE is password-protected, and PW123456 is the password.
- The DOS file is stored in ASCII format and is to be converted to EBCDIC by the TSO file transfer program.
- Carriage return and line feed characters are to be removed before storing the TSO data set.
- The second emulator is used.
- The logical record length is to be set to 132.
- The record format is fixed.
- An initial space of 20 tracks is to be allocated in increments of 10.
- The time-out value is set at 50 seconds.

At the DOS prompt, the workstation operator should enter the following:

```
C:SEND A:FIL.ASC 'NEWFILE(NEWMBR)/PW123456' /H:T2/T/C/L:132/R:F/S:20,10/U:
```

VM/CMS SEND command

The format of the command is:

```
[accpath]SEND [d:][path]filename[.ext] fn ft [fm] /H:V[n][options]
```

where:

accpath Is the workstation drive and path where the disk or diskette with the file transfer programs reside. This is required only if it is not the default drive or the default directory, or if the path is not defined.

SEND Is the required command name.

d: Is the workstation drive where the disk or diskette with the DOS file to be sent resides. If it is not specified, the system assumes the default drive.

path Is an optional DOS directory specifier for the directory where the DOS file to be sent resides.

- filename* Is the required name of the DOS file to be sent.
- .ext* Is a DOS file extension. This is required if the file to be sent has an extension.
- fn* Is the required name of the VM/CMS file. If a host computer file already exists with the file name, file type, and file mode specified in the command, the new file replaces the old file, unless the /A option (see below) is specified.
- ft* Is the required file type of the VM/CMS file.
- fm* Is the VM/CMS file mode. This is optional; if omitted, the default value is A1.
- /H:V[n]* Is a HOST VM/CMS parameter that specifies file transfer from the workstation to the VM/CMS host computer. *n* stands for the 3270 emulator short name and ranges from 1 to 5; the default value is 1.

The following *options* are available:

- /A* The APPEND option specifies that the workstation file is to be attached to the end of the host computer file. The default is REPLACE.
- /C* The CRLF option specifies whether carriage return and line feed characters are recognized as record separators, and deleted before storing to the host computer storage medium.
- /L:n* The LRECL option specifies the logical record length of the host computer file records. *n* stands for the number of characters in each record. In RECFM V, this is the maximum record size expected. The default value is 80 for new files.
- /R:format*
 The RECFM option specifies the record format and characteristics of the host computer file.
format is one of the following values:
 F Fixed length
 V Variable length
 If CRLF is specified for new files, the default is V. If CRLF is not specified for new files, the default is F.
- /T* The TRANSLATE option is used when the DOS file is stored in ASCII format and is to be translated to EBCDIC by the host computer file transfer program. If specified, this option causes the CMS file transfer program to translate records from ASCII to EBCDIC before writing them to the file.
- /I:n* The INTERVAL option specifies the number of 10-second intervals to be used as the time-out value during file transfer; *n* stands for the number of 10-second intervals. The minimum value (/I:3, meaning 30 seconds) is the default time-out value. The maximum value is /I:30, meaning 300 seconds (5 minutes). This option is useful when you are transferring files via a heavily loaded or slow communication line.

file transfer facility, OS/2 DOS Windows NT

Example: The example below shows the use of the VM/CMS SEND command, assuming the following:

- The SEND.COM program resides on drive C of the workstation.
- The source DOS file named FIL.ASC is on drive A of the workstation.
- The destination CMS file name is NEWFILE.
- The destination CMS file type is DATA.
- The destination CMS file mode is A1.
- The DOS file is stored in ASCII format and is to be converted to EBCDIC by the CMS file transfer program.
- Carriage return and line feed characters are to be removed before storing the CMS file.
- The second emulator is used.
- The record format is variable.

At the DOS prompt, the workstation operator should enter the following:

```
C:SEND A:FIL.ASC NEWFILE DATA A1 /H:V2/T/C/R:V
```

VSE/ESA and VSE/SP SEND command

For details about host computer setup, refer to:

- *VSE/ESA Administration*, SC33-6505
- *VSE/SP Administration*, SC33-6405

For information about SEND command formats, refer to:

- *VSE/ESA Using IBM Workstations*, SC33-6509
- *VSE/SP Using IBM 3270 Display Stations and PCs*, SC33-6409

MVS with CICS RECEIVE command

The format of the command is:

```
[accpath]RECEIVE [d:][path]filename[.ext] cicsfile /H:C[n][options]
```

where:

accpath Is the workstation drive and path where the disk or diskette with the file transfer programs reside. Entering *accpath* is required only if it is not the default drive or directory, or if the path is not defined.

RECEIVE Is the required command name.

d: Is the workstation drive where the disk or diskette with the DOS file to be received resides. If not specified, the system assumes the default drive.

path Is an optional DOS directory specifier for the directory where the DOS file to be received resides.

- filename* Is the required name of the DOS file to be received.
- .ext* Is the DOS file extension. This is required if the file to be received has an extension.
- cicsfile* Is the required name of the CICS file.
- /H:C[n]* Is a HOST CICS parameter that specifies file transfer from the CICS host computer to the workstation. *n* stands for the 3270 emulator short name and ranges from 1 to 5; the default value is 1.

The following options are available:

- /C* The CRLF option specifies whether carriage return and line feed characters are to be stored as the last two characters of each line when a file is stored on the workstation storage medium.
- /T* The TRANSLATE option is used when the CICS file is stored in EBCDIC format and the DOS file is to be stored in ASCII format. If specified, this option causes the CICS file transfer program to translate records from EBCDIC to ASCII before transmitting them to the workstation.
- /I:n* The INTERVAL option specifies the number of 10-second intervals to be used as the time-out value during file transfer; *n* stands for the number of 10-second intervals. The minimum value (*/I:3*, meaning 30 seconds) is the default time-out value. The maximum value is */I:30*, meaning 300 seconds (5 minutes). This option is useful when you are transferring files via a heavily loaded or slow communication line.

Example: The example below shows the use of the MVS with CICS RECEIVE command, assuming the following:

- The RECEIVE.COM program resides on drive C of the workstation.
- The destination DOS file named FIL.ASC is on drive A of the workstation.
- The source CICS file name is OLDFILE.
- The CICS file is stored in EBCDIC format and is to be converted to ASCII by the CICS file transfer program.
- Carriage return and line feed characters are to be added before transferring the file.
- The second emulator is used.

At the DOS prompt, the workstation operator should enter the following:

```
C:RECEIVE A:FIL.ASC OLDFILE /H:C2/T/C
```

MVS with TSO RECEIVE command

The format of the command is:

```
[accpath]RECEIVE [d:][path]filename[.ext] dsname[(member)] [/password] /H:T[n][options]
```

where:

accpath Is the workstation drive and path where the disk or diskette with the file transfer programs reside. This is required only if it is not the default drive or the default directory, or if the path is not defined.

RECEIVE Is the required command name.

d: Is the workstation diskette drive where the diskette with the DOS file to be received is to reside. If not specified, the system assumes the default drive.

path Is an optional DOS directory specifier for the directory where the DOS file to be received is to reside.

filename Is the required name of the DOS file to be received. If a DOS file already exists with the specified file name and extension, and on the specified drive, the new file replaces the old one.

.ext Is the DOS file extension. This is required if the file to be received is to have an extension.

dsname Is the required name of the MVS/TSO data set.

(*member*) Is an optional name of a member in a partitioned data set.

/password This is required if the data set is password-protected.

/H:T[*n*] Is a HOST MVS/TSO parameter that specifies file transfer from the MVS/TSO host computer to the workstation. *n* stands for the 3270 emulator short name and ranges from 1 to 5; the default value is 1.

There should be no spaces between data set name, member, and password. Single quotes can be typed around the combined data set name, member name, and password to indicate that the user ID is not prefixed. The following *options* are available:

/C The CRLF option specifies whether carriage return and line feed characters are to be stored as the last two characters of each line when a file is stored on the workstation storage medium.

/T The TRANSLATE option is used when the TSO data set is stored in EBCDIC format and the DOS file is to be stored in ASCII format. If specified, this option causes the TSO file transfer program to translate records from EBCDIC to ASCII before transmitting them to the workstation.

/I:*n* The INTERVAL option specifies the number of 10-second intervals to be used as the time-out value during file transfer; *n* stands for the number of 10-second intervals. The minimum value (/I:3, meaning 30 seconds) is the default time-out value. The maximum value is /I:30, meaning 300 seconds

(5 minutes). This option is useful when you are transferring files via a heavily loaded or slow communication line.

Example: The example below shows the use of the MVS/TSO RECEIVE command, assuming the following:

- The RECEIVE.COM program resides on drive C of the workstation.
- The destination DOS file named FIL.ASC is on drive A of the workstation.
- The source TSO data set name is OLDFILE.
- OLDFILE is a partitioned data set, and OLDMBR is a member.
- OLDFILE is password-protected, and PW123456 is the password.
- The TSO data set is stored in EBCDIC format and is to be converted to ASCII by the TSO file transfer program.
- Carriage return and line feed characters are to be added before transferring the file.
- The second emulator is used.
- The time-out value is set to 100 seconds.

At the DOS prompt, the workstation operator should enter the following:

```
C:RECEIVE A:FIL.ASC 'OLDFILE(OLDMBR)/PW123456' /H:T2/T/C/I:10
```

VM/CMS RECEIVE command

The format of the command is:

```
[accpath]RECEIVE [d:][path]filename[.ext] fn ft [fm] /H:V[n][options]
```

where:

- accpath* Is the workstation drive and path where the disk or diskette with the file transfer programs reside. This is required only if it is not the default drive or the default directory, or if the path is not defined.
- RECEIVE Is the required command name.
- d:* Is the workstation drive where the disk or diskette with the DOS file to be received is to reside. If not specified, the system assumes the default drive.
- path* Is an optional DOS directory specifier for the directory where the DOS file to be received is to reside.
- filename* Is the required name of the DOS file to be sent. If a DOS file already exists with the specified file name and extension, and on the specified drive, the new file replaces the old one.
- .ext* Is the DOS file extension. This is required if the file being received is to have an extension.
- fn* Is the required name of the VM/CMS file.

file transfer facility, OS/2 DOS Windows NT

- ft* Is the required file type of the VM/CMS file.
- fm* Is the VM/CMS file mode. This is optional; if omitted, the default value is A1.
- /H:V[n]* Is a HOST VM/CMS parameter that specifies file transfer from the VM/CMS host computer to the workstation. *n* stands for the 3270 emulator short name and ranges from 1 to 5; the default value is 1.

The following *options* are available:

- /C* The CRLF option specifies whether carriage return and line feed characters are to be stored as the last two characters of each line when a file is stored on the workstation storage medium.
- /T* The TRANSLATE option is used when the CMS file is stored in EBCDIC format and the DOS file is to be stored in ASCII format. If specified, this option causes the CMS file transfer program to translate records from EBCDIC to ASCII before transmitting them to the workstation.
- /I:n* The INTERVAL option specifies the number of 10-second intervals to be used as the time-out value during file transfer; *n* stands for the number of 10-second intervals. The minimum value (*/I:3*, meaning 30 seconds) is the default time-out value. The maximum value is */I:30*, meaning 300 seconds (5 minutes). This option is useful when you are transferring files via a heavily loaded or slow communication line.

Example: The example below shows the use of the VM/CMS RECEIVE command, assuming the following:

- The RECEIVE.COM program resides on drive C of the workstation.
- The destination DOS file named FIL.ASC is on drive A of the workstation.
- The source CMS file name is OLDFILE.
- The source CMS file type is DATA.
- The source CMS file mode is A1.
- The CMS file is stored in EBCDIC format and is to be converted to ASCII by the CMS file transfer program.
- Carriage return and line feed characters are to be added before transferring the file.

At the DOS prompt, the workstation operator should enter the following:

```
C:RECEIVE A:FIL.ASC OLDFILE DATA A1 /H:V2/T/C
```

VSE/ESA and VSE/SP RECEIVE command

For details about host computer setup, refer to:

- *VSE/ESA Administration*, SC33-6505
- *VSE/SP Administration*, SC33-6405

For information about RECEIVE command formats, refer to:

- *VSE/ESA Using IBM Workstations*, SC33-6509
- *VSE/SP Using IBM 3270 Display Stations and PCs*, SC33-6409

File transfer messages

When either a SEND command or a RECEIVE command is issued, a message is produced if:

- Command syntax errors are detected
- I/O errors occur on the disk or diskette
- Communication errors occur
- The operator is notified of the status of the file transfer process.

All the messages are contained on the IPTXFER.MSG file, which must be available in the current directory on the default drive when either the SEND program or the RECEIVE program runs. All the messages are also written to the standard output device, unless redirected by the workstation operator when issuing the SEND and RECEIVE commands. The standard output device is the display screen.

Each message begins with a 4-digit number and the name of the LANDP program component issuing the message. This information is used to identify the message source, if the command is issued from a program or from a BAT file with ECHO OFF.

If the file transfer process ends successfully, the message number 23 and a zero return code are issued. If the process ends due to an error, a nonzero return code is issued that corresponds to the number of the last message produced. Return codes can be interrogated by the batch subcommands IF and ERRORLEVEL. If the SEND and RECEIVE programs are invoked from a program using the IBM DOS EXEC facility (call 4BH), the return codes can also be interrogated by the IBM DOS EXEC facility (call 4DH). For information on how to use the IBM DOS EXEC facility, refer to "Simplifying SEND and RECEIVE program use" on page 224.

If the workstation operator presses [Ctrl][Break] to end either the SEND program or the RECEIVE program, the return code is X'1A'.

If the IPTXFER.MSG message file is not available when invoking either the SEND program or the RECEIVE program, or if the message to be displayed is not on the file, the following message is displayed:

```
9999 IPTXFER.MSG
```

When this problem occurs, the SEND program and the RECEIVE program continue running; but if a message is to be displayed, only the message number is shown.

file transfer facility, OS/2 DOS Windows NT

For information on the messages and return codes, their causes and the appropriate action to be taken, refer to the *LANDP Problem Determination* manual.

The workstation to host computer connection can be the cause of the error. In this case, you should check the following:

- Availability of the host computer file transfer program
- Existence of error messages on the host computer screen

If you use the TSO file transfer program and an error occurs, you may need to end the file transfer program manually by following these steps:

1. Press the RESET key to clear the operator information area (OIA) on the host computer screen.
2. Press the CLEAR key to clear the TSO broadcast message, if displayed.
3. Press the F2 key to send the cancel file transfer command to the TSO file transfer program.
4. Press the Enter key to set the TSO session to ready condition.

Simplifying SEND and RECEIVE program use

The SEND program and the RECEIVE program can be invoked through the IBM DOS EXEC facility (call 4BH). In this way, the SEND program and the RECEIVE program can be invoked from a program. Thus, you can develop an application that provides a simplified interface to the workstation operator.

You can set the SEND and the RECEIVE commands default options with the DOS SET command. At the DOS prompt, the workstation operator should enter the following:

```
SET IPTOPT= [options]
```

where *options* is one or more SEND or RECEIVE options. They must be written in exactly the same way as they are written when issuing a SEND or RECEIVE command at the DOS prompt. Once the options are set as part of the environment, the SEND and RECEIVE commands read them from the environment.

A parameter value entered in the command line overrides the parameter value set in the environment. For example, entering the option

```
L:132
```

to set logical record length 132, in a SEND or RECEIVE command, overrides current logical record length in the environment.

Some options have neither operands nor inverse options. Thus, if one of those options is set in the environment, it cannot be overridden through a command issued at the DOS prompt. If a command option conflicts with an option in the environment, the SEND program and RECEIVE program end with an error message.

To remove an option from the environment, you must use the DOS SET command. For detailed information about the IBM DOS SET command, refer to the *IBM Personal Computer Disk Operating System* manuals.

Compatibility with other emulators

Due to syntax differences between LANDP emulators and other emulators, problems can arise when invoking the file transfer programs through another program.

To solve those problems, IBM LANDP provides the capability of passing all the command options, without checking, to the host computer session command line. Then, the host computer file transfer program IND\$FILE is invoked, with those options unchanged.

To activate that feature, set the environment variable IPTSYN to EE, before using the file transfer programs. To do that, type the following in the command line:

```
SET IPTSYN=EE
```

Note that this SET statement can be included in the AUTOFBSS.BAT file. The file transfer programs check the variable setting before interpreting the command issued through any of the available ways. Note that the behavior of the file transfer programs depends on that setting. While the IPTSYN variable is set to EE, the information about the parameters and options provided in this chapter does not apply.

To deactivate the feature, type the following in the command line:

```
SET IPTSYN=
```

3270 send/receive facility



The 3270 send/receive facility provides an improved method of sending files from a workstation to a host computer, and of receiving files from the host computer at the workstation. It supports the transmission of DBCS data only under DOS.

You cannot use this facility under a DBCS version of OS/2 or Windows NT because the 3270 send/receive facility uses the LANDP 3270 emulator, which does not support DBCS mode operation under OS/2 or Windows NT.

However, if your workstation is running a single-byte character set (SBCS) version of DOS, OS/2 or Windows NT, you can use either the 3270 send/receive facility or the file transfer facility (see “File transfer facility” on page 211).

3270 send/receive facility, OS/2 DOS Windows NT

The file transfer facility and the 3270 send/receive facility are both installed in the same directory. These programs are:

File transfer facility	SEND.COM and RECEIVE.COM
3270 send/receive facility	SEND.EXE and RECEIVE.EXE

By default, when you issue a SEND or RECEIVE command, the programs for the file transfer facility are used. To use the 3270 send/receive facility instead, delete or rename the SEND.COM and RECEIVE.COM files.

Workstation requirements

The 3270 send/receive facility has the following requirements:

- The LANDP 3270 emulator must be installed.
- Under OS/2 and Windows NT, the LANDP 3270 send/receive facility must be used in the virtual DOS machine that is running the LANDP 3270 emulator.

Host computer requirements

To use the 3270 send/receive facility, the host computer must have received a logon from the LANDP workgroup.

If you are transferring data from a DBCS workstation, one of the following 3270 file transfer programs must be installed in the host computer. You may know these programs by the name APVUFILE.

System	IBM Program Required	Program Number
VM/CMS	PS/55 File Transfer Program VM/CMS Version 2	5799-PGX
MVS with TSO	PS/55 File Transfer Program MVS/TSO Version 2	5799-PGY
MVS with CICS	PS/55 File Transfer Program CICS (MVS, VSE) Version 2	5799-PGZ

If you are transferring data from an SBCS workstation, however, the 3270 send/receive facility will make use of the same host program that the file transfer utility uses, and you will need to ensure that a suitable program is present on the host computer. You can find a list of such programs in "Host computer requirements" on page 212. (You may know these programs by the name IND\$FILE.)

Note that some options for the 3270 send/receive facility are supported under only one type of host program, but not under both.

The 3270 send/receive facility uses a 3270 extended data stream containing Distributed Data Management (DDM) structured fields. To accommodate these, you should ensure that the entry for your workstation in your host computer's VTAM logon mode table has X'80' as the second byte of the PSERVIC string in the MODEENT macro.

Using the 3270 send/receive facility commands

The 3270 send/receive facility provides two commands, SEND and RECEIVE. This section explains both commands, and the options that you can use with your host computer operating system.

The following notational conventions are used in this section:

- [] The parameter is optional.
- (IND\$FILE only)** The option is supported only if the host computer will be using one of the IND\$FILE programs (see “Host computer requirements” on page 226).
- (APVUFILE only)** The option is supported only if the host computer will be using one of the APVUFILE programs (see “Host computer requirements” on page 226).

The following syntax is common to all host operating systems supported:

For 3270 send:

```
SEND filename [hostid] hostfile [(options)]
```

For 3270 receive:

```
RECEIVE filename [hostid] hostfile [(options)]
```

where:

filename Is the required name and extension of the DOS file to be sent or received. If the SEND or RECEIVE command is issued from a directory other than the one containing the file, you need to specify the drive and path where the file is located.

If you receive a file onto a destination path where another file with the same name and extension already exists, the file you receive adds its contents at the end of the existing file if you specify the APPEND option. If you do not specify the APPEND option, the file you receive replaces the existing one.

hostid Is the 3270 emulator short name. It can take the values A: through E:; the default value is A:.

hostfile Is the required name of the host file.

options Are options specific to each host operating system, as described in the following lists.

VM/CMS SEND command options

These are the options you can specify to send a file from your workstation to a VM/CMS host through your LANDP 3270 emulator:

- APPEND Causes the workstation file to be attached to the end of the host file.
- ASCII Translates a file, assumed to be stored in ASCII format, into EBCDIC in order to be read by the host.
- CRLF Specifies that carriage return and line feed characters are to be recognized as record separators, and deleted before the file is stored on the host storage medium.
- LRECL *n* Specifies the logical record length of host file records. *n* is the number of characters in each record. If you specify a variable length record format (RECFM V), *n* is the maximum record size expected. The default value for new files is 80.
- NEW (IND\$FILE only) Cancels the file transfer if a file with the same name as *hostfile* already exists on the host storage medium. You can use this option to protect against accidental erasure of a previously existing file.
- NOCLEAR Causes the emulator screen contents to remain on screen during the file transfer.
- RECFM *x* Specifies the host file record format. *x* can take one of the following values:
F Fixed length
V Variable length

If you specify the CRLF option for a new host file, the default RECFM value is V; otherwise it is F.
- TIMEOUT=*nn* Specifies the number of seconds (*nn*) to be used as the time-out value during file transfer. If you do not specify this parameter, a time-out value of 15 seconds is used.

VM/CMS RECEIVE command options

These are the options you can specify to receive a file at your workstation from your VM/CMS host through your LANDP 3270 emulator:

- APPEND Causes the host file to be attached to the end of the workstation file.
- ASCII Translates the host file from EBCDIC to ASCII so that it can be read at the workstation.
- BLANK (APVUFILE only) If the CRLF option is also specified, continuous blanks are not deleted from the end of each record. If the CRLF option is not specified, the BLANK option is ignored.
- CRLF Causes the host file transfer program to add carriage return and line feed characters as the two last characters of each line before sending the file to the workstation.

- NOCLEAR** Causes the emulator screen contents to remain on screen during the file transfer.
- TIMEOUT=*nn***
Specifies the number of seconds (*nn*) to be used as the time-out value during file transfer. If you do not specify this parameter, a time-out value of 15 seconds is used.

