LANDP® Family

Installation and Customization

Version 5.0

GC34-5530-00

LANDP® Family

Installation and Customization

Version 5.0

GC34-5530-00

Note

Before using this information and the product it supports, be sure to read the general information under Appendix H, “Notices” on page 545.

First Edition (April 2000)

This book is based on the previous edition, *LANDP Family Installation and Customization Version 4.0*, GC33-1961-00, 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-I90 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.

Contents

Figures

Tables

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)

The information provided in this book is intended to help you install the LANDP family, customize it to your own requirements, and make it operational. The book also contains information about how to customize PC/Integrator and PC Integrator/2.

Who should read this book

This book is written for system analysts, system programmers, and application programmers. *LANDP Introduction and Planning*, GC34-5529, is a prerequisite.

Individuals responsible for host computer systems and host computer applications can find useful information on how to include LANDP workstations in wide area networks.

What you need to know

You should be familiar with the operating systems that support your LANDP environment.

If you are involved in wide area communications, you should be familiar with System Network Architecture (SNA) protocols and Synchronous Data Link Control (SDLC), X.25[1] Data Link Control, or Token-Ring Data Link Control.

If you are involved in LANDP internal communications, you should be familiar with NetBIOS, or with the TCP/IP (Transmission Control Protocol/Internet Protocol) implementations of the systems where you are going to use the LANDP family.

[1] X.25 is a CCITT standard for packet switching networks.

Conventions used in this book

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

OS/2 DOS

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: DOS This example shows that the paragraph relates to LANDP for DOS only.

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

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 571.

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/

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.

Part 1. Introduction

This part introduces the installation and customization process.

It contains the following chapter:

Chapter 1. Introduction

OS/2 DOS 6000 NT

This book contains information enabling you to install the IBM LANDP® licensed programs family in your organization, to customize your LANDP workgroup so that it meets your specific needs, and to produce and distribute the workgroup information.

Planning your configurations and their hardware and software requirements must be done before you start the installation and customization process. This is covered in the *LANDP Introduction and Planning* book.

The complete installation and customization cycle includes the following steps:

1. Installing the LANDP family.

   The first step is to install the LANDP family on the workstations that will be used for customization purposes. These workstations are called *customization workstations*.

   Refer to Part 2, “Installation” on page 5 for further information.

2. Customizing the LANDP family.

   Customization is done by editing customization *vectors*. Vectors can be created from scratch (as described in Part 3, “Customization” on page 33), or by migrating from FBSS (the predecessor product to LANDP) or an earlier release of LANDP (as described in Chapter 16, “Migrating and generating customization data” on page 239).

   When all the vectors are defined according to your configuration plans, they contain all the operational and configurational characteristics for the workstations in the workgroup.

   If you want to add your own user servers, refer to Appendix B, “User servers” on page 275.

3. Making the LANDP family operational.

   After installing and customizing, the run-time files for all workstations are ready to be distributed.

   These files are located in the specified paths of the workstations that will be used to distribute the files to the individual workstations. These workstations are called *distribution workstations*.

   Refer to Part 4, “Distribution” on page 69 for further information.

Figure 1 on page 4 summarizes the installation, customization, and distribution process and shows where to find more information.

The remainder of the book shows how to prepare the various LANDP platforms for LANDP, utilities you can use as part of the installation and customization process, hints on performance tuning, and appendixes providing detailed customization information.