MVS with TSO SEND command options

These are the options you can specify to send a file from your workstation to a MVS/TSO host through your LANDP 3270 emulator:

- APPEND** Causes the workstation file to be attached to the end of the host data set.
- ASCII** Translates a file, assumed to be stored in ASCII format, into EBCDIC in order to be read by the host.
- BLKSIZE(*n*)**
Specifies the block size of the host data set. *n* is the length of the data block in bytes. The default for new data sets is the LRECL value. If the host data set already exists, this option is ignored.
- CRLF** Specifies that carriage return and line feed characters are to be recognized as record separators, and deleted before the file is stored on the host storage medium.
- LRECL *n*** Specifies the logical record length of host data set records. *n* is the number of characters in each record. If you specify a variable length record format (RECFM V), *n* is the maximum record size expected. The default value for new data sets is 80.
- NEW** (IND\$FILE only) Cancels the file transfer if a data set with the same name as *hostfile* already exists on the host storage medium. You can use this option to protect against accidental erasure of a previously existing data set.
- NOCLEAR** Causes the emulator screen contents to remain on screen during the file transfer.
- RECFM *x*** Specifies the host data set record format. *x* can take one of the following values:
F Fixed length
V Variable length
If you specify the CRLF option for a new host data set, the default RECFM value is V; otherwise it is F.
- RELEASE** (APVUFILE only) Causes unused portions of a data set to be released when file transfer is complete.

3270 send/receive facility, OS/2 DOS Windows NT

SPACE(*n,m*)

Specifies the amount of space to be allocated for a new data set (the option is ignored for an existing data set). *n* units are to be allocated initially, and increments of *m* units allocated subsequently if needed. If this option is present, the *spaceunits* option is also valid. If the SPACE option is not present, SPACE(10,50) AVBLOCK(1000) is assumed.

spaceunits

Specifies the units intended in the SPACE option. It is valid only when the SPACE option is specified.

spaceunits can take one of the following values:

AVBLOCK(<i>n</i>)	Unit of space is a block, where <i>n</i> is the average block size of the records being transferred
CYLINDERS	Unit of space is a cylinder
TRACKS	Unit of space is a track

The default value of *spaceunits* is AVBLOCK(*b*), where *b* is the value you supply in the BLKSIZE option. If BLKSIZE is omitted, AVBLOCK(80) is assumed.

TIMEOUT=*nn*

Specifies the number of seconds (*nn*) to be used as the time-out value during file transfer. If you do not specify this parameter, a time-out value of 15 seconds is used.

UNIT(*unit-type*)

(APVUFILE only) Specifies the magnetic disk unit type to be used for data set allocation.

VOLUME(*serial-numbe*)

(APVUFILE only) Specifies the volume serial number of the magnetic disk to be used for data set allocation.

MVS with TSO RECEIVE command options

RECEIVE

These are the options you can specify to receive a file at your workstation from your MVS/TSO host through your LANDP 3270 emulator:

APPEND	Causes the host data set to be attached to the end of the workstation file.
ASCII	Translates the host data set from EBCDIC to ASCII so that it can be read at the workstation.
BLANK	(APVUFILE only) If the CRLF option is also specified, continuous blanks are not deleted from the end of each record. If the CRLF option is not specified, the BLANK option is ignored.
CRLF	Causes the host file transfer program to add carriage return and line feed characters as the two last characters of each line before sending the file to the workstation.

NOCLEAR Causes the emulator screen contents to remain on screen during the file transfer.

TIMEOUT=*nn*

Specifies the number of seconds (*nn*) to be used as the time-out value during file transfer. If you do not specify this parameter, a time-out value of 15 seconds is used.

MVS with CICS SEND command options

SEND

These are the options you can specify to send a file through your LANDP 3270 emulator from your workstation to a MVS with CICS host:

ASCII Translates a file, assumed to be stored in ASCII format, into EBCDIC in order to be read by the host.

CRLF Specifies that carriage return and line feed characters are to be recognized as record separators, and deleted before the file is stored on the host storage medium.

NOCLEAR Causes the emulator screen contents to remain on screen during the file transfer.

TIMEOUT=*nn*

Specifies the number of seconds (*nn*) to be used as the time-out value during file transfer. If you do not specify this parameter, a time-out value of 15 seconds is used.

MVS with CICS RECEIVE command options

RECEIVE

These are the options you can specify to receive a file through your LANDP 3270 emulator at your workstation from your MVS with CICS host:

APPEND Causes the host file to be attached to the end of the workstation file.

ASCII Translates the host file from EBCDIC to ASCII so that it can be read at the workstation.

BLANK (APVUFILE only) If the CRLF option is also specified, continuous blanks are not deleted from the end of each record. If the CRLF option is not specified, the BLANK option is ignored.

CRLF Causes the host file transfer program to add carriage return and line feed characters as the two last characters of each line before sending the file to the workstation.

DELETE (APVUFILE only) Causes the temporary storage queue containing the transferred data to be deleted when the transfer is complete.

NOCLEAR Causes the emulator screen contents to remain on screen during the file transfer.

TIMEOUT=*nn*

Specifies the number of seconds (*nn*) to be used as the time-out value during file transfer. If you do not specify this parameter, a time-out value of 15 seconds is used.

4707 monochrome display support



The IBM 4707 monochrome display supports three display modes. LANDP for DOS provides a subroutine that allows switching between these three modes. The display mode can also be changed using DOS commands.

The IBM 4707 monochrome display is a compact 9-inch monochrome display. Because of its small size, it is often used when space is limited.

This display provides hardware support for the three display modes:

- *2000-mode*. Displays 25 rows of 80 characters.
- *1000-mode*. Displays 25 rows of 40 characters. Note that the displayed characters are *double width* compared to the displayed characters in 2000-mode. The character height is the same in both modes.
- *480-mode*. Displays 12 rows of 40 characters. Note that the displayed characters are both *double width* and *double height* compared to the displayed characters in 2000-mode. Because of the large character size, the characters fill the entire surface when in 480-mode.

A device driver comes with the IBM 4707 monochrome display. Although using the device driver is optional, the 4707 display always operates in 2000-mode if it is not installed.

The device driver itself operates in two states:

- Enabled
- Disabled

If the device driver is disabled, the 4707 display operates as if the device driver is not installed.

To be able to change 4707 display modes, the video adapter card must be one of the following:

- EGA (enhanced graphics adapter)
- VGA (video graphics array), MCGA (multicolor graphics array), or any other adapter that can emulate the subset of the necessary EGA functions.

The 4707 display mode can be changed using either a LANDP-supplied subroutine or DOS commands.

For information about changing the mode using DOS commands, see “Changing mode using IBM DOS commands” on page 233. For information about using the

LANDP-supplied routine and other programming considerations, see *LANDP Programming Reference*.

Changing mode using IBM DOS commands

If the 4707 display device driver is disabled and the `MODE C040` command is entered at the DOS prompt:

```
A>MODE C040
```

the 1000-mode is entered. If mode change is not supported, an *invalid parameter* error message is displayed.

If the 4707 display device driver is enabled and the `MODE C040` command is entered at the DOS prompt:

```
A>MODE C040
```

the 480-mode is entered.

If the `MODE C080` command is entered at the DOS prompt:

```
A>MODE C080
```

the 2000-mode is entered again. Then, the 1000-mode is no longer available until the device driver is disabled. Note that it is not possible to enable or disable the device driver through IBM DOS. It must be done using the LANDP-supplied subroutine.

Multiple virtual DOS machine relays



The LANDP for OS/2 and Windows NT multiple virtual DOS machine (MVDM) relays let LANDP for DOS applications and emulators run on LANDP for OS/2 and Windows NT workstations.

The relays let the LANDP for OS/2 and Windows NT supervisor process requests from LANDP for DOS applications that are running in an OS/2 or Windows NT virtual DOS machine (VDM). For information on the virtual DOS machine, refer to your operating system documentation.

The LANDP for OS/2 MVDM relay supports running the following emulators in an OS/2 VDM:

- LANDP for DOS 3270 emulator (not supported in DBCS mode)
- LANDP for DOS 3287 printer emulator (not supported in DBCS mode)
- FBSI banking interactive workstation program (BIWP), if the FBSI is installed.

facilities and utilities

The LANDP for Windows NT MVDM relay supports running the following emulators in a Windows NT VDM:

- LANDP for DOS 3270 emulator (not supported in DBCS mode)
- LANDP for DOS 3287 printer emulator (not supported in DBCS mode)
- FBSI banking interactive workstation program (BIWP), if the FBSI is installed.

The number of emulators loaded in VDMs cannot exceed the maximum number of emulators that can be loaded in one LANDP for DOS workstation. Up to five 3270 emulators, four 3287 printer emulators, and one BIWP are supported. These emulators are loaded using the load servers inline (LSI) utility program. For more information about LSI, refer to *LANDP Installation and Customization*.

Because the LANDP for OS/2 and Windows NT MVDM relay supports multiple VDMs, multiple LANDP for DOS applications can be run with each one operating in a different virtual DOS machine.

Applications written using the FBSS low-level INT X'7F' API, LANDP common API, LANDP 3270 emulator API, or 3270 emulator HLLAPI can run on a virtual DOS machine.

See the *LANDP Programming Reference* for more information about the functions associated with the MVDM relay.

Microsoft Windows 3.1 support



LANDP for DOS, DOS applications, and Windows applications that request services from LANDP servers can run under Microsoft Windows 3.1 or Windows for Workgroups 3.11.

LANDP for DOS runs under Microsoft Windows 3.1 only in the 386 enhanced mode. This mode provides access to the virtual memory capabilities of the supported Intel processors and lets Microsoft Windows applications use more available memory. It also supports multitasking of non-Windows applications.

The LANDP Microsoft Windows 3.1 support allows applications (Windows or non-Windows) and other Windows or non-Windows applications to use the graphical Windows user interface.

Note however that LANDP for DOS cannot operate in DBCS mode under Microsoft Windows 3.1..

The IBM Windows Connection program is also supported by LANDP. The LANDP high level language API (HLLAPI) allows the IBM Windows Connection to run like any other supported HLLAPI products listed in the IBM Windows Connection documentation. However, only one emulation session is supported. Refer to IBM Windows Connection and Microsoft Windows documentation for more details.

Considerations for Microsoft Windows 3.1 support

Consider the following when using the LANDP for DOS support for Microsoft Windows 3.1:

- The IBM 3270 Emulator LAN Management Program is not supported. This means that alerts generated by the LAN adapters will not be collected by the system manager server.
- Only eight Windows or non-Windows applications can be supported.
- The SHRDIRDD and FILESYS commands should be loaded, if needed, in the LANDP application virtual machine, just after EHCWVDMI and before the LANDP application. When using remote LOG for the shared-file server, they should be loaded in the LANDP kernel VM as well.
- No utility, other than ECHWKDE (which improves performance), or additional programs should be loaded in the LANDP Kernel VM. Any other programs or utilities should be started in other VMs.
- To move from the LANDP application to the 3270 emulator, two hotkeys are required. The first one to move from the LANDP application VM to the LANDP kernel VM; the second one to activate the 3270 emulator. If the application uses the Simulate Hotkey (EE) supervisor function, only the first hotkey is needed to activate the 3270 emulator. In any case, two hotkeys are required to move backwards. The same explanation applies to the following LANDP functional areas:
 - Operator interface
 - System manager operator
 - Trace tools.
- All the LANDP operators should be in the background while working with the LANDP application.
- The token-ring data link control (TRDLC) server cannot be run under Windows for Workgroups 3.11.
- The IBM X.25 Interface Co-Processor/2 adapter used by the X.25 data link control (X25DLC2) server is not compatible with Microsoft Windows 3.1.
- Although the IBM 4717 MSR/E and IBM 4718 PIN Pads are not supported, serially connected IBM 4777 MSR/E and IBM 4778 PIN pads are supported.

These restrictions apply only in the workstations where LANDP for DOS and Microsoft Windows 3.1 coexist.

Some programming restrictions apply. All applications must request an IN function before requesting any services from another LANDP service. To end a LANDP session, the application should request an EJ function to the supervisor to free all the resources attached to the application.

Some installation and configuration restrictions also apply.

general utilities

See the *LANDP Installation and Customization* for information on how to configure Windows 3.1 support, the *LANDP Programming Reference* for information about functions used with the Windows 3.1 environment, and the *LANDP Installation and Customization* for information about tuning and using Windows applications in a LANDP workgroup.

General utility programs



This section describes the general utility programs provided with the LANDP family.



Some of the utility programs are closely related to specific LANDP servers, and are described in the appropriate server chapters elsewhere in this book.



Others are related to loading LANDP onto the five platforms, DOS, OS/2, Windows NT, and AIX. These are described in the *LANDP Installation and Customization*.

VERSION utility program



LANDP provides a VERSION utility program to determine the current version number and the last authorized program analyses report (APAR) or program temporary fix (PTF) that has been applied to a program in a LANDP component.



This utility program displays the name, the version, and the APAR/PTF level of each component and writes this information into text file.

See the *LANDP Installation and Customization* for more information about this program.

APPLYFIX utility program



The APPLYFIX utility program is used to apply program fixes to LANDP for DOS, OS/2, and Windows NT, and to maintain records of the installed changes.



LANDP for AIX fixes must be installed using the standard functions provided by the IBM AIX INSTALLP command or the SMIT program.

See the *LANDP Installation and Customization* for more information about this program.

System verification utility programs



The system verification utility programs are used to test and verify the operation of LANDP servers. The names of the system verification utility programs are:

- SVPCPRB.EXE for LANDP for DOS and OS/2
- SVPCPRBN.EXE for LANDP for Windows NT
- SVPCPRBW.EXE for Microsoft Windows 3.1
- dczysvp or dczyxsvp (Motif version) for LANDP for AIX

These programs are used to:

- Verify the correct operation of the LANDP program components without creating a special application.
- Test the servers. The system verification programs supplement the standard personal computer system diagnostic procedures and the diagnostic procedures for the adapters.
- Train LANDP application developers.

See the *LANDP Programming Reference* for more information.

Displaying server status



In OS/2 and Windows NT, LANDP provides the EHCINFO.EXE program to display a LANDP Status window with information about the status of the servers loaded in the workstation where EHCINFO.EXE runs. The status of each server is refreshed each time the user focus is on the Status window or when you press any key.

In DOS, OS/2, and Windows NT, LANDP provides the EHCCINFO.EXE program to list information about the status of the servers loaded in the workstation where EHCCINFO.EXE runs. The command is entered on the DOS command line and the listing follows. The contents of the listing are the same as that provided in the LANDP Status window by LANDP for OS/2 and Windows NT.

In OS/2 and Windows NT, the EHCINFO.EXE program can be included at the end of the AUTOFBSS.COM command file that is used to start LANDP for OS/2 or Windows NT. You can write an AUTOFBSS.COM command that meets your own requirements. An example follows overleaf:

general utilities

```
LOADER SPV /AA
IF ERRORLEVEL 1 GOTO END
LOADER EHCTACW
IF ERRORLEVEL 1 GOTO END
LOADER LAN
IF ERRORLEVEL 1 GOTO END
GOTO OK
:END
EHCFREE SPV
EXIT
:OK
EHCINFO
```

Output from EHCINFO and EHCCINFO

A sample output screen from EHCINFO by LANDP for OS/2 and Windows NT, or a listing from EHCCINFO by LANDP for DOS, OS/2, and Windows NT, might look like this:

Local access	Server Name	Server Alias	PC	System	Status	PID
Yes	SPV	SPV	JA	OS/2	ON	238
Yes	EHCDBTR	EHCDBTR	JC	DOS	OFF	0
Yes	EHCSQL04	EHCSQL04	JA	OS/2	OFF	0
Yes	EHCSVPUE	EHCSVPUE	JA	OS/2	ON	244
No	LAN	LAN	JA	OS/2	ON	240
No	EHCVMGR	EHCVMGR	JA	OS/2	ON	243
No	EHCINFO	EHCINFO	JA	OS/2	ON	245

where:

Local access

Yes Indicates that the program (a server or its client) was set up explicitly during customization process and that calls can be made to it. The sole exception is SPV, the supervisor, which always appears with Yes in this column.

Some servers may have several entries. For example, you may have specified SNA## (the SNA server) in your configuration, but SNA01 and SNA33 might appear in the Server Name column, each with Yes in the Local access column.

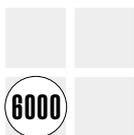
No Indicates one of the following:

- that the program was not set up explicitly during customization, but has called the supervisor. For example, EHCINFO itself calls the supervisor and therefore appears on the output screen with No in the Local access column.
- that no calls can be made to the program (for example, EHCTACW) even though it was set up explicitly during customization.

Server Name Is the name of the server being used.

Server Alias	Is another name by which you can refer to a server. The name was chosen during customization.
PC	Is the identifier for the workstation on which the server is running.
System	Is the type of operating system being used.
Status	Tells whether the related system is running and accessible (ON) or not (OFF).
PID	Is the identifier for the process you are running. It is unique among servers running under the same operating system on the same workstation.

Displaying LANDP for AIX status



To display LANDP for AIX status:

1. Access the Diagnose and Query LANDP for AIX panel in SMIT.

To access this panel, select **Communications Applications and Services** on the SMIT main menu and then select the **IBM AIX LAN Distributed Platform/6000 (LANDP for AIX)** option on the subsequent menu.

2. Select **Display active LANDP for AIX servers and applications** from the Diagnose and Query LANDP for AIX panel.

Translating between ASCII and EBCDIC



The LANDP family include translation routines to enable an application to translate between IBM PC ASCII code page CP 850 and the IBM EBCDIC MLP code page CP 500. However, they can be modified to meet any specific need. The source is also provided so that you can change the routines.

See the *LANDP Programming Reference* for information about the functions associated with the translation routines.

ASCII-EBCDIC translation server

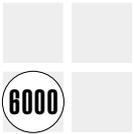


The ASCII-EBCDIC translation server enables a LANDP application to translate any string (DBCS, SBCS, or mixed) from ASCII to EBCDIC or vice versa. This server supports translation on workstations with Taiwan, Korea, or People's Republic of China code pages.

The following translation servers are provided:

Country	PC code page	Host code page
Korea	949	933
People's Republic of China	1381 1381	935 1388
Taiwan	938 950	937 937

LANDP for AIX record definition utility program



The LANDP for AIX record definition utility program is used to define the record formats to be used by the electronic journal and the store-for-forwarding servers.

This utility program must be run before adding any records to the electronic journal and store-for-forwarding data sets. The client/server mechanism, the query server, and the electronic journal or the store-for-forwarding server must be loaded first.

If the utility receives any return code, the code is displayed and the program stops processing. If the database returns a message as a result of an error, this message is also displayed.

You can also define record formats with your applications, by using the functions provided by the electronic journal and the store-for-forwarding servers for that purpose. For more information, refer to the *LANDP Programming Reference*.

Syntax

If you want to define record formats, invoke the program as follows:

```
dczyrdf servername path
```

where:

servername

Is the name of the server for which you are defining the record format.

path Is the path where the input file containing the record definitions is located.

If you want to delete record definitions, invoke the program as follows:

```
dczyrdf servername -d formatname
```

where:

servername

Is the name of the server for which you are deleting a record format definition.

formatname

Is the name of the format you want to delete.

If the server name during the definition of record formats is CHECK, only a syntax check is performed. It is not necessary to load LANDP for AIX to perform this check.

Input file

The input file, which contains the record format definitions, is an ASCII file that you generate with a standard editor. You define the record formats by using the FORMATNAME, COLUMN, and ENDFORMATNAME keywords.

FORMATNAME: This keyword indicates the beginning of a record format definition. The syntax is as follows:

```
FORMATNAME formatname
```

where:

formatname

Is the name of the format to be defined. It can have a maximum of 8 characters.

COLUMN: This keyword indicates one field in the record format definition. The syntax is as follows:

```
COLUMN columnname columntype columnlength
```

where:

columnname

Is the name you assign to this column. It can have a maximum of 18 characters.

columntype

Is the type of data contained in this column. You can define the following types of data:

- BINARY
- CHAR
- DATE
- DECIMAL
- FLOAT
- INT
- NULLBINARY
- NULLCHAR

AIX record definition utility

- NULLDATE
- NULLDECIMAL
- NULLFLOAT
- NULLINT
- NULLSMALLINT
- NULLTIMESTAMP
- SMALLINT
- TIMESTAMP

Records defined as BINARY are translated into RAW type in Oracle7 and into BYTE type in Informix.

Data types NULLSMALLINT and SMALLINT are not supported in Oracle7, therefore, these data types are translated into NULLINT and INT respectively.

columnlength

Is the length of the column. This value applies to the following types of data:

- BINARY
- CHAR
- DECIMAL
- NULLBINARY
- NULLCHAR
- NULLDECIMAL
- NULLTIMESTAMP (optional)
- TIMESTAMP (optional)

All other types of data have a fixed column length.

If you define the data type as TIMESTAMP, you can specify the FROM/TO interval. To specify the interval, use the first byte for the FROM value and the second byte for the TO value. Refer to the *LANDP Programming Reference* for further information about specifying the FROM and TO values. If you do not specify this interval, the program defaults to YEAR TO FRACTION. In Oracle7, the FROM and TO values are always YEAR TO SECOND.

ENDFORMATNAME: This keyword indicates the end of a record format definition. The syntax is as follows:

```
ENDFORMATNAME
```

Keywords must start in the first position of the line in the file. Empty lines or lines beginning with # are ignored. You can define multiple record formats in one file.

If syntax errors are found in the file, the utility displays the number of the line where errors have been encountered and stops processing.

Example: The following is an example of an input file containing two record format definitions.

```

FORMATNAME      alltypes

COLUMN          FIRSTNAME      CHAR 10
COLUMN          ZIPCODE        SMALLINT
COLUMN          REVENUE        INT
COLUMN          IN_COMPANY     DATE
COLUMN          TAX_RATE       FLOAT
COLUMN          LAST_UPDATE    TIMESTAMP 70 (X'0406'. DAY TO HOUR)
COLUMN          NET_INCOME     DECIMAL (6,2)
COLUMN          PICTURE        BINARY 100

ENDFORMATNAME

```

```
#####
```

```

FORMATNAME      nulltypes

COLUMN          FIRSTNAME     NULLCHAR 10
COLUMN          ZIPCODE       NULLSMALLINT
COLUMN          REVENUE       NULLINT
COLUMN          IN_COMPANY    NULLDATE
COLUMN          TAX_RATE      NULLFLOAT
COLUMN          LAST_UPDATE   NULLTIMESTAMP
COLUMN          NET_INCOME    NULLDECIMAL (6,2)
COLUMN          PICTURE       NULLBINARY 100

ENDFORMATNAME

```

AIX record definition utility

Chapter 10. Migration

This chapter describes the migration paths provided by IBM LANDP. These include:

- Query server from FBSS/2 to LANDP for OS/2
- Shared-file server to LANDP for OS/2 query server
- Shared-file server and SQL data to LANDP for AIX query server

Each migration path may involve several tasks for which IBM LANDP provides utilities. The utilities required to perform the migration tasks are also described in this chapter.

Query server from FBSS/2 to LANDP for OS/2



For products which support forward recovery for the database manager (for example, IBM Database Server for OS/2 Warp, Version 4) the query server forward recovery and logging are inhibited by default. You can switch them on by adding the /EL parameter to the loading statement of the server. However, for performance reasons, it is recommended that you leave them off (this is the default) and remake your backup and recovery procedures so that they exploit this new database manager feature.

If you want to use this database manager feature, you have to modify some parameters in the default database manager configuration. For more information, refer to the appropriate documentation for the product. See "Bibliography" on page 285.

Shared-file server to LANDP for OS/2 query server



If you want to migrate your shared-file server data structures and data files to the LANDP for OS/2 query server environment and you have already migrated your application, do the following:

1. Migrate the shared-file server data structures (DBD and PCB) to SQL tables using the utilities provided by LANDP for OS/2:

- EHCMGR1

This program reads the .PCB and .DBD definition files and optionally the record definition file, RDF.CFG, and creates an ASCII file with the correct SQL sentences to create the SQL tables.

Refer to “EHCMGR1” on page 249 for detailed information.

- EHCOS2Q (for LANDP for OS/2 workstations) or EHCDOSQ (for LANDP for DOS or FBSS (DOS) workstations).

These programs read the ASCII files created by EHCMGR1 and use the query server to create the SQL tables.

Refer to “EHCOS2Q and EHCDOSQ” on page 251 for detailed information about these programs.

2. Migrate existing shared-file server data files to SQL files using the utilities provided by LANDP for OS/2:

- EHCMGR2

This program runs on LANDP for DOS and reads the shared-file server data files and the output files from the data structure migration program (EHCMGR1). It uses the query server to move data to the SQL files.

Refer to “EHCMGROS and EHCMGR2” on page 251 for detailed information.

- EHCMGROS

This program runs on LANDP for OS/2 and is functionally equivalent to EHCMGR2.

Refer to “EHCMGROS and EHCMGR2” on page 251 for detailed information.

LANDP for OS/2 query server to LANDP for Windows NT ODBC query server



For migration to the LANDP for Windows NT ODBC query server to succeed, your applications must use only the LANDP for OS/2 query server SQL interface. When using the ODBC query server, a return code AD (the function is not supported) is received by any calls for the shared file interface.

When migrating your applications and data to use the ODBC query server, you need to be aware of the following points. Not all RDBMS systems use the same base data types and therefore data being passed between the application and ODBC query server may not be in the types expected. Oracle maps ANSI data types (which map closely to DB2) to their internal data types. For example, if Oracle maps an integer to a number (38), ODBC sees the number type as a DECIMAL type with a precision of 38. And thus the ODBC query server would view this as a decimal type. If the ODBC query server is processing the request using the ODBC data type interface, then a character string of 39 (precision plus the sign) is returned. However, if the ODBC query server is processing the request using the LANDP for OS/2 query server data type interface, the value returned could be incorrect. It is advisable to use the ODBC data type interface in situations where the underlying RDBMS data types are not ANSI data types.

The usage of date and time types depends on the selected data type interface and the ODBC device driver used. ODBC specifies the format yyyy-mm-dd for date types and hh:mm:ss for time types, although the ODBC device driver could return a different format. For example, the Microsoft ODBC device driver for the SQL Server has the ability to return the date and time types in local format. Because the LANDP for OS/2 query server's date format is directly related to the local environment set up of the DB2, the ODBC query server can not guarantee the same format for date and time types.

To migrate your LANDP for OS/2 query server database to your LANDP for Windows NT ODBC query server database, follow the migration path given with your database management software.

To aid application migration, the OPENODB utility (see "OPENODB - creating connection criteria" on page 122) is provided so that the extended GF function can still be called. OPENODB must be called before your application is started. After the OPENODB has been issued, an application that calls the GF function receives a YP return code, which means that the shared file is already open.

Shared-file server and SQL data to LANDP for AIX query server

OS/2 data migration utilities



You can migrate the LANDP for DOS and OS/2 shared-file server database and the LANDP for OS/2 SQL database to the LANDP for AIX environment using the MIG.EXE utility. For more information about this program, refer to “MIG.EXE data migration utility” on page 252.

If the source for migration is a shared-file server database, the utility reads the .PCB and .DBD definition files and optionally the record definition file, RDF.CFG, and converts the data and data definitions written for the shared-file server database to the corresponding query server data and data definitions.

The utility creates the SQL data definition statements and processes them using the target query server.

LANDP for OS/2 data migration utilities



The LANDP for OS/2 query server provides utilities to support data migration from the shared-file server environment to the LANDP for OS/2 query server environment. Descriptions of those utilities and their syntax are presented in the following sections.

If you use the record definitions from the RDF.CFG file, an equivalent data type will be defined for each column from the input field definitions. Table 31 shows how RDF defined fields are converted to SQL columns.

Table 31 (Page 1 of 2). EHCMGR1 field type conversion: RDF to SQL

RDF Field Definition					Create Statement
TYPE	ABRV	LENGTH	SIGN	DEC	Resulting Data Type
Character	C	x<255	2 ¹	0 ¹	CHAR(x)
Long character	C	x>254	2 ¹	0 ¹	VARCHAR(x) ³
ASCII numeric	N	x<255	2 ²	0 ²	CHAR(x)
Signed ASCII numeric	SN	x<255	1 ²	0 ²	CHAR(x)
Long ASCII Numeric	N	x>254	2 ²	0 ²	VARCHAR(x) ³
Integer	I	2 ¹	2 ²	0 ¹	SMALLINT
Signed integer	SI	1 ¹	1 ²	0 ¹	SMALLINT
Inverted binary	B	2 ¹	1 ¹	0 ¹	SMALLINT
Binary hexadecimal	H	x	2 ¹	0 ¹	CHAR(x) FOR BIT DATA
Long integer	L	4 ¹	2 ²	0 ¹	INTEGER
Signed long integer	SL	4 ¹	1 ²	0 ¹	INTEGER

RDF Field Definition					Create Statement
TYPE	ABRV	LENGTH	SIGN	DEC	Resulting Data Type
Unsigned Packed	P	x	2	y	DECIMAL(x,y)
Signed Packed	P	x	1	y	DECIMAL(x-1,y)
Notes:					
¹ Only allowed value.					
² These values are not taken into account for the conversion.					
³ Data in SQL type VARCHAR will be preceded by a two-byte length field.					

TYPE Is the format in which the field is stored.

ABRV Is the abbreviation corresponding to the type field.

LENGTH
Is the field length.

SIGN Indicates whether the numeric field has a sign. 1 means Yes, 2 means No.

DEC Is the number of decimal places.

EHCMGR1

The EHCMGR1 program is provided to migrate data structures from the shared-file server environment to the SQL environment, where the query server works. It uses the shared-file server data structure definition files, with extensions DBD and PCB, as input. If the complete record layout has been defined during customization, the EHCMGR1 program uses the RDF.CFG file created during customization, unless otherwise specified.

The EHCMGR1 program creates two files:

- A file with extension SQL, which is an ASCII file containing SQL sentences to create the SQL data definition tables.
- File with extension EHC, which specifies the changes required for the shared-file server data that is being migrated to SQL data.

The EHCMGR1 program is an IBM DOS program and does not require LANDP for DOS. It is called as follows:

```
EHCMGR1 [/E] [/cnfname]
```

where:

E Is a parameter specifying that the RDF.CFG file will not be used. Then, all fields will be migrated as *char fields* or *varchar fields*, if longer than 254 bytes.

cnfname

Is the complete path and file name of the shared-file server profile files with extensions DBD and PCB.

OS/2 data migration utilities

The EHCMGR1 program displays error messages, which usually specify that a file has been damaged or that an error in the definition of the validated field has been detected.

Those messages that are not self-explanatory are presented in the table below. For all of them, the action to be taken is either to replace the file damaged or to verify the specified field. You should also verify that the shared-file server profile files, with extensions DBD and PCB, match with the RDF.CFG file used.

Table 32. EHCMGR1 error messages

Message	Cause
DBD file record without DBD name	DBD file damaged
x DBD name without record name	DBD file damaged
x DBD record name not found in RDF.CFG file	DBD file or RDF.CFG file damaged
x DBD definition area not found in RDF.CFG file	RDF.CFG file damaged
x DBD name with zero record length	RDF.CFG file damaged
x record not type F data format in RDF.CFG file	RDF.CFG file damaged
x record length does not match with field length in RDF.CFG file	DBD file or RDF.CFG file damaged
x DBD name with not valid key name	DBD file damaged
x record with no key field in RDF.CFG file	
x field with not valid length in RDF.CFG file	DBD file or RDF.CFG file damaged
x field type B with length different from 2 and 4 in RDF.CFG file	RDF.CFG file damaged
x field type I with length different from 2 in RDF.CFG file	RDF.CFG file damaged
x field type J with length different from 4 in RDF.CFG file	RDF.CFG file damaged
x field with store format not valid in RDF.CFG file	RDF.CFG file damaged

EHCOS2Q and EHCDOSQ

The EHCOS2Q program is provided to create the SQL data definition tables. It is an interactive program that uses the query server to process SQL sentences. The source code is supplied as a sample on how to use the query server.

The EHCOS2Q program is called as follows:

```
EHCOS2Q [servername]
```

where:

servername

Is the complete name of the LANDP for OS/2 server that processes the requests. The default is EHCSQL01.

You can also create the SQL data definition tables from a LANDP for DOS workstation. In this case, the workstation must have LANDP for DOS and IBM FBSS 2.2.1 or higher installed, and must be customized to receive services from the query server. Then, you must call the EHCDOSQ program, as follows:

```
EHCDOSQ [servername]
```

where:

servername

Is the complete name of the LANDP for OS/2 server that processes the requests. The default is EHCSQL01.

When using the EHCOS2Q or EHCDOSQ program for migration, you must redirect the input to the configuration file, with extension SQL, created by the EHCMGR1 program. You can redirect the EHCOS2Q or EHCDOSQ program input and output, using the standard operating system syntax (<infile and >outfile). In this case, the program is called as follows:

```
program [servername] <cfgname
```

where:

program

Is either EHCOS2Q or EHCDOSQ.

servername

Is the complete name of the LANDP for OS/2 server that processes the requests. The default is EHCSQL01.

cfgname

Is the name of the configuration file.

EHCMGROS and EHCMGR2

The EHCMGROS program is provided to migrate shared-file server records to SQL table rows, using the shared-file server. This is the last migration step.

MIG.EXE data migration utility, OS/2 DOS AIX

The EHCMGROS program is called as follows:

```
EHCMGROS [/serverid] [/cnfname]
```

where:

serverid

Is a unique alias specifier for the query server.

cnfname

Is the complete path where the following files are located:

filename.DBD

filename.PCB

filename.EHC

The actual data files must be located in the path that was defined during the shared-file server customization.

You can also migrate shared-file server records to SQL table rows from a LANDP for DOS workstation. In that case, the workstation must have LANDP for DOS, IBM FBSS 2.2.1 or higher, installed and be customized to receive services from the query server. Then, you must call the EHCMGR2 program, as follows:

```
EHCMGR2 [/serverid] [/cnfname]
```

where:

serverid

Unique alias specifier for the query server.

cnfname

Complete path where the following files are located:

cnfname.DBD

cnfname.PCB

cnfname.EHC

MIG.EXE data migration utility



The MIG.EXE utility can be run under LANDP for DOS or OS/2 to migrate the following:

- LANDP for DOS or OS/2 shared-file data to LANDP for AIX SQL data
- LANDP for OS/2 query server data to LANDP for AIX SQL data



LANDP must be running on both the source and target workstations, because file transfer within the workgroup is performed using LANDP capabilities. The configuration used for migration must include the database to be migrated and the target one. In the configuration used for migration, you must use the LANDP for DOS and OS/2 components from the current version of LANDP.

Note: The data format of a field *may* be changed by the migration. You should make sure that applications and/or PCBs are then changed accordingly.

Migration utility menu

The migration utility uses a common main menu which is shown, together with sample entries, below. The entries that are required will be given when each type of data migration is explained.

```

Source Server Name:      SHFILE01
Source Server Type:     [■] Shared-File Server   [ ] Query Server
Target Server Name:     EHCSQL01
Target LANDP Program:   [■] LANDP for AIX
Qualifier:              |          |          [ ] [ ] Date Format
Collection Name:        |          |
DBD Path:               C:\SAMPLES\SHFILES\SAMPLE.DBD
PCB Path:               C:\SAMPLES\SHFILES\SAMPLE.PCB
RDF Path:               C:\SAMPLES\SHFILES\RDF.CFG

F1=Help                F2=Process                F3=Quit
F4=Toggle Server       F5=Toggle LANDP Vers.
    
```

Figure 48. MIG.EXE main menu

Files required to perform a migration

The following files are located on the “customization workstation” (the workstation at which the installation of LANDP was performed):

- MIG.EXE (the migration utility)
- DMPANLB.PNL (the panel library)
- DM*.DAT (the message files)

These files are located in the following directories, on the drive and path that you have used for your installation:

LANDP for DOS	EHCD500
LANDP for OS/2	EHC0500

Migrating shared-file data to SQL data

This section explains how to migrate LANDP for DOS and OS/2 shared-file data to LANDP for AIX query server data.

To migrate the shared-file data, do the following:

1. Log on to the LANDP workstation at which the shared files to be migrated are installed. You must repeat the complete migration at each workstation that contains shared files to be migrated.
2. Copy the migration utility files (listed in “Files required to perform a migration” on page 253) from the customization workstation to the workstation containing the shared-files to be migrated.
3. Enter MIG to start the migration. The main menu is displayed. Enter the following:
 - The source server name at which the migration is running (for example SHFILE01).
 - The target server name at which the SQL data is to be created (for example EHCSQL01).
 - The target LANDP program. Use F5 or the mouse to move the “checkbox” to either LANDP for OS/2 or AIX.
 - The DBD path (path/file name of DBD files). The DBD path was defined during customization for this workstation.
 - The PCB path (path/file name of PCB files). The PCB path was defined during customization for this workstation.
 - The RDF path (path/file name of RDF file). The RDF file was created during customization.

The date format and qualifier are not required.

4. Press F2. A selection panel is displayed, containing the shared-file DBDs that can be selected for migration.
5. Make your selection of the DBDs to be migrated. Use F6 or the mouse to make your selections.
6. Press F2 to start migrating the selected tables.

The utility migrates the data shown in the tables: Table 33 on page 255 and Table 34 on page 256. For a comprehensive list of field types supported by IBM LANDP, refer to the query server chapter in the *LANDP Programming Reference*.

7. After the migration has completed, check the file produced by the migration utility, EHCERR.LOG for any error information.

Shared-File Field Definition					Create Statement
TYPE	ABRV	LENGTH	SIGN	DEC	Resulting Data Type
Character	C	x<255	NO	0	CHAR(x)
Long character	C	x>254	NO	0	VARCHAR(x) ¹
ASCII numeric	N	x<255	NO	0	CHAR(x)
Signed ASCII numeric	SN	x<255	YES	0	CHAR(x)
Long ASCII numeric	N	x>254	NO	0	VARCHAR(x) ¹
Integer	I	2	NO	0	SMALLINT
Signed integer	SI	2	YES	0	SMALLINT
Inverted binary	B	2	YES	0	SMALLINT
Binary hexadecimal	H	x	NO	0	CHAR(x) FOR BIT DATA
Long integer	L	4	NO	0	INTEGER
Signed long integer	SL	4	YES	0	INTEGER
Unsigned packed	P	x	NO	y	DECIMAL(x,y)
Signed packed	SP	x	YES	y	DECIMAL(x,y)

¹ Data in SQL type VARCHAR will be preceded by a two-byte length field.

MIG.EXE data migration utility, OS/2 DOS AIX

Table 34. Field Conversion: Shared-File to LANDP for AIX SQL

Shared-File Field Definition					Create Statement
TYPE	ABRV	LENGTH	SIGN	DEC	Resulting Data Type
Character	C	x	NO	0	CHAR(x)
ASCII numeric	N	x	NO	0	CHAR(x)
Signed ASCII numeric	SN	x	YES	0	CHAR(x)
Integer	I	2	NO	0	SMALLINT
Signed integer	SI	2	YES	0	SMALLINT
Inverted binary	B	2	YES	0	SMALLINT
Binary hexadecimal	H	x	NO	0	CHAR(x)
Long integer	L	4	NO	0	INTEGER
Signed long integer	SL	4	YES	0	INTEGER
Unsigned Packed	P	x	NO	y	DECIMAL(x,y)
Signed Packed	P	x	NO	y	DECIMAL(x,y)

- TYPE** Is the format in which the field is stored.
ABRV Is the abbreviation corresponding to the type field.
LENGTH Is the field length.
SIGN Indicates whether the numeric field has a sign.
DEC Is the number of decimal places.

Migrating LANDP for OS/2 SQL data

This section explains how to migrate LANDP for OS/2 SQL data to LANDP for AIX SQL data.

To migrate the SQL data, perform the following:

1. Log on to a LANDP workstation that contains SQL data to be migrated. You can run the complete migration for all LANDP for OS/2 workstations containing SQL data to be migrated from the same LANDP for OS/2 workstation.
2. Copy the migration utility files (listed in "Files required to perform a migration" on page 253) from the customization workstation to the workstation from which you are going to perform the complete migration.
3. Enter MIG to start the migration. The main menu that is shown on page 253 is displayed. Enter the following:
 - The source server name at which the migration is running (for example EHCSQL02).
 - The source server type. Use F4 or the mouse to position the "checkbox" on the entry for Query Server.

- The target server name at which the SQL data is to be created (for example EHCSQL04).
 - The target LANDP program: Use F5 or the mouse to move the “checkbox” to LANDP for AIX.
 - An optional Qualifier. This is the qualifier assigned to SQL tables during their generation. If you specify a qualifier, only the tables belonging to this qualifier are displayed when F2 is pressed.
 - A date format: A number corresponding to one of the date formats supported. If you require a list of the possible date formats then Press F1 (they are then listed under the “Date Format” heading).
4. Press F2. A selection panel is displayed, containing the SQL tables which can be selected for migration.
 5. Make your selection of the tables to be migrated. You can use either F6 or the mouse to make your selection.
 6. Press F2 to start the migration of the selected tables. The fields are migrated as shown in the following table. Note however that only fields that are supported by IBM LANDP can be migrated. For a comprehensive list of field types supported by LANDP, refer to the query server chapter in the *LANDP Programming Reference*.
 7. After the migration has completed, check the file produced by the migration utility, EHCERR.LOG for any error information.
 8. If another LANDP for OS/2 workstation in the LANDP workgroup contains SQL data to be migrated, then repeat steps 3 to 7. In step 3, you enter in the source server name field the server name installed at the next workstation (for example EHCSQL06).

<i>Table 35. Field Conversion: LANDP for OS/2 to LANDP for AIX</i>	
LANDP for OS/2	LANDP for AIX
DATE	DATE
TIME	CHAR(8)
TIMESTAMP	DATETIME YEAR TO FRACTIONS(5)
VARCHAR(x)	CHAR(x)
CHAR(x)	CHAR(x)
FLOAT	FLOAT
DECIMAL(x,y)	DECIMAL(x,y)
INTEGER	INTEGER
SMALLINT	SMALLINT

Appendix A. 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 information 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 described in this information. The furnishing of this information 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 documentation or non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those documents or Web sites. The materials for those documents or Web sites are

not part of the materials for this IBM product and use of those documents or Web sites is at your own risk.

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 United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire,
England
SO21 2JN.

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 information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Trademarks and service marks

The following terms are trademarks of the IBM Corporation in the United States or other countries, or both:

ACF/VTAM	MQSeries
AIX/6000	MVS/XA
AnyNet	OfficeVision
Application System/400	OfficeVision/MVS
AS/400	OfficeVision/VM
AT	Operating System/2
BookManager	Operating System/400
C/2	OS/2
CICS	OS/390
CICS OS/2	Presentation Manager
CICS/ESA	RETAIN
COBOL/2	RISC System/6000
Common User Access	RS/6000
CUA	S/390
DB2 Universal Database	SecureWay
Distributed Database	SP
Connection Services/2	ThinkPad
Distributed Relational	VisualAge
Database Architecture	VisualGen
DRDA	VM/ESA
Extended Services	VSE/ESA
IBM	VTAM
IBMLink	WIN-OS/2
IMS/ESA	Workplace Shell
LAN Distance	WebSphere
LANDP	Xstation Manager
Macro Assembler/2	XT
Micro Channel	

Lotus is a registered trademark of Lotus Development Corporation in the United States and other countries.

Tivoli and NetView are registered trademarks of Tivoli Systems Inc. in the United States and other countries. In Denmark, Tivoli is a trademark licensed from Kjøbenhavn Sommer - Tivoli A/S.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc in the United States and/or other countries.

Microsoft, Windows, Windows NT, Windows 2000, and the Windows logo are registered trademarks of Microsoft Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

Intel is a registered trademark of Intel.

PC/TCP and NetManage are registered trade marks of NetManage Inc..

Other company, product, and service names may be trademarks or service marks of others.

Glossary

This glossary includes abbreviations, terms, and definitions used in the IBM LANDP Licensed Programs Family publications. It does not include all terms previously established for IBM networks, programs, operating systems, or other IBM products.

If you do not find the term you are looking for, refer to the *IBM Dictionary of Computing*.