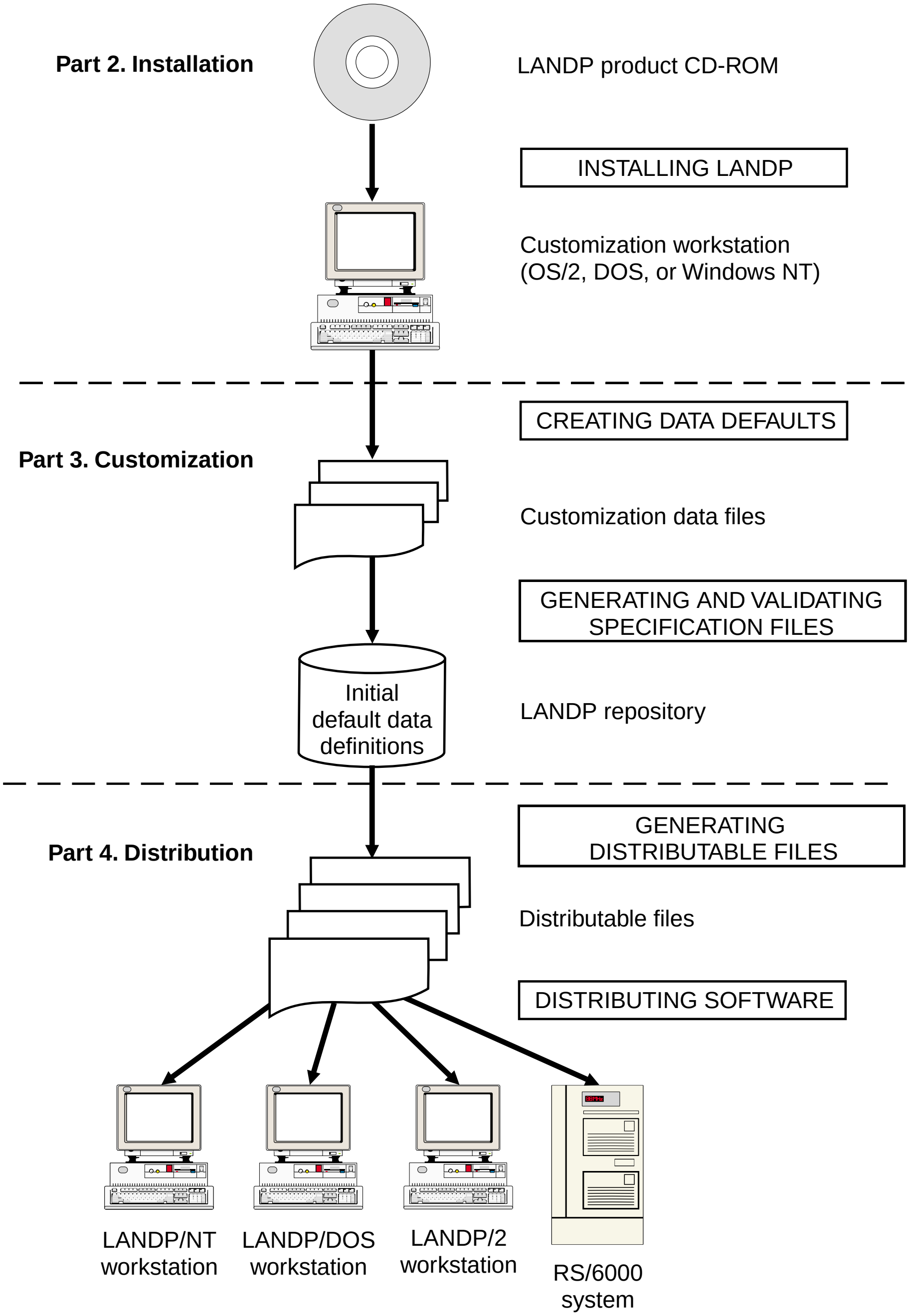


Figure 1. LANDP Installation, Customization, and Distribution Process

Part 2. Installation

This part describes how to install LANDP on your customization workstation.

It contains the following chapter:

Chapter 2. Installing the LANDP family

OS/2 DOS NT

The environment of your workgroup needs to be identified so that you can create configuration files that define the characteristics of the various workstations in your production sites.

This chapter describes how to install the LANDP family on the fixed disks of workstations that are used for this customization process.

These workstations are called *customization workstations*. You can customize from a workstation that has OS/2 or DOS installed on it. Alternatively, you may customize in an MS-DOS window running on Windows NT.

There are four stages required for installation:

- Planning for the installation
- Installing LANDP on the customization workstations
- Completing the installation
- Installing the FBSI Family and Banking Products (optional)

Planning for installation

OS/2 DOS NT

This section describes what you need to consider for your particular circumstances before beginning installation. It includes sections on the following:

- Location of the customization workstation
- Software requirements for the customization workstation
- Operating system requirements of the network
- The installation environment

You should read the *LANDP Introduction and Planning* book for detailed information about planning considerations for your own environment.

Note: You should read the READ.ME file on the LANDP product CD-ROM before starting the installation.

Location of the customization workstation

You should consider installing LANDP on:

- A unique customization workstation located in the development site for customizing different workgroups in the production site.

  This is useful if the production sites are unlikely to have LANDP skills.

- Customization workstations located in the production sites so that the customization process is performed for the workgroups in which the workstation is integrated.

This is useful if the workgroups often change, or are unique to the production sites.

Software requirements for the customization workstation

You can install LANDP for DOS, OS/2, or Windows NT and the LANDP online books on the customization workstation. If your network contains DOS, OS/2, and Windows NT workstations, you need to install all three components in the same location on the customization machine. The order does not matter. If your network includes workstations which are running FBSS, you'll need to install this too.

You can run the customization process on a DOS, OS/2, or Windows NT customization workstation. Your choice of customization workstation is independent of the mix of workstation types in the target workgroup.

Note: Under Windows NT, LANDP installation and customization uses a 16-bit implementation that can be run under a Windows NT command prompt.

If you choose to install onto a Windows NT workstation by using this method, you must add a DOSONLY statement to the CONFIG.NT file (usually found in the WINNT\SYSTEM32 directories). This ensures that the Windows NT DOS window is in the correct mode for the installation program.