This glossary includes terms and definitions from the following sources:

- The *IBM Dictionary of Computing*, New York: McGraw-Hill, copyright 1994 by International Business Machines Corporation. Copies may be purchased from McGraw-Hill or in bookstores.
- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The ANSI/EIA Standard—440-A, *Fiber Optic Terminology*. Copies may be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue, N.W., Washington, DC 20006. Definitions are identified by the symbol (E) after the definition.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

Definitions that are specific to IBM products are so labeled, for example, "In LANDP," or "In SNA."

A

abend. Abnormal end of task.

abnormal end of task (abend). Termination of a task before its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing.

abstract class. A class that provides common information for subclasses, and that therefore cannot be instantiated. Abstract classes provide at least one abstract method.

abstract method. A method with a signature, but no implementation. You provide the implementation of the method in the subclass of the abstract class that contains the abstract method.

account. In the AIX operating system, the log-in directory and other information that gives a user access to the system.

ACF. Advanced Communications Function.

ACF/NCP. Advanced Communications Function for the Network Control Program.

activate logical unit request (ACTLU). A request, sent by the host to the LANDP SNA server, to establish a logical session. The LANDP SNA server sends a positive response if the logical unit has been defined for this workstation.

activate physical unit request (ACTPU). A request, sent by the host to the LANDP SNA server, to establish a physical session.

active. In an XLR environment, the server (and, by implication, the workstation) that handles client requests and sends logging data to the backup.

ACTLU. Activate logical unit request.

ACTPU. Activate physical unit request.

adapter. (1) A part that electrically or physically connects a device to a computer or to another device. (2) A printed circuit board that modifies the system unit to allow it to operate in a particular way.

address. The unique code assigned to each device or workstation connected to a network. A standard Internet address is a 32-bit address field. This field can be broken into two parts. The first part contains the network address; the second part contains the host number.

Advanced Communications Function (ACF). (1) A group of IBM licensed programs, principally VTAM programs, TCAM, NCP, and SSP, that use the concepts of Systems Network Architecture (SNA), including distribution of function and resource sharing. (2) See also Network Control Program (NCP).

Advanced Communications Function for the Network Control Program (ACF/NCP). (1) An IBM program product that provides communication controller support for single-domain, multiple-domain, and interconnected network capability. (2) See also Advanced Communications Function (ACF) and Network Control Program (NCP).

advanced program-to-program communication (APPC). The general facility characterizing the LU 6.2 architecture and its various implementations in products.

AID. Attention identifier.

AIX (Advanced Interactive Executive). IBM's licensed version of the UNIX operating system.

alert. (1) A message sent to a management services focal point in a network to identify a problem or an impending problem. (2) In the NetView program, a high-priority event that warrants immediate attention. A database record is generated for certain event types that are defined by user-constructed filters.

alert condition. A problem or impending problem for which information is collected and possibly forwarded for problem determination, diagnosis, or resolution.

alert description. Information in an alert table that defines the contents of a Systems Network Architecture (SNA) alert for a particular message ID.

alert focal point. The system in a network that receives and processes (logs, displays, and optionally forwards) alerts. An alert focal point is a subset of a problem management focal point.

alert ID number. A value created from specific fields in the alert using a cyclic redundancy check. A focal point uses this value to refer to a particular alert, for example, to filter out duplicate alerts.

alert type. A value in an alert that indicates the problem being reported.

American National Standards Institute (ANSI). An organization consisting of producers, consumers, and general interest groups, that establishes the procedures by which accredited organizations create and maintain voluntary industry standards in the United States. (A)

ANSI. American National Standards Institute.

APAR. Authorized program analysis report.

API. Application program interface.

APPC. Advanced program-to-program communication.

applet. A Java program designed to run within a Web browser. Contrast with application.

application. (1) In LANDP, a program using IBM LANDP for DOS, IBM LANDP for OS/2, IBM LANDP for Windows NT, IBM LANDP for AIX, IBM FBSS/2, IBM PC/Integrator, or IBM PC Integrator/2, tailored to the needs of the workstation user. (2) The use to which an information processing system is put; for example, a payroll application, an airline reservation application, a network application. (3) A collection of software components used to perform specific types of user-oriented work on a computer. (4) In Java programming, a self-contained, stand-alone Java program that includes a static main method. It does not require an applet viewer. Contrast with applet.

application program. (1) A program that is specific to the solution of an application problem. Synonymous with application software. (T) (2) A program written for or by a user that applies to the user's work, such as a program that does inventory control or payroll. (3) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

application program interface (API). (1) In LANDP, the common interface by which server functions are requested. Requests are expressed by issuing a call to the supervisor. (2) A functional interface supplied by the operating system or by a separately orderable licensed program that allows an application program written in a high-level language to use specific data or functions of the operating system or the licensed program. (3) The interface through which an application program interacts with an access method.

application software. (1) Software that is specific to the solution of an application problem. (T) Synonymous with application program. (2) Software coded by or for an end user that performs a service or relates to the user's work. (3) Software products such as games, spreadsheets, and word processing programs designed for use on a personal computer.

argument. (1) An independent variable. (l) (A) (2) Any value of an independent variable; for example, a search key; a number identifying the location of an item in a table. (l) (A) (3) A parameter passed between a calling program and a called program.

arrival sequence. An order in which records are retrieved that is based on the order in which records are stored in a physical file.

AS/400®. IBM Application System/400®.

ASCII (American National Standard Code for Information Interchange). The standard code, using a coded character set consisting of 7-bit coded characters (8-bits including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A)

Note: IBM has defined an extension to ASCII code (characters 128-255).

ASCIIZ format. A string of ASCII characters ending with a null character (X'00').

ASYNC. Asynchronous.

asynchronous (ASYNC). (1) Pertaining to two or more processes that do not depend upon the occurrence of specific events such as common timing signals. (T) (2) Without regular time relationship; unexpected or unpredictable with respect to the execution of program instructions.

attention identifier (AID). (1) A code in the inbound 3270 data stream that identifies the source or type of data that follow. (2) A character in a data stream indicating that the user has pressed a key, such as the Enter key, that requests an action by the system.

authorization. (1) In computer security, the right granted to a user to communicate with or make use of a computer system. (T) (2) An access right. (3) The process of granting a user either complete or restricted access to an object, resource, or function.

authorized program analysis report (APAR). A report of a problem caused by a suspected defect in a current unaltered release of a program.

B

back-out. To restore a file to a previous condition by removing changes in the inverse chronological order from which the changes were originally made.

backup. In an XLR environment, the server (and, by implication, the workstation) that accepts logging data from the active and maintains a mirror set of databases (at a transaction level).

BASIC. (1) Beginner's all-purpose symbolic instruction code. A procedural algebraic language originally designed for ease of learning with a small instruction repertoire. (A) (2) A high-level programming language with a small number of statements and a simple syntax that is designed to be easily learned and used and that is widely used for interactive applications on microcomputers.

Basic Input/Output System (BIOS). (1) Code that controls basic hardware operations, such as interactions with diskette drives, hard disk drives, and the keyboard. (2) See also NetBIOS.

BAT, bat. (1) A DOS batch file extension (.BAT). (2) A batch file that contains a series of commands to be processed sequentially.

BB. Begin bracket.

begin bracket (BB). (1) An SNA bracket protocol term issued by the LANDP SNA server when bracket protocol is requested in the bind session. (2) Contrast with end bracket.

BID. In SNA, a request to start a bracket.

bind. To associate a variable with an absolute address, identifier, or virtual address, or with a symbolic address or label in a program.

BIND. (1) In SNA, a request to start a session between two logical units. (2) Contrast with UNBIND.

binding. (1) In programming, an association between a variable and a value for that variable that holds within a defined scope. The scope may be that of a rule, a function call, or a procedure invocation. (T) (2) In the AIX operating system, a temporary association between

a client and both an object and a server that exports an interface to the object. A binding is meaningful only to the program that sets it and is represented by a bound handle.

BIOS. Basic Input/Output System.

block. (1) The smallest complete unit of data that can be transmitted between units in a communication network. The maximum size of a block depends on the characteristics of the sending or receiving unit. (2) A group of contiguous characters recorded as a unit. (3) See also connectivity programming request block, program control block.

browser. An Internet-based tool that lets users browse web sites.

buffer. (1) A routine or storage used to compensate for a difference in rate of flow of data, or time of occurrence of events, when transferring data from one device to another. (A) (2) A portion of storage used to hold input or output data temporarily.

C

C language. A language used to develop software applications in compact, efficient code that can be run on different types of computers with minimal change.

call. In LANDP, the invocation of one of the LANDP API routines, RMTREQ, GETRPLY and RMTAREQ (client calls) and GETREQ, RMTRPLY, and SRVINIT (server calls). A LANDP client uses the RMTREQ call to request a LANDP function. Calls use the connectivity programming request block (CPRB) to pass and receive information. The syntax of a call varies with the programming language. The following examples are for COBOL and C respectively

```
CALL "RMTREQ" USING BY REFERENCE EHC-CPRB  
                   BY VALUE      EHC-RESERVED
```

```
retcode = GETREQ(&mycprb, EHC_RESERVED);
```

CCITT. Comité Consultatif International Télégraphique et Téléphonique. The International Telegraph and Telephone Consultative Committee.

CD. Compact disc.

CD-ROM. Compact disc-read-only memory.

CICS®. Customer Information Control System.

CID. Configuration, Installation, and Distribution. An IBM standard methodology for installing and distributing products under DOS, OS/2, and Windows 3.1.

ciphertext. (1) In computer security, text produced by encryption. (2) Synonym for enciphered data.

cleartext. (1) Nonencrypted data. (2) Synonym for plaintext.

class. An encapsulated collection of data and methods to operate on data. A class can be instantiated to produce an object that is an instance of the class.

CLASSPATH. In your deployment environment, the environment variable keyword that specifies the directories in which to look for class and record files.

client. (1) A functional unit that receives shared services from a server. (T) (2) A user. (3) See also client/server, client workstation, server, and user.

client workstation. (1) In IBM LANDP for DOS, IBM LANDP for OS/2, IBM LANDP for AIX, IBM LANDP for Windows NT, IBM FBSS/2, IBM PC/Integrator, and IBM PC Integrator/2, a workstation in a LAN that uses a service. (2) See also client, client/server, server, and user.

client/server. (1) In communications, the model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called a client; the answering program is called a server. (2) See also client, client workstation, server, and user.

CLIST, clist. Command list.

close. (1) A LANDP family function used to release a server. (2) To end the processing of a file. (3) A data manipulation function that ends the connection between a file and a program. (4) Contrast with open.

COBOL. Common business-oriented language. A high-level programming language, based on English, that is used primarily for business applications.

code page. An assignment of graphic characters and control function meanings to all code points; for example, assignment of characters and meanings to 256 code points for an 8-bit code, assignment of characters and meanings to 128 code points for a 7-bit code.

collating sequence. A specified arrangement used in sequencing. (I) (A)

COM, com. A DOS file with the file extension .COM.

command. (1) Loosely, a mathematical or logic operator. (A) (2) A request from a terminal for performance of an operation or processing of a program. (3) A character string from a source external to a system that represents a request for system action.

command list (CLIST, clist). A list of commands and statements designed to perform a specific function for the user.

Common User Access™ architecture. Guidelines for the dialog between a human and a workstation or terminal. One of the three SAA architectural areas.

communication configuration. In LANDP, the process of selecting and describing to the LANDP programs the particular arrangement of communication functions about a particular user.

communication controller. (1) A device that directs the transmission of data over the data links of a network; its operation may be controlled by a program executed in a processor to which the controller is connected or it may be controlled by a program executed within the device. (T) (2) A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit. It manages the details of line control and the routing of data through a network.

communication server. A server that communicates with a remote computer for various workstations in a local area network.

Communications Server. An IBM licensed program that supports the development and use of OS/2 applications involving two or more connected systems or workstations. IBM SecureWay Communications Server for OS/2 Warp provides multiple concurrent connectivities using different protocols for IBM 3270 and 5250 emulation sessions, printer sessions, and file transfers. It supports a range of application programming interfaces (API), which may be called concurrently and are designed for a variety of applications. IBM SecureWay Communications Server for OS/2 Warp includes the necessary interfaces for network management.

compact disc (CD). (1) A disc, usually 4.75 inches in diameter, from which data is read optically by means of a laser. (2) A disc with information stored in the form of pits along a spiral track. The information is decoded by

a compact-disc player and interpreted as digital audio data, which most computers can process.

compact disc-read-only memory (CD-ROM). A 4.75-inch optical memory storage medium, capable of storing about 550 megabytes of data. The standards for CD-ROM storage are known as the "Yellow Book."

compaction. (1) Any method for encoding data to reduce the storage it requires. (2) In SNA, the transformation of data by packing two characters in a byte so as to take advantage of the fact that only a subset of the allowable 256 characters is used; the most frequently sent characters are compacted. (3) See also compression and encode.

compression. (1) The process of eliminating gaps, empty fields, redundancies, and unnecessary data to shorten the length of records or blocks. (2) In SNA, the replacement of a string of up to 64 repeated characters by an encoded control byte to reduce the length of the data stream sent to the LU-LU session partner. The encoded control byte is followed by the character that was repeated (unless that character is the prime compression character). (3) Contrast with decompression.

config.sys. A file created during the customization process that holds the details about the system configuration. The CONFIG.SYS file is used during system operation.

configuration. (1) The manner in which the hardware and software of an information processing system are organized and interconnected. (T) (2) The physical and logical arrangement of devices and programs that make up a data processing system. (3) The devices and programs that make up a system, subsystem, or network.

connection. (1) An association established between functional units for conveying information. (2) The path between two protocol modules that provide reliable stream delivery service. On the Internet, a connection extends from a TCP module on one machine to a TCP module on the other.

connectivity. The capability to attach a variety of functional units without modifying them.

connectivity programming request block (CPRB). The control block used for communication between a server and a client. This control block contains the information that is exchanged between clients and

servers, and the information required for routing the requests and replies.

constructor. A method called to set up a new instance of a class.

control program. A computer program designed to schedule and supervise the execution of programs of a computer system. (I) (A)

coprocessor. (1) A supplementary processor that performs operations in conjunction with another processor. (2) In personal computers, a microprocessor on an expansion board that extends the address range of the processor in the system unit or adds specialized instructions to handle a particular category of operations; for example, an I/O coprocessor, math coprocessor, or networking coprocessor.

corrective service diskette. A diskette provided by IBM to registered service coordinators for resolving user-identified problems with previously installed software. This diskette includes program updates designed to resolve problems.

CPRB. Connectivity programming request block.

CRC. The cyclic redundancy check character. (A)

critical error handler. A routine that the operating system calls automatically if an error occurs in an operating system function call. There is a standard error handler or the user can provide one for special functions.

CRV. Cryptography verification request.

cryptography. (1) The transformation of data to conceal its meaning. (2) In computer security, the principles, means, and methods for encrypting plaintext and decrypting ciphertext.

cryptography key. A parameter that determines cryptographic transformations between plaintext and ciphertext.

cryptography verification (CRV) request. A request unit sent by the primary logical unit (PLU) to the secondary logical unit (SLU) as part of cryptographic session establishment, to allow the SLU to verify that the PLU is using the correct session cryptography key and initialization vector (IV).

CTS. Clear to Send.

CUA™ architecture. Common User Access™ architecture.

cursor. (1) A movable, visible mark used to show a position of interest on a display surface. (A) (2) In SAA Common User Access architecture, a visual cue that shows a user where keyboard input will appear on the screen.

Customer Information Control System (CICS®). An IBM licensed program that allows transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

Customer Information Control System for Virtual Storage (CICS/VS). An IBM licensed program used in a communications network.

customization. The process of designing a data processing installation or network to meet the requirements of particular users.

customization workstation. A workstation on which LANDP is installed, and which is used to customize a LANDP workgroup.

cyclic redundancy check character (CRC). A character used in a modified cyclic code for error detection and correction. (A)

D

DASD. Direct access storage device.

data circuit-terminating equipment (DCE). In a data station, the equipment that provides the signal conversion and coding between the data terminal equipment (DTE) and the line. (I)

Notes:

1. The DCE may be separate equipment or a part of the DTE or an integral part of the DTE or of the intermediate equipment.
2. A DCE may perform other functions that are usually performed at the network end of the line.

Data Encryption Standard (DES). In computer security, the National Institute of Standards and Technology (NIST) Data Encryption Standard, adopted by the U.S. government as Federal Information Processing Standard (FIPS) Publication 46, which allows only hardware implementations of the data encryption algorithm.

data flow control (DFC). In SNA, a request/response unit (RU) category used for requests and responses exchanged between the data flow control layer in one half-session and the data flow control layer in the session partner. Half duplex, flip-flop is the only LANDP-supported data flow control for both send and receive.

data link control (DLC). (1) In SNA, the layer that consists of the link stations that schedule data transfer over a link between two nodes and perform error control for the link. Examples of data link control are SDLC for serial-by-bit link connection and data link control for the System/370™ channel. (2) See also Systems Network Architecture (SNA). (3) In SNA, a set of rules used by two nodes on a data link to accomplish an orderly exchange of information.

data set. The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access. Sometimes called a file.

data terminal equipment (DTE). The part of a data station that serves as a data source, data sink, or both. (I) (A)

database description (DBD). (1) In LANDP, the shared-file server descriptor. (2) In IMS/VS, the collection of macro-parameter statements that describes an IMS/VS database. These statements describe the hierarchical structure, IMS/VS organization, device type, segment length, sequence fields, and alternate search fields. The statements are assembled to produce database description blocks.

datagram. The basic unit of information that is passed across the Internet. It consists of one or more data packets.

DBCS. Double-byte character set.

DBD. Database description.

DBM. Database manager.

DCA. Direct communication adapter.

DCE. (1) Data circuit-terminating equipment. (2) Distributed Computing Environment.

DDE. Dynamic data exchange.

DDT. Diagnostic and debugging tool.

decipher. (1) To convert enciphered data in order to restore the original data. (T) (2) In computer security, to convert ciphertext into plaintext by means of a cipher system. (3) To convert enciphered data into clear data. (4) Synonymous with decrypt. (5) Contrast with encipher.

decompression. (1) A function that expands data to the length that preceded data compression. (2) Contrast with compression.

decrypt. (1) In computer security, to decipher or decode. (2) Synonymous with decipher. (T)

default. A value, attribute or option that is assumed when none is explicitly specified.

delimiter. (1) A character used to show the beginning and end of a character string. (T) (2) A character that groups or separates words or values in a line of

deprecation. An obsolete component that may be deleted from a future release of a product.

DES. Data Encryption Standard.

development workstation. A workstation which is part of a LANDP workgroup, and which is customized via a customization workstation.

device driver. In Advanced DOS, a file that contains the code needed to attach and use a device.

DFC. Data flow control.

DIN. Deutsches Institut für Normung.

direct access. (1) The capability to obtain data from a storage device, or to enter data into a storage device, in a sequence independent from their relative position, by means of addresses indicating the physical position of the data. (T) (2) Contrast with sequential access.

direct access storage device (DASD). A device where access time is effectively independent of the location of the data.

directory. (1) A table of identifiers and references to the corresponding items of data. (I) (A) (2) A type of file containing the names and controlling information for other files or other directories. (3) An index that is used by a control program to locate one or more blocks of data that are stored in separate areas of a data set in direct access storage. (4) A listing of the files stored on a diskette.

directory service (DS). An application service element that translates the symbolic names used by application processes into the complete network addresses used in an OSI environment. (T)

disk. (1) A round, flat data medium that is rotated to read or write data. (T) (2) Loosely, a magnetic disk unit.

disk operating system. An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data.

diskette. (1) A thin, flexible magnetic disk and a semi-rigid protective jacket, where the disk is permanently enclosed. (2) Contrast with hard disk.

Distributed Computing Environment (DCE). The Open Software Foundation (OSF) specification (or a product derived from this specification) that assists in networking. DCE provides such functions as authentication, directory service (DS), and remote procedure call (RPC).

distributed system. A data processing system where processing, storage, and control functions, and also input and output operations, are distributed among remote locations.

distribution diskette. A diskette on which IBM sends programs and documentation to a customer.

DLC. Data link control.

DLL. Dynamic link library.

DMA. Direct memory access.

domain. (1) The part of a computer network where the data processing resources are under common control. (T) (2) In a database, all the possible values of an attribute or a data element. (3) In SNA, a system services control point (SSCP) and the physical units (PUs), logical units (LUs), links, link stations, and all associated resources that the SSCP could control with activation requests and deactivation requests.

DOS. Disk Operating System.

double-byte character set (DBCS). (1) A set of characters in which each character is represented by 2 bytes. Languages such as Japanese, Chinese, and Korean, which contain more symbols than can be represented by 256 code points, require double-byte character sets. Because each character requires 2

bytes, the typing, display, and printing of DBCS characters requires hardware and programs that support DBCS. (2) Contrast with single-byte character set (SBCS).

DS. Directory service.

DSR. Data Set Ready.

DTE. Data terminal equipment. (A)

DTE/DCE interface. The physical interface and link access procedures between a data terminal equipment (DTE) and a data circuit-terminating equipment (DCE).

dynamic data exchange (DDE). The exchange of data between programs or between a program and a data-file object. Any change made to information in one program or session is applied to the identical data created by the other program.

dynamic link library (DLL). A file containing executable code and data bound to a program at load time or run time, rather than during linking. The code and data in a dynamic link library can be shared by several applications simultaneously.

E

EB. End bracket.

EBCDIC. Extended binary-coded decimal interchange code.

EGA. Enhanced graphics adapter.

EID. End-of-message (EOM) identification.

EMM. Expanded memory manager.

emulation. The use of a data processing system to imitate another data processing system, so that the imitating system accepts the same data, executes the same programs, and achieves the same results as the imitated system. Emulation is usually achieved with hardware or firm-ware. (T)

encipher. (1) To scramble data or to convert data to a secret code that masks the meaning of the data to any unauthorized recipient. Synonymous with encrypt. (T) (2) In computer security, to convert plaintext into an unintelligible form by means of a cipher system. Synonymous with cipher. (3) Contrast with decipher. See also encode.

enciphered data. (1) Data whose meaning is concealed from unauthorized users or observers. (2) Synonymous with encode.

encode. (1) To convert data by the use of a code in such a manner that reconversion to the original form is possible. (T) (2) In computer security, to convert plaintext into an unintelligible form by means of a code system. (3) See also plaintext.

encrypt. (1) In computer security, to encode or encipher. (2) Synonym for encipher. (T)

end bracket (EB). (1) An SNA bracket protocol term used when the bind session specifies the end bracket call. If specified in the bind session, the personal computer may send both begin bracket and end bracket calls (not-response mode protocol). (2) Contrast with begin bracket.

end-of-message (EOM). The character or sequence of characters that shows the end of a message or record.

enhanced graphics adapter (EGA). An adapter, such as the IBM Enhanced Graphics Adapter, that provides high-resolution graphics, allowing the use of a color display for text processing and also graphics applications.

environment. A named collection of logical and physical resources used to support the performance of a function.

EOM. End-of-message.

erase. To remove data from a data medium. Erasing is usually accomplished by overwriting the data or deleting the references. (T)

error log. (1) A data set or file in a product or system where error information is stored for later access. (2) A record of machine checks, device errors, and volume statistical data.

error message. An indication that an error has been detected. (A)

ERRORLEVEL. A parameter of the IF command used by batch files. It is used in testing for failure of recently loaded programs.

event. (1) An occurrence or happening. (2) An occurrence of significance to a task; for example, the completion of an asynchronous operation, such as an input/output operation. (3) A data link control command

and response passed between adjacent nodes that allows the two nodes to exchange identification and other information necessary for operation over the data link. (4) In the NetView program, a record indicating irregularities of operation in physical elements of a network.

exception. An object that has caused some new condition, such as an error. In Java, throwing an error means passing that object to an interested party. A signal indicates what condition has occurred. Catching the condition means receiving the sent object. Handling this exception means dealing with the problem after receiving the object (though it might mean doing nothing, which is bad programming practice).

exchange identification (XID). The ID that is exchanged with the remote physical unit when an attachment is first established.

EXE, exe. An executable file with the file extension .EXE.

extended ASCII. A set of ASCII codes that uses the eighth (most significant) bit to define 127 additional codes. Standard ASCII uses 7 bits and defines 128 codes.

extended binary-coded decimal interchange code (EBCDIC). A coded character set of 256 8-bit characters.

external logging replicator (XLR). Shared-file mode of operation in which fault-tolerant data replication is achieved by logging database updates to an external server.

F

facility. (1) An operational capability, or the means for providing such a capability. (T) (2) A service provided by an operating system for a particular purpose; for example, the checkpoint/restart facility.

FBSI. Financial Branch Systems Integrator.

FBSS (DOS). IBM Financial Branch Systems Service (DOS). The predecessor to LANDP.

FBSS/2. Financial Branch Systems Service/2.

FCB. File control block.

FIC. First-in-chain.

file. (1) A named set of records stored or processed as a unit. (T) (2) A collection of information treated as a unit. (3) A collection of data that is stored and retrieved by an assigned name.

file control block (FCB). A record that contains all of the information about a file, such as its structure, length, and name.

file index table (FIT). A table used by WorkSpace On-Demand to redirect file access requests from a client workstation's boot drive to the appropriate location on the boot server.

file server. A high-capacity disk storage device or a computer that each computer on a network can use to access and retrieve files that can be shared among the attached computers.

file transfer. In remote communications, the transfer of one or more files from one system to another over a communications link.

first-in-chain (FIC). A request unit (RU) whose request header (RH) begin chain indicator is on and whose RH end chain indicator is off.

FIT. file index table

fixed disk. Synonym for hard disk.

flag. (1) A variable indicating that a certain condition holds. (T) (2) Any of various types of indicators used for identification; for example, a word mark. (A) (3) A character that signals the occurrence of some condition, such as the end of a word. (A)

FMH. Function management header.

format identification (FID) field. In SNA, a field in each transmission header (TH) that shows the format of the transmission header; that is, the presence or absence of certain fields.

forward recovery. The process of reconstructing a file from a particular point by restoring a saved version of the file and then applying changes to that file in the same order in which they were originally made.

function. (1) In IBM LANDP for DOS, IBM LANDP for OS/2, IBM LANDP for Windows NT, IBM FBSS (DOS), IBM FBSS/2, IBM PC/Integrator, and IBM PC Integrator/2 a function is the specification of an activity to be performed by a server. (2) In computer programming, synonym for procedure.

function management header (FMH). (1) A special record or part of a record that contains control information for the data that follow. (2) In SNA, one or more headers optionally present in the leading request units (RUs) of an RU chain that allow a half-session in an LU-LU session to: (a) select a destination as session partner and control way where end-user data it sends are handled at the destination, (b) change destination or characteristics of data during session, and (c) send between session partners status or user information about destination; for example, whether it is a program or device.

G

gateway. (1) In LANDP, the workstation that connects the LANDP workgroup to a host computer with the necessary LANDP software and the respective physical attachment. (2) A functional unit that interconnects two computer networks with different network architectures. A gateway connects networks or systems of different architectures. A bridge interconnects networks or systems with the same or similar architectures. (T) (3) A network that connects hosts. (4) Contrast with router.

generic alert. A product-independent method of encoding alert data by means of both (a) code points indexing short units of stored text and (b) textual data.

H

hard disk. (1) A rigid magnetic disk such as the internal disks used in the system units of IBM personal computers and in external hard disk drives. (2) Synonym for fixed disk. (3) Contrast with diskette.

HDLC. High-level data link control.

hexadecimal. Describing a numbering system with base of sixteen; valid numbers use the digits 0 through 9 and characters A through F, where A represents 10 and F represents 15.

high-level data link control (HDLC). In data communication, the use of a specified series of bits to control data links under the International Standards for HDLC: ISO 3309 Frame Structure and ISO 4335 Elements of Procedures.

host, host computer, host processor, or host system. (1) The primary or controlling computer in a multiple computer installation. (2) A computer used to prepare programs for use on another computer or on

another data processing system; for example, a computer used to compile, link edit, and test programs to be used on another system.

hot-key. The key combination used to change from one session to another on the workstation.

Hypertext Transfer Protocol (HTTP). The Internet protocol, based on TCP/IP, that is used to fetch hypertext objects from remote hosts.

I

I/O. Input/output.

IBM Operating System/2® (OS/2). Pertaining to the IBM licensed program that can be used as the operating system for personal computers. The OS/2 licensed program can perform multiple tasks at the same time.

ICV. Initial chaining value.

ID. (1) Identifier. (2) Identification.

identification. In computer security, the process that allows a system to recognize an entity with personal, equipment, or organizational characteristics or codes.

identifier. One or more characters used to identify or name a data element or possibly to show certain properties of that data element. (A)

IEEE. Institute of Electrical and Electronics Engineers.

IMS/VS. Information Management System/Virtual Storage.

indexed access. Pertaining to the organization and accessing of the records of a storage structure through a separate index to the locations of the stored records. (A)

indexed sequential access. Pertaining to the organization and accessing of records through an index of the keys that are stored in arbitrarily partitioned sequential files. (A)

initial chaining value (ICV). An 8-byte pseudo-random number used to verify that both ends of a session with cryptography have the same session cryptography key. The initial chaining value is also used as input to Data Encryption Standard (DES) algorithm to encipher or decipher data in a session with cryptography.

initial program load (IPL). (1) The initialization procedure that causes an operating system to begin

operation. (2) The process by which a configuration image is loaded into storage at the beginning of a work day or after a system malfunction. (3) The process of loading system programs and preparing a system to run jobs.

initialization. (1) The operations required for setting a device to a starting state, before the use of a data medium, or before implementation of a process. (T) (2) Preparation of a system, device, or program for operation.

initiate self. An SNA command issued by the LANDP SNA server to initiate a host application. The SNA command is issued in response to the receipt of an Open command from the personal computer.

INITSELF. Initiate self.

input/output (I/O). (1) Describing a device whose parts can perform an input process and an output process at the same time. (I) (2) Describing a functional unit or channel involved in an input process, output process, or both, concurrently or not, and to the data involved in such a process.

Instruction Pointer (IP). In System 38, a pointer that provides addressability for a machine interface instruction in a program.

interface. A shared boundary between two functional units, defined by functional characteristics, signal characteristics, or other characteristics, as appropriate. The concept includes the specification of the connection of two devices having different functions. (T)

International Organization for Standardization (ISO). An organization of national standards bodies from various countries established to promote development of standards to simplify international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

Internet Protocol (IP). A protocol used to route data from its source to its destination in an Internet environment.

interoperability. (1) The capability to communicate, execute programs, or transfer data among various functional units in a way that requires the user to have little or no knowledge of the unique characteristics of those units. (T) (2) In SAA usage, the ability to link SAA and non-SAA environments and use the combination for distributed processing.

IP. (1) Instruction Pointer. (2) Internet Protocol.

IPL. Initial program load.

ISAM. Indexed sequential access method.

ISO. International Organization for Standardization.

J

Jar file format. Java Archive, a platform-independent file format that aggregates many files into one. Multiple Java applets and their requisite components (.class files, images, sounds, and other resource files) can be bundled in a JAR file and subsequently downloaded to a browser in a single HTTP transaction.

Java. An object-oriented programming language for portable, interpretive code that supports interaction among remote objects. Java was specified and developed by Sun Microsystems, Incorporated. The Java environment consists of the JavaOS, the Virtual Machines for various platforms, the object-oriented Java programming language, and several class libraries.

Java Development Kit (JDK). A set of Java technologies made available to licensed developers by Sun Microsystems. Each release of JDK consists of the Java compiler, Java virtual machine, Java class libraries, Java applet viewer, Java debugger, and other tools.

JavaDoc. Sun Microsystems tool for generating HTML documentation of classes by extracting comments from the Java source code files.

Java Remote Method Invocation (RMI). Method invocation between peers, or between client and server, when applications at both ends of the invocation are written in Java. Java RMI is included in JDK 1.1.

Java Virtual Machine. A software implementation of a central processing unit (CPU) that runs compiled Java code (applets and applications).

journal. (1) A chronological record of changes made in a set of data; the record may be used to reconstruct a previous version of the set. (T) (2) A special-purpose data set that provides an audit trail of operator and system actions, or as a means of recovering superseded data.

JVM. Java Virtual Machine.

K

KB. Kilobyte; 1024 bytes.

key. (1) An identifier within a set of data elements. (T) (2) One or more characters used to identify the record and establish the order of the record within an indexed file.

keystroke. Actuation of a key on a keyboard to perform or release a machine function. (T)

keyword. A name or symbol that identifies a parameter or an ordered set of parameters.

L

LAN. Local area network.

LAN configuration. The process by which the details about the structure of the LAN for a particular user are provided to the LAN family programs. This includes details about the workstations forming the LAN, the services provided by each workstation, and the workstations that receive the services.

LAN trace. A LAN family trace facility that informs about the LAN-related LAN and displays the status of the local area network.

LAN Distributed Platform. The former name for the LAN family of products.

last-in-chain (LIC). A request unit (RU) whose request header (RH) end chain indicator is on and whose RH begin chain indicator is off.

LDA. Logical device address.

LED. Light-emitting diode.

LIC. Last-in-chain.

light-emitting diode (LED). A semiconductor chip that gives off visible or infrared light when operated.

link connection. In SNA, the physical equipment providing two-way communication between one link station and one or more other link stations; for example, a telecommunication line and data circuit-terminating equipment (DCE).

LIP. LAN Internet Protocol.

LLAP. Logical link access path.

loader. A routine, commonly a computer program, that reads data into main storage. (A)

local area network (LAN). A computer network located on a user's premises within a limited geographical area. Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary may be subject to some form of regulation. (T)

local host. In the Internet, the computer to which a user's terminal is directly connected without using the Internet.

logging. The recording of data about specific events.

logical device address (LDA). (1) A number used to represent a terminal or terminal component within a workstation. (2) See also physical device address.

logical link access path (LLAP). In a multi-system environment, the path between any two systems. One or more logical link paths must be defined for each logical link.

logical unit (LU). (1) In SNA, a port through which an end user accesses the SNA network to communicate with another end user and through which the end user accesses the functions provided by the system services control points (SSCPs). An LU can support at least two sessions, one with an SSCP and one with another LU, and may be capable of supporting many sessions with other logical units. (2) A type of network addressable unit that allows end users to communicate with each other and gain access to network resources.

longitudinal parity check. A parity check of a row of binary digits that are members of a set forming a matrix; for example, a parity check of the bits of a track in a block on a magnetic stripe. (T)

longitudinal redundancy check (LRC). Synonym for longitudinal parity check.

LRC. Longitudinal redundancy check.

LU. Logical unit.

LU—LU session type 0. In SNA, a type of session between two LU—LU half-sessions using SNA-defined protocols for transmission control and data flow control, but using end-user or product-defined protocols to augment or replace FMD services protocols.

LU—LU session type 1. In SNA, a type of session between an application program and single- or multiple-device data processing terminals in an interactive, batch data transfer, or distributed processing environment.

LU—LU session type 2. In SNA, a type of session between an application program and a single display terminal in an interactive environment, using the SNA 3270 data stream.

LUSTAT. An SNA command used to send logical unit status information.

M

MAC. Message authentication code.

mapper. A device, such as a piece of code, which performs a mapping function.

mapping. (1) A list, usually in a profile, that establishes a correspondence between items in two groups; for example, a keyboard mapping can establish what character is displayed when a certain key is pressed. (2) In a database, the establishing of correspondences between a given logical structure and a given physical structure. (T)

MB. Megabyte; 1 048 576 bytes.

memory. All of the addressable storage space in a processing unit and other internal storages that is used to execute instructions. (T)

message. (1) An assembly of characters and sometimes control codes that is transferred as an entity from an originator to one or more recipients. A message consists of two parts: envelope and content. (T) (2) A communication sent from a person or program to another person or program. (3) A unit of data sent over a telecommunication line. (4) One or more message segments transmitted among terminals, application programs, and systems. (5) In SAA Common User Access architecture, information not requested by a user but displayed by an application in response to an unexpected event, or when something undesirable could occur.

message authentication code (MAC). (1) In computer security, a value, part of, or accompanying a message, used to determine that the contents, origin, author, or other attributes of all or part of the message are as they appear to be. (2) In cryptography: (a) a number or

value derived by processing data with an authentication algorithm, (b) the cryptographic result of block cipher operations on text or data using a cipher block chain (CBC) mode of operation, (c) a digital signature code.

method. A fragment of Java code within a class that can be invoked and passed a set of parameters to perform a specific task.

MIC. Middle-in-chain.

MICR. Magnetic ink character recognition.

microcode. (1) One or more microinstructions. (2) A code, representing the instructions of an instruction set, that is done in a part of storage that is not program-addressable. (3) To design, write, and also to test one or more microinstructions.

middle-in-chain (MIC). A request unit (RU) whose request header (RH) begin chain indicator and RH end chain indicator are both off.

mnemonic. A symbol chosen to help the user remember the significance of the symbol.

mode. A method of operation.

mode switching. Operator switching between a concurrently running personal computer application and 3270 emulation or other internal application.

MSR, MSR/E. Magnetic stripe reader; Magnetic stripe reader/encoder.

multi-tasking. A mode of operation that provides for concurrent performance, or interleaved execution of two or more tasks. (I) (A)

MVDM. Multiple Virtual DOS Machine.

N

name server. (1) The server that stores resource records about hosts. (2) In the AIX operating system, a host that provides name resolution for a network. Name servers translate symbolic names assigned to networks and hosts into the Internet addresses used by machines. (3) In TCP/IP, synonym for domain name server.

NAU. Network addressable unit.

NCP. Network Control Program.

NDIS. Network Driver Interface Specification

NetBIOS. (1) Network Basic Input/Output System. A standard interface to networks, IBM personal computers (PCs), and compatible PCs, that is used on LANs to provide message, print-server, and file-server functions. Application programs that use NetBIOS do not need to handle the details of LAN data link control (DLC) protocols. (2) See also BIOS.

NetView program. An IBM licensed program used to monitor and manage a network and to diagnose network problems.

network. (1) An arrangement of nodes and connecting branches. (T) (2) A configuration of data processing devices and software connected for information interchange.

network addressable unit (NAU). (1) In SNA, a logical unit, a physical unit, or a system services control point. The NAU is the origin or the destination of information transmitted by the path control network. (2) See also logical unit, physical unit, system services control point (SSCP).

Network Control Program (NCP). (1) An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability. (2) See also Advanced Communications Function (ACF).

network management vector transport (NMVT). A management services request/response unit (RU) that flows over an active session between physical unit management services and control point management services (SSCP-PU session).

network resource. In ACF/VTAM®, a network component such as a local network control program, an SDLC data link, or a peripheral node.

network services procedure error (NSPE). A request unit that is sent by a system services control point (SSCP) to a logical unit (LU) when a procedure requested by that LU has failed.

NLS. National language support.

NMVT. Network management vector transport.

node. (1) In a network, a point at which one or more functional units connect channels or data circuits. (I) (2) In network topology, the point at an end of a branch. (T)

NPSI. X.25 NCP Packet Switching Interface.

NSPE. Network services procedure error.

O

object. The principal building block of object-oriented programs. Objects are software programming modules. Each object is a programming unit consisting of related data and methods.

object-oriented programming (OOP). A programming approach based on the concepts of data abstraction and inheritance. Unlike procedural programming techniques, object-oriented programming concentrates on the data objects that constitute the problem and how they are manipulated, not on how something is accomplished.

ODBC. Open Database Connectivity is a standardized set of API function calls that can be used to access data stored in both relational and non-relational DBMSs.

OIA. Operator information area.

OIC. Only-in-chain.

only-in-chain (OIC). A request unit (RU) for which the request header (RH) begin chain indicator and RH end chain indicator are both on.

OOP. object-oriented programming

open. (1) The function that connects a file to a program for processing. (2) Contrast with close.

open system. A system with specified standards, and that therefore can be readily connected to other systems that comply with the same standards.

operating system. Software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible. (T)

operator information area (OIA). In the 3270 Information Display System, the area near the bottom of the display area where terminal or system status information is displayed.

option. A specification in a statement that may be used to influence the processing of the statement.

OS/2 operating system. IBM Operating System/2.

P

padding. A technique by which a receiving station controls the rate of transmission of a sending station to prevent overrun.

package. A program element that contains classes and interfaces.

packet. A sequence of binary digits, including data and control signals, that is transmitted and switched as a composite entity.

panel. A formatted display of information that appears on a display screen.

parallel port. (1) On a personal computer system, a port used to attach devices such as dot matrix printers and input/output units; it transmits data one byte at a time. (2) See also serial port.

parameter. (1) A variable that is given a constant value for a specified application and that may denote the application. (I) (A) (2) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (3) Data passed between programs or procedures.

Pascal. A high-level, general purpose programming language, related to ALGOL. Programs written in Pascal are block structured, consisting of independent routines. They can run on different computers with little or no modification.

path. In a personal computer system, the logical relationship between directories.

PBM. Personal banking machine.

PC. Personal computer.

PC-ID. Workstation identifier.

PCB. Program control block.

PC/TCP. FTP Software's implementation of TCP/IP for systems running DOS and Windows. Now called PC/TCP Network Software version 5.0 and available from NetManage Inc..

PDA. Physical device address.

PDP. Problem determination procedure.

personal computer system. IBM Personal System/2 and also the various IBM Personal Computer system units, unless otherwise described.

Personal Identification Number (PIN) pad. A pad with twelve keys in a specific arrangement that display alphabetic and numeric characters that may be entered onto a financial transaction terminal. (T) (A)

physical device address (PDA). An address or set of addresses that identifies a particular device.

physical unit (PU). In SNA, the component that manages and monitors the resources, such as attached links and adjacent link stations, associated with a node, as requested by an SSCP via an SSCP-PU session. An SSCP starts a session with the physical unit to indirectly manage, through the PU, resources of the node such as attached links. This term applies to type 2.0, type 4, and type 5 nodes only.

PIN. Personal identification number.

plaintext. (1) Nonencrypted data. Synonymous with cleartext. (2) Synonym for clear data.

PLU. Primary logical unit.

PM. Presentation Manager® (in OS/2).

pointing device port. The IBM PS/2 port that allows attachment of various devices including pointing devices.

port. (1) An access point for data entry or exit. (2) A connector on a device to which cables for other devices such as display stations and printers are attached. (3) A specific communications end point within a host. A port is identified by a port number.

Post Telephone and Telegraph Administration (PTT). An organization, usually a government department, that provides communication common carrier services in countries other than the USA and Canada. Examples of PTTs are the Bundespost in Germany, and the Nippon Telephone and Telegraph Public Corporation in Japan.

PPC. Program to program communications.

Presentation Manager. A component of OS/2 that provides a complete graphics-based user interface, with pull-down windows, action bars, and layered menus.

primary logical unit (PLU). (1) In SNA, the logical unit (LU) that contains the primary half-session for a particular LU—LU session. (2) Contrast with secondary logical unit (SLU). (3) See also logical unit (LU).

problem determination procedure (PDP). A prescribed sequence of steps taken to identify the source of a problem.

process. (1) A unique, finite course of events defined by its purpose or by its effect, achieved under defined conditions. (2) Any operation or combination of operations on data. (3) A function being performed or waiting to be performed. (4) A program in operation.

processor. (1) In a computer, a functional unit that interprets and executes instructions. A processor consists of at least an instruction control unit and an arithmetic and logic unit. (T) (2) The functional unit that interprets and processes instructions.

profile. (1) In computer security, a description of the characteristics of an entity to which access is controlled. (2) Data that describes the significant characteristics of a user, a group of users, or one or more computer resources.

program. A sequence of instructions suitable for processing by a computer. Processing may include the use of an assembler, a compiler, an interpreter, or a translator to prepare the program for execution, and also to execute it. (I)

program control block (PCB). LANDP family shared-file server pointer related to a specific DBD.

Program temporary fix (PTF). A temporary solution or by-pass of a problem diagnosed by IBM as resulting from a defect in a current unaltered release of the program.

protocol. In SNA, the meanings of and the sequencing rules for requests and responses used for managing the network, transferring data, and synchronizing the states of network components.

PS/2. Personal System/2.

PTF. Program temporary fix.

PTT. Post Telephone and Telegraph Administration.

PU. Physical unit.