Note: LANDP for DOS restricts you to a maximum of 120 alias definitions for each workstation in a workgroup. Aliases are defined with the CLIENT keyword of the LWSCONF vector; see Appendix D, "Editing configuration data" on page 407.

LANDP for OS/2 and Windows NT also have a maximum of 120. On LANDP for AIX the maximum is 5000 aliases per workstation.

To improve the performance of customization, you are recommended to install a disk cache program, for example, SMARTDRV.

If your customization machine is OS/2, and you plan to use the OS/2 Enhanced Editor for customization purposes, for example when using the EDITSPC command, you must install that option from the OS/2 installation diskettes. The EDITSPC command has some built-in macros to help with customization.

You should read the *LANDP Introduction and Planning* book for information about the necessary configurations for your own environment.

Checking for software prerequisites

You can install LANDP Version 5.0 on top of:

- LANDP Version 4
- LANDP/DOS Version 3
- LANDP/2 Version 3
- LANDP/DOS Version 2
- LANDP/2 Version 2
- FBSS (DOS)
- FBSS/2

You can also install:

- FBSS/2 Version 1.1
- PC/Integrator
- PC Integrator/2

on top of LANDP Version 5.0.

Notes:

1. PC/Integrator is the abbreviation for IBM Financial Branch System Integrator, which is the licensed program needed to use FBSS (DOS) workstations or LANDP for DOS workstations in conjunction with the IBM 4700 Finance Communication System.
2. PC Integrator/2 is the abbreviation for IBM Financial Branch System Integrator/2, which is the licensed program needed to use FBSS/2 workstations or LANDP for OS/2 workstations in conjunction with the IBM 4700 Finance Communication System.
3. For information on both PC/Integrator and PC Integrator/2, refer to *IBM Financial Branch System Integrator General Information,*.

If you plan to use the following products, you need to have the corresponding installation diskettes to copy the required files to the required paths.

- LAN Support program
- IBM PC X.25 Communication Support Version 1.11
- IBM PC X.25 Co-Processor/2 Support Program
- IBM Realtime Interface Co-Processor DOS Support
- IBM 4009 Universal Banking Printer
- IBM 4712 Transaction Printer, using serial ports
- IBM 4717 Magnetic Stripe Reader/Encoder
- IBM 4718 Personal Identification Number Keypad
- IBM 4721 Self Service Document Printer
- IBM 4722 Document Printer, using serial ports
- IBM 4731 Personal Banking Machine, Model Pxx
- IBM 4737 Self-Service Transaction Station
- IBM 4738 Personal Banking Machine, Model Pxx
- IBM 4739 Personal Banking Machine, Model Pxx
- IBM 4748 Document Printer
- IBM 4772 Universal Financial Printer
- IBM 4777 Magnetic Stripe Reader/Encoder
- IBM 4778 PIN Pad Magnetic Stripe Reader
- IBM 9055 Document Printer
- IBM 9068 Multi-Purpose Passbook Printer
- IBM 9069 Multi-Purpose Transaction Printer

For information on copying and customizing these files, refer to the manuals of the corresponding products. See “Bibliography” on page 571. See also “Installing FBSI family and 47xx banking products” on page 29.

For user-written servers you need to have the server program and all the files required by the server.

Operating system requirements of the network

This section includes information on:

- Storage requirements
- Requirements for installing LANDP for AIX

Storage requirements

The operating system in the workstation where LANDP is to be installed can be:

- IBM DOS Version. 7.1
- IBM OS/2 Warp, Version 4.0 or higher
- Microsoft Windows NT Version 4.0 or Windows 2000

When you install on a DOS workstation, a minimum of 512 KB (KB equals 1024 bytes) of free memory is required.

LANDP for DOS: When you install LANDP for DOS, the workstation must have at least 17 MB (MB equals 1,048,576 bytes) of free disk space.

LANDP for OS/2: When you install LANDP for OS/2, the workstation must have at least 20 MB of free disk space. An additional 2 MB is needed for Software Installer/2 files and data.

LANDP for Windows NT: When you install LANDP for Windows NT, the workstation must have at least 16 MB of free disk space.

If you install LANDP for DOS, OS/2, or Windows NT on top of another LANDP product that has modules common to LANDP for DOS, OS/2, or Windows NT, the workstation requires a lower amount of free disk space than the sum of the individual requirements.

Requirements for installing LANDP for AIX

The customization programs required for LANDP for AIX are included with LANDP for DOS and OS/2, so a separate installation for LANDP for AIX in the customisation workstations is not required.

LANDP for AIX is delivered on a 4 mm or 1/4 inch tape cartridge.