Q

QLLC. Qualified logical link control.

qualified logical link control (QLLC). An X.25 protocol that allows the transfer of data link control information between two adjoining systems network architecture (SNA) nodes that are connected through an X.25 packet-switching data network. The QLLC provides the qualifier "Q" bit in X.25 data packets to identify packets that carry logical link protocol information.

query. (1) A request for information from a file relying on specific conditions. (2) In the AS/400 system, the query management object that is used to define queries against relational data.

quiescing. The process of bringing a device or a system to a stop by rejection of new requests for work. (A)

R

RAM. Random access memory. (A)

random access memory (RAM). A storage device where data can be written and read.

RC. Return code.

RCMS. Remote change management services.

RDBMS. Relational database management system. A generic name for any relational database system such as DB2.

re-synchronization. Restarting the transmission of a function at the point where it was interrupted.

read-only memory (ROM). (1) A storage device where data, under normal conditions, can only be read. (T) (2) See also read-only storage (ROS).

read-only storage (ROS). (1) A storage device whose contents cannot be modified, except by a particular user, or when operating under particular conditions. (2) See also read-only memory (ROM).

record. (1) In programming languages, an aggregate that consists of data objects, possibly with different attributes, that usually have identifiers attached to them. In some programming languages, records are called structures. (I) (2) A set of data treated as a unit. (T)

(3) A set of one or more related data items grouped for processing.

remote attachment. A method of connecting two devices over a telecommunication line.

remote initial program load (remote IPL). A feature that permits a computer to receive its initial program from another computer, rather than from its own internal disk or diskette storage.

remote method invocation. A specific instance of the more general term RPC (remote procedure call). Remote method invocation (RMI) allows objects to be distributed over a network, that is, a Java program running on one computer can call the methods of an object running on another computer. RMI and java.net are the only 100% pure Java APIs for controlling Java objects in remote systems.

remote procedure call (RPC). A facility that a client uses to request the execution of a procedure call from a server. This facility includes a library of procedures and an external data representation.

REMS. Reader/encoder magnetic stripe.

request/response header (RH). In systems network architecture (SNA), control information preceding a request/response unit (RU) that specifies the type of RU and contains control information associated with the RU.

request/response unit (RU). In systems network architecture (SNA), a generic term for a request unit or a response unit.

resource. (1) Any of the data processing system elements needed to perform required operations, including storage, input/output units, one or more processing units, data, files, and programs. (T) (2) See also network resource.

retry. To resend data a prescribed number of times or until the data is received correctly.

return code (RC). (1) A code used to influence the execution of succeeding instructions. (A) (2) A value returned to a program to indicate the results of an operation requested by that program.

RH. Request/response header.

roll back. To remove changes that were made to database files under commitment control since the last commitment boundary.

RMI. Remote Method Invocation.

rollback. (1) A programmed return to a prior checkpoint. (A) (2) The process of restoring data changed by an application program or user to the state of its last commitment boundary. (3) In SQL, the process of restoring data changed by an application program or user to the state of its last commit point.

ROM. Read-only memory. (A)

ROS. Read-only storage.

router. (1) A computer that determines the path of network traffic flow. The path selection is made from several paths based on information obtained from specific protocols, algorithms that attempt to identify the shortest or best path, and other criteria such as metrics or protocol-specific destination addresses. (2) An attaching device that connects two LAN segments, which use similar or different architectures, at the reference model network layer. Contrast with bridge, gateway. (3) In OSI terminology, a function that determines a path by which an entity can be reached.

RPC. Remote procedure call.

RTR. Ready to Receive.

RU. Request/response unit.

S

SAM. Service availability manager.

SAP. Service access point.

SBCS. Single-byte character set.

scan code. A code generated by a keyboard.

SCS. Systems network architecture character string.

SDLC. Synchronous data link control.

secondary logical unit (SLU). (1) In systems network architecture (SNA), the logical unit (LU) that contains the secondary half-session for a particular LU-LU session. (2) Contrast with primary logical unit (PLU). (3) See also logical unit (LU).

SEQ. Sequential file.

sequential access. (1) The capability to enter data into a storage device or a data medium in the same

sequence as the data is ordered, or to obtain data in the same order as it has been entered. (T) (2) An access method in which records are read from, written to, or removed from a file based on the logical order of the records in the file. (3) Contrast with direct access.

serial port. (1) On personal computer systems, a port used to attach devices such as display devices, letter-quality printers, modems, plotters, and pointing devices such as light pens and mice; it transmits data one bit at a time. (2) See also parallel port.

serialization. Turning an object into a stream and back again.

server. (1) A functional unit that provides shared services to workstations over a network; for example, a file server, a print server, a mail server. (T) (2) In LANDP, a functional area that provides functions to LANDP workstations in a LANDP workgroup. (3) The computer that hosts the Web page that contains an applet. The .class files that make up the applet, and the HTML files that reference the applet reside on the server. When someone on the Internet connects to a web page that contains an applet, the server delivers the .class files over the Internet to the client that made the request. The server is also known as the originating host. (4) See also client, client workstation, and user. (5) In LANDP, a function provided by a server.

service access point (SAP). A logical point made available by a token-ring adapter where information can be received and transmitted.

service availability manager (SAM). Facility used by the shared-file server to provide fault-tolerant data access in an XLR environment.

servlet. Server-side program that executes on and adds function to a Web server. Java servlets allow for the creation of complicated, high-performance, cross-platform Web applications. They are highly extensible and flexible, making it easy to expand from from client or single-server applications to multi-tier applications.

session. (1) In systems network architecture (SNA), a logical connection between two network addressable units (NAU) that can be started, tailored to provide various protocols, and deactivated, as requested. (2) The time during which programs or devices can communicate with each other.

single-byte character set (SBCS). (1) A character set in which each character is represented by a one-byte

code. (2) Contrast with double-byte character set (DBCS).

SLU. Secondary logical unit.

SNA. Systems network architecture.

SNUF. Systems network architecture up-line facility.

socket. (1) An end-point for communication between processes or applications. (2) A pair consisting of TCP port and IP address.

SOM. Start-of-message code.

SPC, spc. Specification file.

specification file (SPC, spc). In LANDP, a file with the file extension .SPC. This file can be edited. It contains information for customization purposes.

SQL. Structured query language.

SSCP. System services control point.

start-of-message code (SOM). A character or group of characters transmitted by the polled terminal and indicating to other stations on the line that what follows are addresses of stations to receive the answering message.

storage. A functional unit into which data can be placed, where it can be retained, and from which it can be retrieved. (T)

stream. A continuous sequence of data elements being transmitted, or intended for transmission, in character or binary-digit form, using a defined format.

structured query language (SQL). An established set of statements used to manage information stored in a database. By using these statements, users can add, delete, or update information in a table, request information through a query, and display the results in a report.

subdirectory. A directory contained within another directory in a file system hierarchy.

synchronous. (1) About two or more processes that depend on the occurrence of a specific event such as common signal timing. (2) Occurring with a regular or predictable time relationship. (3) See also asynchronous.

synchronous data link control (SDLC). A discipline conforming to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute (ANSI) and High-level Data Link Control (HDLC) of the International Organization for Standardization, for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or not-switched links. The configuration of the link connection may be point-to-point, multi-point, or loop. (I)

system diskette. (1) The diskette, either real or virtual, that contains your control program. (2) In personal computer systems, the diskette on which you have the operating system.

system distribution manager. A system that contains the files and programs required for product installation, and initiates or manages the installation process.

system services control point (SSCP). In systems network architecture (SNA), the focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network.

systems network architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

systems network architecture character string (SCS). In systems network architecture (SNA), a character string composed of EBCDIC controls, optionally intermixed with end-user data, that is carried within a request/response unit (RU).

systems network architecture network (SNA network). In systems network architecture (SNA), the part of an application program network that conforms to the formats and protocols of SNA. It allows reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAU), boundary function components, and the path control network.

systems network architecture up-line facility (SNUF). The communications support that allows an AS/400 system to communicate with CICS/VS and IMS/VS application programs on a host computer.

T

takeover. In an XLR environment, the process by which a backup server assumes the role of the (failed) active. This involves backing out incomplete transactions, rebuilding indexes, and informing SAM of the new active workstation.

TCP/IP. Transmission Control Protocol/Internet Protocol.

terminal status line. Synonym for operator information area (OIA).

TH. Transmission header.

thin client. A client workstation that loads its operating system environment and applications across a network from a server. The degree of local processing power in a thin client can vary considerably depending on the implementation of the thin client concept.

The term thin client usually refers to a system that runs on a resource-constrained machine or that runs on a small operating system. These clients do not require local system administration, and they execute Java applications delivered over the network.

Time Sharing Option (TSO). An operating system option; for the System/370 system, the option provides interactive time sharing from remote terminals.

token-ring network. (1) A ring network that allows unidirectional data transmission between data stations by a token passing procedure, so that the transmitted data returns to the transmitting station. (T) (2) A network that uses a ring topology, where tokens are passed in a circuit from node to node. A node that is ready to send can capture the token and insert data for transmission.

trace. (1) A record of the execution of a computer program. It exhibits the sequences in which the instructions were executed. (A) (2) The process of recording the sequence in which the statements in a program are executed and, optionally, the values of the program variables used in the statements. (3) To record a series of events as they occur. (4) For data links, a record of the frames and bytes transmitted or received.

trace file. A file that contains a record of events that occur in a system.

trace function. A function used for problem determination.

trace log. A file in which trace events are recorded.

trace program. A computer program that performs a check on another computer program by exhibiting the sequence in which the instructions are executed and, usually, the results of executing the instructions. (I) (A)

trace routine. A routine that provides a historical record of specified events in the execution of a computer program. (A)

transaction. An exchange between a workstation and another device that accomplishes a particular action or result.

translation. Conversion of a code or codes to another code or codes according to a set of specifications.

transmission. The sending of data from one place for reception elsewhere. (A)

Notes:

1. Transmission implies only the sending of data; the data may or may not be received.
2. The term transmit is used to describe the sending of data in telecommunication operations. The terms move and transfer are used to describe movement of data in data processing operations.

transmission control (TC) layer. The layer within a half-session or session connector that synchronizes and paces session-level data traffic, checks session sequence numbers of requests, and enciphers and decipheres end-user data.

Transmission Control Protocol (TCP). A communications protocol used in the Internet and in any network that follows the US Department of Defense standards for inter-network protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It assumes that the Internet protocol is the underlying protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP). A set of communication protocols that support peer-to-peer connectivity functions for both local and wide area networks.

transmission header (TH). In systems network architecture (SNA), control information, optionally followed by a basic information unit (BIU) or a BIU segment, that is created and used by path control to

route message units and to control their flow within the network.

transmission services (TS) profile. In systems network architecture (SNA), a specification in a session activation request (and, optionally in the responses) of transmission control (TC) protocols, such as session-level pacing and the usage of session-level requests, to be supported by a particular session. Each defined TS profile is identified by a number.

trap. An unprogrammed conditional jump to a specified address that is automatically activated by hardware. A recording is made of the location from which the jump occurred.

TRDLC. Token-ring data link control.

TS. Transmission services.

TSO. Time Sharing Option.

U

UDP. User Datagram Protocol.

UNBIND. (1) In systems network architecture (SNA), a request to deactivate a session between two logical units (LU). (2) Contrast with BIND.

user. (1) A function that uses the services provided by a server. A host can be a user and a server at the same time. (2) Any person or any thing that may issue or receive commands and messages to or from the information processing system. (T) (3) Any person who requires the services of a computing system. (4) See also client, client/server, client workstation, and server.

User Datagram Protocol (UDP). In TCP/IP, a packet-level protocol built directly on the Internet protocol layer. UDP is used for application-to-application programs between TCP/IP host systems.

user profile. In computer security, a description of a user that includes such information as user identification (ID), user name, password, access authority, and other attributes obtained at log-on.

user-written server. In LANDP, a server not supplied with a LANDP program, but developed by the customer.

utility program. (1) A computer program which supports computer processes; for example, a sort program. (T) (2) A program designed to perform an everyday task such as copying data from one storage device to another. (A)

V

validation. The checking of data for correctness, or compliance with applicable standards, rules, and conventions. (A)

VDM. Virtual DOS machine.

vector. A set of keyword=parameter statements that define configuration items. These items can correspond to both model and real configurations.

verify. To determine whether a transcription of data or other operation has been accomplished accurately. (A)

VFS. Virtual file system.

virtual DOS machine (VDM). A functional simulation of a machine running under DOS.

virtual file system (VFS). A remote file system that has been mounted so that it is accessible to the local user.

virtual machine (VM). A virtual data processing system that seems to be at the exclusive disposal of a particular user, but whose functions are accomplished by sharing the resources of a real data processing system. (T)

Virtual Telecommunications Access Method

(VTAM). A set of programs that maintain control of the communication between terminals and application programs running under Disk Operating System/Virtual Storage (DOS/VS), OS/VS1, and OS/VS2 operating systems.

VisualGen®. A high-level object-oriented programming language.

VM/CMS. Virtual machine/conversational monitor system.

VTAM. Virtual Telecommunications Access Method.

W

WAN. Wide area network.

wide area network (WAN). A network that provides communication services to a geographical area larger than that served by a local area network.

WebSphere™. A comprehensive solution to build, deploy, and manage e-business Web sites. WebSphere is the cornerstone of IBM's overall Web strategy. The Websphere product line provides companies with an open, standards-based, Web server deployment platform, together with Web site development and management and management tools to help accelerate the process of moving to e-business.

window. A division of a screen where one of several programs being run concurrently can display information.

workgroup. In LANDP, the logical connection of LANDP for DOS, LANDP for OS/2, LANDP for Windows NT, and LANDP for AIX workstations through the LANDP client/server mechanism, which is available with each LANDP program.

Workspace On-Demand. (1) A set of management utilities that enables OS/2 Warp Server to remotely load a thin client operating system, known as Workspace On-Demand client, into a client workstation across a LAN. (2) The client workstation component of Workspace On-Demand, which is loaded into a client workstation from a server machine running OS/2 Warp Server and Workspace On-Demand Server.

Workspace On-Demand Server. A server, running OS/2 Warp Server and Workspace On-Demand, that is used to boot client workstations.

workstation. (1) A functional unit at which a user works. (2) In LANDP, personal computer system in a local area network (LAN).

wrapper. A language binding.

X

X.25. A CCITT recommendation that defines the physical level (physical layer), link level (data link layer), and packet level (network layer) of the open systems inter-connection (OSI) reference model. An X.25 network is an interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) operating in the packet mode, and connected to public data networks by dedicated circuits. X.25 networks use the connection-mode network service.

X.25 NCP Packet Switching Interface. An IBM-licensed program that allows systems network architecture (SNA) users to communicate over packet switched data networks that have interfaces complying with Recommendation X.25 (Geneva 1980) of the International Telegraph and Telephone Consultative Committee (CCITT). It allows SNA programs to communicate with SNA equipment or with non-SNA equipment over such networks.

XID. Exchange identification.

XLR. External logging replicator.

XOR. Logical operation exclusive-or.

Numerics

4700 Processor. IBM Finance Communication System 4701 Controller Model 3 and IBM 4702 Branch Automation Processor, unless otherwise described.

Bibliography

This bibliography includes publications cited in this book and other publications on related topics. Where a shortened title is used in the text, the short title is listed after the full title.

IBM LANDP Family

IBM LANDP Family: Introduction and Planning.
GC34-5529.

Short title: *LANDP Introduction and Planning.*

IBM LANDP Family: Installation and Customization.
GC34-5530.

Short title: *LANDP Installation and Customization.*

IBM LANDP Family: Programming Guide.
SC34-5781.

Short title: *LANDP Programming Guide.*

IBM LANDP Family: Programming Reference.
SC34-5531.

Short title: *LANDP Programming Reference.*

IBM LANDP Family: Servers and System Management. SC34-5532.

Short title: *LANDP Servers and System Management.*

IBM LANDP Family: Problem Determination.
GC34-5533.

Short title: *LANDP Problem Determination.*

IBM Financial Branch System Services Licensed Programs

IBM FBSS Licensed Programs Family General Information. GC19-5172.

IBM FBSS Licensed Programs Family Installation and Customization. SC19-5173.

IBM FBSS Licensed Programs Family Program Description. SC19-5176.

IBM FBSS Licensed Programs Family Version 2 Programmer's Reference. GA19-5450.

IBM FBSS Licensed Programs Family Version 2 Application Programming. SC19-5174.

IBM Financial Branch System Integrator Licensed Programs

Financial Branch System Integrator and Financial Branch System Integrator/2 General Information.
GC19-5187.

Financial Branch System Integrator Programmer's Reference Manual. GA19-5452.

Financial Branch System Integrator/2 Programmer's Reference Manual. SC19-5188.

IBM Transaction Security System

IBM Transaction Security System: General Information Manual and Planning Guide.
GA34-2137.

IBM Transaction Security System: Programming Guide and Reference. SC31-2934.

Banking Self-Service

IBM 4721 Self-Service Document Printer Programmer's Reference. GA19-5342.

IBM 4731/38/39 Personal Banking Machines P-Models Software Customization and Programming Reference. GA19-5462.

IBM 4733 Teller Assist Unit Programmer's Reference. GA19-5425.

IBM 4737 Self-Service Transaction Station Programmer's Reference. GA19-5408.

IBM Financial Application Development Toolkit Version 2 Program Description and Operation.
SB11-8461.

IBM workstations

PC DOS 7 Technical Update. GG24-4459.
PC DOS 7 User Guide. S83G-9260.
PC DOS 7 Command Reference. S83G-9309.
PC DOS 7 Keyboard and Code Pages. S83G-9310.
IBM TCP/IP Version 2.1.1 for DOS: Installation and Administration. SC31-7047.
IBM TCP/IP Version 2.1.1 for DOS: User's Guide. SC31-7045.
IBM TCP/IP Version 2.1.1 for DOS: Programmer's Reference. SC31-7046.
OS/2 Warp Version 4 Up and Running!. S84H-3098.
OS/2 Warp Server for e-Business. SG24-5393.
OS/2 Warp, PM Programming Reference Vol I. G25H-7190.
OS/2 Warp PM Programming Reference Vol II. G25H-7191.
OS/2 2.0 Application Design Guide. S10G-6260.
OS/2 2.0 Virtual Device Driver Reference. S10G-6310.
DB2/2 Guide. S62G-3663.
OS/2 LAN Server Network Administration Reference Volume 1: Planning, Installation and Configuration. S10H-9680.
OS/2 LAN Server Network Administrator Reference Volume 2: Performance Tuning. S10H-9681.
OS/2 LAN Server Network Administrator Reference Volume 3: Network Administrator Tasks. S10H-9682.
IBM Systems Application Architecture Common Programming Interface Dialog Reference. SC26-4356.
IBM Systems Application Architecture Common Programming Interface Presentation Reference. SC26-4359.
IBM OS/2 Programming Tools and Information V1.3 Programming Guide. S91F-9259.
TCP/IP for OS/2 Warp Programming Reference, SC31-8407.

IBM Network SignON Coordinator/2 Getting Started. S96F-8629.

IBM RISC System/6000®

AIX SNA Server/6000: User's Guide. SC31-7002.
AIX SNA Server/6000: Transaction Program Reference. SC31-7003.
AIX SNA Server/6000: Configuration Reference. SC31-7014.
IBM AIX V3.2 Commands Reference for RISC System/6000, Volume 1. GC23-2376.
IBM AIX V3.2 Commands Reference for RISC System/6000, Volume 2. GC23-2366.
IBM AIX V3.2 Commands Reference for RISC System/6000, Volume 3. GC23-2367.
IBM AIX V3.2 Commands Reference for RISC/6000, Volume 4. GC23-2393.
SNA Transaction Programmer's Reference for LU Type 6.2. GC30-3084.
Assembler Language Reference for IBM AIX Version 3 for RISC System/6000. SC23-2197.
General Programming Concepts for IBM RISC System/6000. SC23-2205.
IBM AIX V3.2 User Interface Programming Concepts, Volume 1. SC23-2404.
IBM AIX NetBIOS on Token-Ring/6000. SC23-2336.
Managing Application Software with the Resource Management System. SC33-9110.
IBM AIX Windows Programming Guide. GG24-3382.
Writing a Device Driver for IBM AIX V4.1. SC23-2593.
IBM AIX Calls and Subroutines Reference for RISC System/6000. SC23-2198.
IBM AIX Communications Programming Concepts for RISC System/6000. SC23-2206.
IBM AIX for RISC System/6000 Performance Monitoring and Tuning Guide. SC23-2365.

¹ This information is available in multiple languages. Contact your IBM representative for ordering information.

IBM AIX Files Reference for RISC System/6000. SC23-2512.

AIX V3.2 Topic Index and Glossary. GC23-2201.

IBM RISC System/6000 Planning for Your System Installation V3.2. GC23-2407.

IBM RISC System/6000 System Overview V3.2. GC23-2406.

IBM RISC System/6000 CD-ROM Hypertext Information Base Library. SC23-2163.

AIX V3.2 System Management Guide: Operating System and Devices. GC23-2486.

AIX 4777/4778 Programming Guide. SA34-2358.

IBM Local Area Network

IBM Token-Ring Network: Introduction and Planning Guide. GA27-3677.

IBM Token-Ring Network: Problem Determination Guide. SX27-3710.

Local Area Network: Administrator's Guide. GA27-3748.

*IBM PC Network: Technical Reference.*²

*IBM Personal Computer LAN Support Program.*²

*IBM Personal Computer Baseband and Broadband.*²

IBM Cabling System Planning and Installation Guide. GA27-3361.

Using the IBM Cabling System with Communication Products. GA27-3620.

IBM Token-Ring Network Architecture Reference. SC30-3374.

IBM Local Area Network Technical Reference. SC30-3587.

IBM Local Area Network Support Program User's Guide. SC21-8288.

IBM 3270

IBM 3270 Personal Computer Control Program Programming Guide. SC23-0165.

IBM 3270 Information Display System Character Set Reference. GA27-2837.

IBM PC 3270 Emulation Program, Entry Level V2.0 Programmer's Guide. S91F-8583.

IBM 3270 PC High Level Language API Programming Reference. SC23-2473.

Personal Communications Version 4.3 Emulator Programming. SC31-8478.

Wide Area Communications

SNA Primary Custom Feature Description. GC31-2509.

Advanced Function for Communications: System Summary. GA27-3099.

System Network Architecture (SNA) Technical Overview. GC30-3073.

System Network Architecture (SNA) Format and Protocol Reference Manual. SC30-3112.

System Network Architecture (SNA) Formats. GA27-3136.

System Network Architecture (SNA) Format and Protocol Reference Manual: Management Services. SC30-3346.

System Network Architecture (SNA) Sessions between Logical Units. GC20-1868.

CCITT X.25 Recommendations, Interface between Data Terminal Equipment (DTE) and Data Circuit Terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks. Vol. VIII. Fascicle VIII.5. This document is useful when writing X.25 applications.

RT PC X.25 Communication Support User's Guide. SC33-0630.

X.25 Interface for Attaching SNA Nodes to Packet-Switched Data Network General Information Manual. GA27-3345.

RT PC X.25 Communications Support Programmer's Reference. SC33-0631.

IBM Cryptographic Subsystem Concepts and Facilities. GC22-9063.

IBM X.25 Co-Processor Support Program User's Guide. X07F-8915.

IBM X.25 Co-Processor Support Program Programmer's Reference. X07F-8916.

² This publication is shipped with the product. Contact your IBM Representative for ordering information.

IBM X.25 Interface Co-Processor/2 Technical Reference. S16F-1879.

SNA Advanced Peer-to-Peer Networking Dependent LU Requester Architecture Reference. SV40-1010.

Multiprotocol Transport Networking (MPTN) Architecture: Formats. GC31-7074.

Multiprotocol Transport Networking (MPTN) Architecture: Technical Overview. GC31-7073.

Telnet Protocol Specification, STD 8, RFC 854, USC/Information Sciences Institute. J. Postel and J. Reynolds .

To view this book, use the keyword **RFC 854** with an internet search engine.

Printed copies of RFCs are available for a fee from:

SRI International, Room EJ291
333 Ravenswood Avenue
Menlo Park, CA 94025
(415) 859-3695
(415) 859-6387
FAX (415) 859-6028

IBM NetView

NetView Distribution Manager: General Information V1.6. GH19-6792.

NetView Distribution Manager: Planning. SH19-6589.

NetView Distribution Manager Release 6: Installation and Customization. SH19-6794.

NetView Distribution Manager: Operation. SH19-6592.

NetView Distribution Manager: User's Guide V1.6. SH19-6795.

NetView Distribution Manager: Diagnosis R5. LY19-6374.

NetView Distribution Manager: Messages and Codes V1.6. SH19-6798.

IBM Financial I/O Devices

IBM 4009 Operator's manual. GA19-5650.

IBM 4009 Service manual/Parts catalogue. SY19-6392.

IBM 4009 Quick Reference Card. GX11-6316.

IBM 4009 Customer Setup. GA19-5651.

IBM 4009 Safety Instructions. GA19-5651.

IBM 4009 Product and Programming Description (PPD) DOS. SH19-4015.

IBM 4009 Product and Programming Description (PPD) OS/2. SH19-4038.

IBM 4700 Finance Communication System Summary. GC31-2016.

IBM 4700 Financial I/O Planning Guide. GC31-3762.

IBM 4700 Financial I/O Devices Programming Guide. GC31-3770.

IBM 4700 Financial I/O Devices Programming Guide for OS/2. GC31-2661.

IBM 4700 Finance Communication System, Controller Programming Library, Volume 5, Cryptographic Programming. GC31-2070.

IBM 4700 Financial I/O Devices Operating Guide. SC31-3763.

IBM 4712 Transaction Printer Models 1, 2, and 3 Reference Card. SC31-3765.

IBM 4722 Document Printer Model 3 Programming Addendum. GC31-2928.

IBM 4722 Document Printer Models 1, 2, and 3 Reference Card. SC31-3767.

IBM 4748 Document Printer Programming Guide. SA34-2090.

IBM 4748 Document Printer Operating Guide. SA34-2068.

IBM 4748 Document Printer Service Guide. SA34-2091.

IBM 4770 Ink Jet Transaction Printer Product Profile. G571-0276.

IBM 4772 Universal Financial Printer Model 1 Programming Guide. SA34-2199.

IBM 4772 Universal Financial Printer Model 1 and 2 Installation and Operating Guide. GA34-2192.

IBM 4772 Universal Financial Printer Model 1 and 2 Reference Card. GX31-2077.

IBM 4772 Universal Financial Printer Model 1 and 2 Service Guide. SA34-2193.

IBM 4777 Magnetic Stripe Unit Installation and Operating Guide. GA34-2189.

IBM 4777 Magnetic Stripe Unit: Programming Guide for OS/2. SA34-2194.

IBM 4777 Magnetic Stripe Unit: Programming Guide for DOS. SA34-2195.

IBM 4778 PIN-Pad Magnetic Stripe Reader Installation and Operating Guide. GA34-2190.

IBM 4778 PIN-Pad Magnetic Stripe Reader: Programming Guide for OS/2. SA34-2196.

IBM 4778 PIN-Pad Magnetic Stripe Reader: Programming Guide for DOS. SA34-2197.

IBM 4777 Magnetic Stripe Unit and 4778 PIN-Pad Magnetic Stripe Reader AIX Programming Guide. SA34-2358.

IBM 9055-001 Document Printer: Planning and Programming Guide. SA18-7496.

IBM 9055-002 Document Printer: Planning and Programming Guide. SA18-7489.

IBM 9068 Multi-Purpose Passbook Printer Model D01 Planning and Programming Guide. SA18-7505.

IBM 9068 Multi-Purpose Passbook Printer Model S01 Planning and Programming Guide. SA18-7506.

IBM 9068-S01 Multi-Purpose Passbook Printer Operating Guide. SA18-7507.

IBM 9069 Printer Planning and Programming Guide. SA18-7525.

IBM 9069 Operating Guide. SA18-7524.

Distributed Computing Environment

AIX DCE Overview. SC23-2477.

DCE Administration Guide. SC23-2475.

Introduction to DCE. Prentice Hall Inc.

DCE User's Guide and Reference. Prentice Hall Inc.

DCE Administration Reference. Prentice Hall Inc.

DCE Application Development Guide. Prentice Hall Inc.

DCE Application Development Reference. Prentice Hall Inc.

IBM DCE for OS/2: Application Developer's Guide. S96F-8506.

Encryption and Decryption

IBM Cryptographic Subsystem Concepts and Facilities. GC22-9063.

IBM 4700 Finance Communication System, Controller Programming Library, Volume 5, Cryptographic Programming. GC31-2070.

IBM Transaction Security System Workstation Security Services: Installation and Operating Guide. SA34-2141.

IBM Transaction Security System Concepts and Programming Guide: Volume 1, Access Controls and DES Cryptography. GC31-3937.

IBM VisualAge C++

Product web site:

<http://www.ibm.com/software/ad/vacpp/>

IBM VisualAge Generator

Product web site:

<http://www.ibm.com/software/ad/visgen/>

IBM Redbook:

VisualAge Generator Version 3.1 System Development Guide. SG24-4230-02

IBM VisualAge Smalltalk

Product web site:

<http://www.ibm.com/software/ad/smalltalk/>

IBM Redbooks:

VisualAge for Smalltalk Handbook - Volume 1: Fundamentals. SG24-4828-00

VisualAge for Smalltalk Handbook - Volume 2: Features. SG24-2219-00

Java

IBM Java web site:

<http://www.ibm.com/software/java/>

IBM VisualAge for Java product web site:

<http://www.ibm.com/software/ad/vajava/>

IBM developerworks web site:

<http://www.ibm.com/software/developer/java/>

IBM VisualAge Developers Domain web site:

<http://www.ibm.com/software/vadd/>

IBM Redbooks:

Programming with VisualAge for Java Version 2. SG24-5624-00

Application Development with VisualAge for Java Enterprise, SG24-5081-00

IBM Personal Communications

Personal Communications AS/400 and 3270 for OS/2 Up and Running, SC31-8258.

Personal Communications AS/400 and 3270 for Windows NT Up and Running, GC31-8314.

Personal Communications/3270 Programmer's Guide for DOS (Entry Level), S20H-1774.

Personal Communications/3270 Programmer's Guide for OS/2, S85G-8681.

Personal Communications/3270 Reference Guide for OS/2, S85G-8721.

Personal Communications Version 4.3 for Windows 98 and Windows NT Reference, Volumes 1 and 2, SC31-8682, SC31-8680.

Personal Communications Windows NT Quick Beginnings, GC31-8679.

IBM Communications Server

IBM SecureWay Communications Server for OS/2 Warp Version 6 Quick Beginnings, GC31-8189.

IBM Communications Server for Windows NT Quick Beginnings, GC31-8424.

eIBM Network Communications Server for OS/2 Guide to AnyNet, GC31-8193, GC31-8320

WorkSpace On-Demand

WorkSpace On-Demand Road Map Release 2.0, SG24-5117 (10/98)

WorkSpace On-Demand Handbook Release 2.0, SG24-5117 (10/98)

WorkSpace On-Demand Handbook (Release 1), SG24-2028 (12/97)

WorkSpace On-Demand Early Customer Experiences, SG24-5107 (10-98)

IBM Up and Running! OS/2 Warp Server, S25H-8004

WorkSpace On-Demand Administrator's Guide., on the web at www.ibm.com/software/network/workspace/library

MQSeries

MQSeries for Windows NT V5.0 Quick Beginnings, GC33-1871-00

MQSeries for OS/Warp Quick Beginnings, GC33-1868-01

MQSeries Planning Guide, GC33-1349-05

MQSeries Intercommunication, SC33-1872-00

MQSeries Clients, GC33-1632-04

MQSeries System Administration, SC33-1873-00

MQSeries Command Reference, SC33-1369-08

MQSeries Programmable System Management, SC33-1482-05

MQSeries Messages, GC33-1876-00

MQSeries Application Programming Guide, SC33-0807-07

MQSeries Application Programming Reference, SC33-1673-03

MQSeries Using C++, SC33-1877-00

Index

Numerics

- 1000-mode, 4707 232
- 2000-mode, 4707 232
- 3270
 - characteristics 26
 - display 25
 - file transfer programs
 - DBCS workstation to host 226
 - host computer requirements 212
 - hotkeys 27
- 3270 books 287
- 3270 emulator 25, 28
 - See also 3270 emulator
 - See also 3270 send/receive facility
 - common codes 34
 - display 29
 - displaying the emulation screen 29
 - ending 29
 - enter hotkey 29
 - exit hotkey 29
 - FBSS3270.SCN file 37
 - hotkey definitions 29
 - operator information area 29
 - operator information area 29
 - operator information area (OIA) 29
 - operator interface 182
 - printing panels 37
 - resetting insert mode 37
 - saving panels 37
 - starting 28
 - status indicator codes 34
 - X.25 network 29
- 3270 send/receive facility 225
 - APVUFILE 226, 227
 - CICS (MVS) 231
 - DBCS systems, for 225
 - file transfer facility, alternative to 225
 - IND\$FILE 226, 227
 - RECEIVE command 227
 - SEND command 227
 - TSO (MVS) 230
 - TSO (MVS) SEND 229
 - VM/CMS RECEIVE 228
 - VM/CMS SEND 228
- 3274 Model 51C Control Unit 25
- 3278 Model 2 Display Station 25
- 3279 Model S2A Display Station 25
- 3287 emulator 38
 - operator interface 182
- 3287 printer emulator 41
- 4009 printer server 68
- 4019 printer 40
- 4029 printer 40
- 4039 printer 40
- 4700 file transfer facility 212
- 4707 display modes 232
- 4707 monochrome display support 232
- 4712 printer server 68
- 4717 MSR/E server 76
- 4718 PIN pad server 77
- 4722 printer server 68
- 4748 printer server 72
- 4770 printer server 73
- 4772 printer server 68
- 4777 MSR/E server 76
- 4778 PIN pad server 77
- 480-mode, 4707 232
- 5575 Printer 41
- 5577 Printer 41
- 9055-002 printer server 68
- 9068 printer server 68
- 9069 multi-purpose transaction printer 68

A

- A/B keys 69
- access
 - DDE access server 198
 - identifying files 46
 - LAN common data 157
 - to host data 25
 - to relational database 3
 - to workstation from host 43
 - user profile application data 160
 - user profile data 158
- access methods, shared-file 80
- ACSRDECR.DLL 15
- ACSRENCR.DLL 15
- advanced program-to-program API 10
- AIX, SNA Services/6000 12
- alerts management 149

- aliases, defining 46
- alternate print mode 74
- APAR level 236
- APPC, API 10
- application data
 - other's, getting 147
 - own, getting 147
- application integration programs 197
 - batch machine log file 200
 - dce_login 208
 - LANDP-DCE interoperability 208
 - mail program 200
- application integration servers 197
- application libraries, maintaining 43
- applications
 - effect of XLR takeover 93
 - on host 25
 - sharing 137
- APPLYFIX utility 236
- applying program fixes 236
- APVUFILE transfer program 226, 227
- ASCII data 45
- ASCII to EBCDIC
 - translation 13, 239
 - translation table 41
- asynchronous operation 12
- audio signal
 - customizing system manager 152
 - pending message 164
- authorization level, getting 147
- authorized program analysis report 236
- AUTOEXEC.BAT 55
- AUTOFBSS command 237
 - including EHCINFO.EXE 237
- AUTOFBSS file
 - EHCINFO.EXE 237
 - example 238
- AUTOFBSS.BAT file
 - DOS PRINT 75
 - OfficeVision 225
 - RCMS loading statement 55
- AUTOFBSS.BAT file.
 - OfficeVision. 225

B

- backout
 - batch machine operator 205
 - log files 82
 - transactions 82

- backup
 - DBBACKUP shared file utility 103
 - DBRESTOR shared file utility 104
 - dynamic reconfiguration of XLR servers 193
 - example backup procedures
 - DOS, OS/2, Windows NT, AIX 104
 - global data 146
 - shared files 103—105
 - system communication 3
- BACKUP2 query server program 118
- banking self-service books 285
- batch machine
 - facility 199
 - loading server 199
 - operator 204
- batch machine log file 200
- BCK.BAT backup procedure for LANDP for DOS 103
- BCK.COMD backup procedure for LANDP for OS/2 103
- bibliography 285
 - 3270 287
 - banking self-service 285
 - Communications server 290
 - Distributed Computing Environment 289
 - encryption and decryption 289
 - FBSS 285
 - Financial Branch System Integrator 285
 - Financial I/O Devices 288
 - LANDP 285
 - Local Area Network 287
 - NetView 288
 - Personal Communications 290
 - Personal Computer 286
 - Personal System/2 286
 - RISC System/6000 286
 - Transaction Security System 285
 - VisualAge C++ 289
 - VisualAge Generator 289
 - wide area communications 287
- binary file distribution 45
- BIOS INT 17H 70
- block transmission 135
- books for LANDP xv
- browse the system log file 162
- buffers
 - for data link controls 13
 - for SDLC, number 8
 - for TRDLC, number 9
 - for X25DLC, number 9

C

- calculating the number of buffers 8
- call packets 14
- CCITT 1980 and CCITT 1984 7
- chaining emulators 27
- change modes, 4707 232
- character set
 - 3270 translation tables 26
 - support 69
- CICS interface server 197
- CICS/VS support 136
- clients, testing 237
- clipboard services, DDE access server 199
- code page 850 239
- code pages 69
 - DBCS, for 3270 emulator 27
- collate tables 89
- COMMAND.COM 55
- commands, Tivoli NetView DM 52
- committed transactions 82
- common data maintenance 145
- common status indicator codes (3270 emulator) 34
- communication
 - backup 3
 - books 287
 - layers 4
 - monitored 43
 - servers 3
 - with host 25
 - X.25 25
- communications
 - conversation 20
- Communications Server
 - books 290
 - for OS/2 Warp 38
 - for Windows NT 11
- compatibility 225
- component verification 237
- compression server 19
- CONFIG.SYS file
 - NLS support 70
 - SHELL statement 55
- configuring DLC buffers 13
- controlling functions 145
- conventional LU application custom feature (LUA) 10, 11
- conversation, LANDP 20
- conversion, ASCII to EBCDIC 45
 - translation mode in RCMS 45

- copies of database 112
- cryptographic interface 15
- cryptography
 - books 289
- cryptography interface
 - errors 32
 - LANDP for OS/2 SNA server 43
 - server-controlled 43
 - SNA session level 43
- ctrl-print screen 74
- current owner of print port 75
- current pointer 81
- current version 236

D

- data
 - conversion 43
 - distribution 43
 - getting in link format from clipboard 199
 - maintenance, workgroup 145
 - translation 48
 - validation 145
- data link
 - buffer pools 7
 - control 4
 - support 4
- data management
 - servers 79
- data migration 252
- database
 - access, relational 3
 - copies 112
 - description file 81
 - disjoint 110
- date and time synchronization 145
- DB2 for OS/2 xiv
- DB2 Universal Database xiv
- DBBACKUP utility 103
- DBCS
 - 3270 send/receive facility intended for 225
 - and 3270 emulator 27
 - KS (Korean Standard) codes 72
 - mode restrictions
 - batch machine operator 205
 - shared-file server 90
- DBD file 81
- DBRESTOR utility 104
- DCADLC 4

- DCE
 - books 289
- DCE support 206
- dce_login 208
- dczyrdf utility 240
 - input file 241
- dczyxsvp file 237
- DDE
 - access server 198
 - Server Application 198
- deadlock detection 80
- decryption and encryption books 289
- defined record structures, retrieving 145
- defined users, listing 147
- defining
 - aliases 46
 - DLC buffers 13
 - print formats 69
 - printer formats 72
- defining jobs (batch machine operator) 204
- definitions of terms 263
- DELETE command 53
- deleting files (batch machine operator) 205
- destination batch machine (batch machine operator) 204
- device servers 67
- direct access, shared-file 80
- directory tasks 138
- disjoint database data 110
- disk space management 81
- display attributes, 3270 26
- display characteristics, 3270 25
- display messages for the operator 164
- display modes, 4707 232
- display support, 4707 232
- display system status and enter commands 155
- displaying
 - LANDP for AIX status 239
 - server status 237
- Distributed Computing Environment 206
 - See *also* DCE
- distributed database 110
- Distributed Relational Database Architecture 114
- distributing production data 43
- distribution, binary/test files 45
- distributor server 110
- DLC 4
- DLC buffer definition 13
- document printer server 68

- DOS
 - H7.0 72
 - print function 74
- DOS box (VDM) 233
- DOS EXEC facility 223, 224
- DOS office direct connect 225
- download code pages 70
- DRDA 114
- DSA 198
- dynamic data exchange 198
- dynamic logging 82

E

- EBCDIC data 45
- EBCDIC MLP code page 500 239
- EBCDIC-to-ASCII translation 13
- EBCDIC/ASCII translation table 41
- EHCCINFO.EXE utility program 237
- EHCDOSQ 251
- EHCINFO.EXE utility program 237
- EHCMGR1 249
- EHCMGR2 252
- EHCMGROS 251
- EHCS2Q 251
- EHCRCMS.LOG 62
- EHCRCMS.MSG 61
- EHCSAM, service availability manager 193
- EHCSQL server migration 245
- EHLAPI 38
- EJUTIL.EXE program for electronic journal data sets 128
- electronic journal server 123—128
 - data integrity in LANDP for AIX 125
 - data integrity in LANDP for DOS, OS/2, and Windows NT 125
 - DBCS mode restrictions 127
 - environments 126
 - files 124
 - search operations 126
 - server utility 128
- electronic journal server utility 128
- emulator
 - 3287 38
 - print modes 39
 - support 25
- encrypting SNA session data 15
- encryption and decryption
 - books 289

- examples
 - CICS (MVS) RECEIVE command 219
 - CICS (MVS) SEND command 214
 - record format definition file 243
 - TSO (MVS) RECEIVE command 221
 - TSO (MVS) SEND command 216
 - VM/CMS RECEIVE command 222
 - VM/CMS SEND command 218
- exchanging data with host 25
- exclusive print mode 74
- extended logging replicator (XLR)
 - dynamic reconfiguration of XLR servers 193
 - maintaining state information for XLR servers 193
 - message broadcast facility. 194
- external logging replicator (XLR) 80, 90—96
 - effect on applications of XLR takeover 93
 - log files 93
 - messages 92
 - relationship between components 94
 - servers in an XLR configuration 91
 - service availability manager (EHCSAM) 92

F

- facilities 211
 - 3270 send/receive 225, 227
 - 4707 monochrome display support 232
 - file transfer commands 212
 - Microsoft Windows 3.1 support 234
 - multiple virtual DOS machines 233
 - MVDM 233
 - Windows 3.1 support 235
- FBSI 4
- FBSS
 - books 285
- FBSS3270.SCN file (3270 emulator) 37
- FBSSP command 75
- file
 - CONFIG.SYS 70
 - distribution, binary/test 45
 - host access 43
 - integrity 82
 - locking 82
 - pointer 81
 - RCMSPLAN 46
 - sharing 137
 - STARTUP.CMD 70
 - transfer
 - facility 211
 - programs, 3270 212, 226

- file access 46
- FILESYS utility 138, 141
- financial printer server 68
- financial printers 40
- fix records 236
- font code page 69
- forward recovery 82
- forwarding priority 135
- forwarding server 134—137
 - DBCS mode restrictions 137
 - forwarding stored records 132
 - forwarding priority 132
 - forwarding sessions 132
 - host applications 132
 - transmission 134
 - block transmission 135
 - CICS/VS support 136
 - host computer considerations 136
 - IMS/MVS 136
 - modes of transmission 135
 - signon feature 135
 - Transmission priority 135
- functions, controlling 145
- functions, system manager operator 154

G

- gateway 16
- GENLOG utility 97
- GENLOG2 query server program 117
- getting
 - authorization level 147
 - other's application data 147
 - other's profiles 147
 - own application data 147
 - own profile 147
 - PC ID 147
- global data, backup 146
- glossary 263

H

- Hewlett Packard print mode 39
- HLLAPI, LANDP 38
- host access 25
- host access to workstation 43
- host data stream errors 32
- host files
 - receiving 211, 225
 - sending 211, 225

- host printing support 38
- host resource tracking 43
- host/workstation data conversion 43
- hotkey
 - 3270 27
 - operator interface 182
 - system manager 152
- HP print mode 39

I

- I/O books 288
- I/O device servers 67
- ID
 - local session 13
 - session 13
- IMS/MVS 136
- IMS/VS 136
- inactive queues 200
- IND\$FILE transfer program 225, 226, 227
- indexed access, shared-file 80
- indexed sequential access 81
- information about IBM products 285
- information display system terminal 25
- information sources, public 3
- INITIATE CLIST command 54
- ink jet transaction printer server 73
- inner node 110
- input queue (batch machine operator) 205
- INT X'7F' API 234
- integration servers 197
- interface
 - printer manager 74
 - printer server 70
- Interface Co-processor/2 6
- interoperability, LANDP-DCE 206
- IPTXFER.MSG message file 223

J

- Java websites and redbooks 289
- journal server 123

K

- key assignment set, 3270 26
- keys, shared-file 80
- Korean Standard (KS) codes 72
- KS (Korean Standard) codes 72

L

- LAN books 287
- LAN common data 157
- LAN system status and common data 168
- LANDP
 - 3270 emulator 25
 - 3287 emulator 38
 - conversation 20
 - emulator support 25
 - HLLAPI 38
 - session 13
 - status, displaying 239
- LANDP family books xv
- LANDP for AIX
- LANDP for AIX record definition utility 240
- LANDP-DCE interoperability 206
- layers, communication 4
- leaf node 110
- leased connections 5
- libraries, maintaining 43
- line speed 5
- list of files (batch machine operator) 204
- listing
 - defined users 147
 - signed-on users 147
- load servers inline utility 234
- local resource manager server 192
- local session ID 13
- local sessions, maximum 14
- lock management 80
- lock management, shared-file server 89
- locking DDE topics 198
- log file 82
 - shared-file server 82
- log file, query server 116
- log file, shared-file server 82
- log files
 - displaying 98
 - generating 97
 - GENLOG utility 97
 - listing 98
 - managing 84
 - opening 84, 103
 - OPENLOG utility 103
 - TYPELOG utility 98
- log files, XLR 93
- log management 145
- logical unit of work 198
- logical unit pooling 17

- low-level INT X'7F' API 234
- LSI 234
- LU pooling 17
- LU_1 session 40
- LU_2 session 25
- LUA 10, 11

M

- magnetic stripe reader/encoder server 76
- MAIL program 200
- maintaining
 - application libraries 43
 - workgroup data 145
- managing
 - alerts 145, 149
 - logs 145
 - phases 46
 - record locks 80
 - server data 151
 - user identification 145
 - user profile 145
- manually dialed connections 5
- maximum SDLC speed 5
- messages
 - external logging replicator (XLR) 92
 - operator 43
 - sending 147
 - support 145
 - system manager 152
 - translation 13
- messages for the operator 181
- Microsoft SNA Server 11
- Microsoft Windows 3.1 support 234
- MIG.EXE 252
- migration 245
 - data 252
 - from shared-file to SQL data 254
 - LANDP for DOS to LANDP for AIX 256
 - LANDP for DOS to LANDP for OS/2 254
 - LANDP for OS/2 SQL data 256
 - LANDP for OS/2 to LANDP for AIX 257
 - MIG.EXE 252
 - program menu 253
 - required files 253
 - shared-file fields to SQL columns 254
 - shared-file to query server 252
 - data migration utilities 248
 - data migration utilities, LANDP for OS/2
 - EHCOSQ 251
 - EHCMGR1 249

- migration (*continued*)
 - data migration utilities, LANDP for OS/2 (*continued*)
 - EHCMGR2 251
 - EHCMGROS 251
 - EHCOS2Q 251
 - query server from FBSS/2 to LANDP for OS/2 245
 - shared-file server and SQL data to LANDP for AIX 247
 - shared-file server to LANDP for OS/2 query server 246
- MODEENT macro in VTAM logon mode table 226
- monitored communication 43
- monochrome display support, 4707 232
- MQSeries Link server 23
- MSR/E server 76
- multiple copies of database 112
- multiple virtual DOS machine relay 233
- multipoint line 5
- multitasking programs 234
- MVDM 233
- MVS with CICS
 - RECEIVE command 218, 231
 - SEND command 213, 231
- MVS with TSO
 - RECEIVE command 220, 230
 - SEND command 214, 229

N

- naming conventions, DDE 198
- National Language Support for printers 70
 - OS/2 70
 - Windows NT 71
- native X.25 communication 3
- native X.25 server 19
- NetView books 288
- NetView Distribution Manager 44
- NetView DM 44
- NetView DM commands 52
 - DELETE 53
 - INITIATE CLIST 54
 - REPLACE 57
 - RETRIEVE 58
 - RETRIEVE by CATALOG 60
 - SEND 60
 - SEND MESSAGE 61
- NetView DM resources, identifying 46
- network packet switching interface (NPSI) 16
- Network SignON Coordinator/2 147

- NLS 89
- NLS for printers
 - OS/2 70
 - Windows NT 71
- node
 - inner 110
 - leaf 110
 - problem management 43
 - root 110
- non-Windows programs, multitasking 234
- NPSI (network packet switching interface) 16
- number of buffers
 - for data link controls 13
 - for SDLC 8
 - for TRDLC 9
 - for X25DLC 9
- number of printer port 75

O

- object post box server 199, 204
- ODBC query server 120
 - creating connection criteria 122
 - data source creation 122
 - migrating from OS/2 query server 247
 - ODBC conformance levels 121
 - ODBC query server utilities 122
 - OPENODB 122
 - operating modes 121
- OfficeVision 225
- OIA 25
- OIA (operator information area) 29
- OPBS 199
- open conversations, DDE 199
- open database connectivity
 - See ODBC query server
- Open Systems Foundation 206
- open transactions 82
- OPENDB2 query server program 118
- OPENLOG utility 103
- operating modes 80
- operator
 - information area 25
 - interface 182
 - hotkey 182
 - messages 43
 - support 145
 - system manager 152
- operator information area (OIA) 29
- operator interface
 - 3270 emulator 182
 - 3287 printer emulator 182
 - actions 185
 - activating 182
 - BPP 182
 - commands 185
 - ending 182
 - FBSI 182
 - function keys 189
 - help panels 183
 - messages 182
 - presentation panels 183
 - printer manager 182
 - return codes 182
 - sample panels 190
 - starting 182
 - status and command panel 182
 - Status and Command panels 183
 - status messages 183
- OS/2
 - MVDM 233
- OS/2 Warp H3.0 72
- OSF 206

P

- packets, call 14
- paper jams 41
- PC codes 72
- PC ID for userid, getting 147
- PC X.25 communications support program 6
- PC/host file transfer 25
- PCB table 81
- PCL4 and PCL5 print modes 39
- Personal Communications AS/400 and 3270 11
- Personal Communications books 290
- personal computer books 286
- personal identification number (pin) pad server 77
- personal printer data stream 39
- phase management 46
- phase remote release 48
- physical layer 4
- physical unit 6, 17
- PIN pad device server 77
- PLAYLOG2 query server program 119
- point-to-point line 5
- pooling, LU 17
- port status 75

- post, telephone, and telegraph administration 7
- power-on reset, printer 69
- PPC
 - communication 3
 - server 20
 - service requests 10
- PPDS 39
- preparing plans utility 45
- print formats 69
- print modes 74
 - emulator 39
- printer formats, defining 72
- printer manager
 - interface 74
 - server 74
- printer manager server operator interface 182
- printer POR 69
- printer server 68
 - interface 70
- printing host data 38
- problem determination 149
- problem management 43
- process ID 20
- process results, RCMS 62
- production data distribution 43
- profile
 - other's, getting 147
 - own, getting 147
- program control blocks 81
- program fixes, applying 236
- program temporary fix 236
- program-to-program communication 3, 13
- program-to-program server 20
- programs, sharing 137
- protocol handling 4
- PTF 236
- PTT parameters 7
- PU 6, 17
- public information sources 3

Q

- query server 114, 116
 - migrating shared-file server data 247
 - migration from FBSS/2 to LANDP for OS/2 245
 - migration from shared-file server 246
 - utilities 248
 - utilities, LANDP for OS/2 117
- query server migration 245

- query server utilities, LANDP for OS/2 248
 - BACKUP2 118
 - EHCDOSQ 251
 - EHCMGR1 249
 - EHCMGR2 251
 - EHCMGROS 251
 - EHCOS2Q 251
 - GENLOG2 utility 117
 - OPENDB2 118
 - PLAYLOG2 119
 - RECOVER2 119
 - RESTORE2 118
 - TYPELOG2 utility 117
- queues, inactive 200

R

- RCMS 43
- RCMS.LNF, creating and modifying 47
 - LNFEEDIT program, using correct VDM for 48
- RCMSPLAN files 46
- RCMSPLAN.NEW 48
- RCMSPLAN.OLD 48
- RCMSPLAN.TXT 48
- Realtime Interface Co-Processor DOS Support 6
- RECEIVE command 212, 227
- RECEIVE messages 223
- RECEIVE.COM and RECEIVE.EXE 226
- receiving
 - files 211, 225
 - resources 43
- record definition utility 240
- record definitions, system manager 152
- record locks, managing 89
- record of applied fixes 236
- record structures, retrieving 145
- recording transactions 123
- records, variable length 45
- RECOVER2 query server program 119
- recovery, forward 82
- relational database access 3
- releasing phases 46
- remote change management services (RCMS) 43
 - defining aliases 46
 - EHRCMS.LOG 62
 - error log 62
 - file access 46
 - NetView DM commands 52
 - RCMSPLAN.NEW 48
 - RCMSPLAN.OLD 48

- remote change management services (RCMS)
 - (continued)
 - RCMSPLAN.TXT 48
 - reporting process results 62
 - requesting phase remote release 48
 - translating data 48
 - transmission log 62
- remote node problem management 43
- remote release, phase 48
- REPLACE command 57
- replicator server 112
- resources
 - receiving 43
 - sending 43
 - tracking 43
- resources assigned 75
- RESTORE2 query server program 118
- restoring shared files 104
- RETRIEVE by CATALOG command 60
- RETRIEVE command 58
- retrieving defined record structures 145
- RISC System/6000 books 286
- rolled-back transactions 82
- root node 110
- root server 111
- RS/6000 Transaction Security System 15
- RU maximum size 8

S

- saving transactions 123
- SDLC 4
 - number of buffers 8
 - server 5
 - speed 5
- security level 142
- security system 147
- SEND command 60, 212, 227
- SEND MESSAGE command 61
- SEND messages 223
- send/receive facility
 - See 3270 send/receive facility
- SEND.COM and SEND.EXE 226
- sending
 - files 211, 225
 - messages 147
 - resources 43
- sequential access, shared-file 80
- server
 - 4717 MSR/E server 76

- server (continued)
 - 4718 PIN pad server 77
 - 4748 printer server 72
 - 4777 MSR/E server 76
 - 4778 PIN pad server 77
 - application integration server 197
 - batch machine facility 199
 - batch machine operator 204
 - CICS interface server 197
 - communication servers 3
 - compression server 19
 - data management servers 79
 - DDE access server 198
 - distributor server 110
 - document printer server 68
 - electronic journal server 123
 - financial printer server 68
 - forwarding server 134
 - ink jet transaction printer server 73
 - journal server 123
 - local resource manager server 192
 - magnetic stripe reader/encoder server 76
 - MAIL program 200
 - MSR/E server 76
 - native X.25 server 19
 - object post box server 204
 - personal identification number pad server 77
 - PIN pad server 77
 - PPC server 20
 - printer manager server 74
 - program-to-program server 20
 - query server 114
 - replicator server 112
 - service availability manager (EHCSAM) 92
 - service availability manager (SAM) 193
 - shared-file distributor server 110
 - shared-file replicator server 112
 - shared-file server 79
 - SNA server 16
 - store-for-forwarding server 128
 - system management servers 145
 - system manager server 145
 - transaction printer server 68
 - universal banking printer server 68
 - universal financial printer server 68
 - wide area communication servers 3
 - X.25 server 19
- server data, managing 151
- server interface, printer 70

- server status, displaying 237
- service availability manager (EHCSAM) 92
- service availability manager (SAM) 193
 - dynamic reconfiguration of XLR servers 193
 - maintaining state information for XLR servers 193
 - message broadcast facility. 194
- service marks 261
- session
 - LANDP 13
 - LU_1 40
 - LU_2 25
 - routing within a LANDP workgroup 13
- session cryptography, SNA 43
- session ID
 - local 13
 - workstation 13
- SET, DOS command 224
- SHARE program (DOS) 141
- shared DOS directory 137
 - authorization level 142
 - CONTINUE 140
 - FILESYS 141
 - FILESYS security 142
 - LOGOFF 140
 - LOGON 140
 - password 140
 - PAUSE 140
 - security level 142
 - SHARE 141
 - SHRDIR 140
 - SHRSHARE 138
 - SHRSHARE security 142
 - START 139
 - STOP 140
 - system security 141
 - user authorization levels 142
 - user ID 140
 - utilities 138
- shared DOS directory services 138
 - directory tasks 138
 - sharing information 138
 - stop sharing information 142
 - unloading 142
- shared files, backup 103
- shared files, restoring 104
- shared-file distributor server 110
- shared-file keys 80
- shared-file profile definition 107
- shared-file replicator server 112
- shared-file server 79
 - DBCS mode restrictions 90
 - lock management 89
 - log files 82
 - migrating data to LANDP for AIX 247
 - migration to LANDP for OS/2 query server 246
 - utilities 97, 248
 - backing up shared files 103
 - displaying log files contents 98
 - generating log files 97
 - listing log files 98
 - listing shared-file server profile definition 107
 - listing shared-file server statistics 105
 - opening log files 103
 - restoring shared files 104
 - updating a shared-file server profile 109
- shared-file server utilities
 - backing up shared files 103
 - DBRESTOR utility 104
 - displaying log files contents 98
 - generating log files 97
 - listing log files 98
 - listing shared-file profile definition 107
 - listing shared-file statistics 105
 - opening the log files 103
 - restoring shared files 104
 - TYPECONF utility 107
 - TYPESTAT utility 105
 - updating a shared-file server profile 109
 - UPDTCONF utility 109
- shared-file statistics 105
- sharing information 138
- sharing programs, files, and applications 137
- SHFILE 81
- SHRSHARE command 138
- SHRSHARE utility 138
- signed-on users, listing 147
- SIGQUIT signal 135
- simplifying RECEIVE use 224
- simplifying SEND use 224
- simultaneous code pages 70
- single message protocol 134
- SNA
 - communication 3
 - compression server 19
 - server 16
 - servers 4
 - session cryptography 43
 - session encryption 15

- sources, public information 3
- SQL
 - server interface 115
 - support 114
- SSCP-PU session 16
- STARTUP.CMD file 70
- status
 - LANDP 239
 - server 237
- status indicator codes 32
- status indicator codes (3270 emulator) 34
- stop sending job (batch machine operator) 205
- stop sharing information 142
- store-for-forwarding server 128
- storing files (batch machine operator) 205
- structure, shared file 81
- structured query language support 114
- summary of changes xvii
- SVPCPRB.EXE file 237
- synchronized databases 112
- synchronizing date and time 145
- system communication, backup 3
- system log file 176
- system log file: browse 163
- system log file: search criteria 162
- system log, managing 145
- system management
 - problem notification 149
 - resolutions 150
 - servers 145
- system manager
 - features 152
 - operator 151
 - server 145
- system manager operator functions 154
 - adding 161
 - deleting 161
 - searching 161
 - updating 161
- system manager operator, LANDP for DOS
 - access authorization 154
 - access LAN common data 157
 - access user profile application data 160
 - access user profile data 158
 - audio signal 164
 - browse the system log file 162
 - commands 156
 - display messages for the operator 164
 - display system status and enter commands 155
 - functions 154, 161
 - system manager operator, LANDP for DOS (*continued*)
 - LAN common data 157
 - main menu 154
 - options 154
 - panels 154
 - password 153
 - system log file: browse 163
 - system log file: search criteria 162
 - system manager operator functions 154
 - system status and commands 155
 - user profile: application data 160
 - user profile: system data 158
 - video signal 164
- system manager operator, LANDP for OS/2
 - access authorization 167
 - LAN system status and common data 168
 - main menu 167
 - messages for the operator 181
 - options 167
 - password 166
 - system log file 176
 - user profile data 172
 - windows 167
- system network architecture 3
- system services control point-physical unit session 16
- system software maintenance 43
- system status and commands 155
- system status area 148
- system verification utility 237

T

- TCP/IP server
 - Wide area communication options 3
- TCP/IP wide area communications server 22
 - LANDP for DOS 5
 - LANDP for OS/2 10
 - LANDP for Windows NT 11
- terminal, information display system 25
- terms, definitions of 263
- test system 237
- testing user-developed clients 237
- text file distribution 45
- time synchronization 145
- Tivoli NetView DM commands 52
- Token-ring data link control (TRDLC)
 - number of buffers 9
 - server 6
- tracking resources 43

- trademarks 261
- transaction printer server 68
- Transaction Security System 15
- transactions
 - opening 82
 - saving 123
- transferring files, PC to host 25
- translating
 - ASCII to EBCDIC 239
 - data 13, 48
- translation table, ASCII/EBCDIC 41
- transmission priority 135
- transmitting records 134
- TRDLC (Token-ring data link control) 4
 - number of buffers 9
 - server 6
- TSS 15
- TYPECONF utility 107
- TYPELOG utility 98
- TYPELOG2 query server program 117
- TYPESTAT utility 105

U

- UDC 72
- universal banking printer server 68
- universal financial printer server 68
- unloading shared DOS directory 142
- updating a shared-file server profile 109
- UPDTCONF utility 109
- use mode of printer port 75
- user identification, managing 145
- user log, managing 145
- user profile data 172
- user profile, managing 145
- user profile: application data 160
- user profile: system data 158
- user-defined character 72
- user-developed clients, testing 237
- utilities 97, 236
 - APPLYFIX utility 236
 - data migration 252
 - displaying LANDP status 239
 - displaying server status 237
 - EJUTIL.EXE 128
 - electronic journal server 128
 - FILESYS 138
 - LANDP for AIX record definition utility 240
 - LANDP for OS/2 query server 117
 - MIG.EXE 252

- utilities (*continued*)
 - preparing plans 45
 - shared file server 97
 - backing up shared files 103
 - displaying log files contents 98
 - generating log files 97
 - listing log files 98
 - listing shared-file server profile definition 107
 - listing shared-file server statistics 105
 - opening log files 103
 - restoring shared files 104
 - updating a shared-file server profile 109
 - SHRSHARE 138
 - system verification utility 237
 - translating ASCII to EBCDIC 239
 - VERSION utility 236

V

- validation, data 145
- variable length records 45
- VDM 233
- verification utility, system 237
- verification, component 237
- version number 236
- VERSION utility 236
- video signal 164
 - customizing system manager 152
 - pending message 164
- viewing files (batch machine operator) 205
- virtual circuit, maximum 7
- virtual DOS machine relay 233
- visual signal, system manager 152
- VisualAge C++ web site 289
- VisualAge Generator books 289
- VisualAge Smalltalk web site 289
- VM/CMS
 - RECEIVE command 221, 228
 - SEND command 216, 228
- VSE/ESA and VSE/SP RECEIVE commands 223
- VSE/ESA and VSE/SP SEND commands 218
- VTAM logon mode table, MODEENT macro in 226

W

- wide area communication
 - books 287
 - server 3
- Windows
 - Connection program 234

Windows (*continued*)
 programs, multitasking 234
 support 234
Windows 2000 xiv
Windows NT
 MVDM 233
workgroups, common data maintenance 145
workstation
 host access 43
 host data conversion 43
 ID 13
 resource tracking 43
workstation files, identifying 46

X

X'7F' API 234
X.25
 API 10
 Co-Processor Support Program 6
 communication 3
 Data Link Control server 6
 DLC server 6
 DLC, number of buffers 9
 Interface Co-processor/2 6
 network communication 25
 server 19
 speed 7
X.25 communications support program 6
X25DLC 4

Sending your comments to IBM

LANDP® Family

Servers and System Management

SC34-5532-00

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book. Please limit your comments to the information in this book and the way in which the information is presented.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

You can send your comments to IBM in any of the following ways:

- By mail, use the Readers' Comment Form (RCF)
- By fax:
 - From outside the U.K., after your international access code use 44 1962 870229
 - From within the U.K., use 01962 870229
- Electronically, use the appropriate network ID:
 - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
 - IBMLink: HURSLEY(IDRCF)
 - Internet: idrcf@hursley.ibm.com

Whichever you use, ensure that you include:

- The publication number and title
- The page number or topic to which your comment applies
- Your name/address/telephone number/fax number/network ID.

Readers' Comments

LANDP® Family

Servers and System Management

SC34-5532-00

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply.

Name

Address

Company or Organization

Telephone

Email



You can send your comments **POST FREE** on this form from any one of these countries:

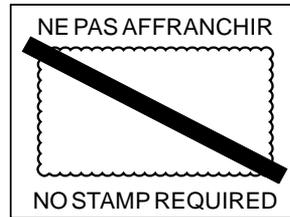
Australia	Finland	Iceland	Netherlands	Singapore	United States
Belgium	France	Israel	New Zealand	Spain	of America
Bermuda	Germany	Italy	Norway	Sweden	
Cyprus	Greece	Luxembourg	Portugal	Switzerland	
Denmark	Hong Kong	Monaco	Republic of Ireland	United Arab Emirates	

If your country is not listed here, your local IBM representative will be pleased to forward your comments to us. Or you can pay the postage and send the form direct to IBM (this includes mailing in the U.K.).

1
Cut along this line

2 Fold along this line

By air mail
Par avion



IBRS/CCRI NUMBER: PHQ - D/1348/SO



**REPONSE PAYEE
GRANDE-BRETAGNE**

IBM United Kingdom Laboratories
Information Development Department (MP095)
Hursley Park,
WINCHESTER, Hants
SO21 2ZZ United Kingdom

3 Fold along this line

From: Name _____
Company or Organization _____
Address _____

EMAIL _____
Telephone _____

1
Cut along this line

4 Fasten here with adhesive tape





Program Number: 5639-I90

Printed in the USA

SC34-5532-00