For further information about installing LANDP for AIX, refer to Chapter 14, “Preparing RS/6000 workstations” on page 203.

Installation environment

This section describes the installation environment required for installing LANDP under either DOS or OS/2.

Installation under DOS

When installing LANDP under DOS, the installation program checks that the following appear in the CONFIG.SYS file:

- A BUFFERS statement that has a parameter value of at least 20.
- A FILES statement that has a parameter value of at least 40.
- A SHELL statement that points to the COMMAND.COM file located in the drive and directory where DOS is installed on your fixed disk. Note that the E value on the SHELL statement should be set to /E:512 or greater.
- A DEVICE statement for the ANSI.SYS device driver.

Installation adds the following to the AUTOEXEC.BAT file:

- A SET COMSPEC statement that points to the COMMAND.COM file located in the drive and directory where DOS is installed on your fixed disk.

Installation under OS/2

When you install and run the customization program in an OS/2 workstation, a LANDP folder is created which contains the customization programs. Before installing LANDP, ensure that the LIBPATH parameter in CONFIG.SYS contains a period to indicate that the currrent directory is included in the search path, for example:

```
LIBPATH=.;C:\OS2\DLL;C:\OS2\MDOS\;C:\;
```

This period between semicolon path delimiters (`;.;`) indicates that the current directory is included in the search.

Installing LANDP on the customization workstations

OS/2 DOS NT

There are three ways to install LANDP:

1. From the installation CD-ROM. For information, refer to “Installing from CD-ROM” on page 12.

   This option is recommended if you plan to use a unique customization workstation that has a CD-ROM drive. You are prompted for responses during the install process.
2. From the diskette images. For information, refer to “Installing from diskette” on page 13.

   This option is recommended if you plan to use a unique customization workstation that does not have a CD-ROM drive. You are prompted for responses during the install process.
3. From a fixed disk on which the installation diskette images or CD-ROM have been copied.

   This method is called the Configuration, Installation, and Distribution (CID) method. For information, refer to “Installing from diskette” on page 13.

   This option is recommended if you plan to install LANDP on more than one customization workstation. The installation is performed non-interactively.

Installing from CD-ROM

There are two methods of installing from the CD-ROM. You can allow the system to guide you through the installation, allowing it to determine whether you are installing under DOS, OS/2, or Windows NT. Alternatively you can select the installation for the required operating system yourself.

The simplest method is to type INSTALL at the prompt for the CD-ROM drive. The install process will determine the operating system and install correctly for that system.

To use the alternative method, you need to run the correct installation program for the operating system of the customization workstation you are using. The DOS installation program is in the DOS subdirectory. The OS/2 installation program can be found in the OS2 subdirectory of the CD-ROM.

1. Insert the CD-ROM in your CD-ROM drive and type:

   ```
   d:\dos\install
   ```

   or

   ```
   d:\os2\install
   ```

   where d: should be replaced with the drive letter of your CD-ROM drive.

2. You will then be asked for some information. The exact format of the prompts you will be shown will differ, depending on which operating system you are using:

   - DOS

     You will be prompted for:

     a. Installation directory

        Type the name of the required target directory at the Installation directory prompt. The default is C:\EHC

     b. LANDP Components?

        ```
        Select any or all of:

        Install LANDP for OS/2 files         : YES
        Install LANDP for DOS files          : YES
        Install LANDP for Windows NT files   : YES
        Install LANDP books                  : YES
        ```

        Type Yes or No against each component; the default is Yes.

     c. Update CONFIG.SYS

        The default is YES. In addition to controlling whether the CONFIG.SYS is updated, this question also controls whether or not your AUTOEXEC.BAT is updated. If you choose not to update the CONFIG.SYS and AUTOEXEC.BAT files at install time (by typing No), the required changes are put into CONFIG.ADD and AUTOEXEC.ADD files in the C:\ directory. These changes need to be made and the machine rebooted before customization can be successfully performed.

Note: You can also install LANDP for DOS, OS/2, and Windows NT through the command prompt in Windows NT. If you install on Windows NT, follow the instructions for DOS.

- OS/2

  After selecting Continue from the initial installation panel, you will asked if you wish to update your CONFIG.SYS file and if you want to continue. If you choose not to update the CONFIG.SYS file at install time, the required changes are put into a CONFIG.ADD file in the root directory of your boot directory. These changes need to be made and the machine rebooted before customization can be successfully performed.

  Next you will asked to select the LANDP components you want to install and to select the installation directory. Select only the components necessary for the workstations that you will customize from this workstation:

  - LANDP_for_DOS
  - LANDP_for_OS/2
  - LANDP_for_Windows_NT
  - LANDP_books

  You must select all three products to customize DOS, OS/2, and Windows NT workstations from this workstation.

  Then enter the required target installation directory. The default installation directory is C:\EHC. You can use the Disk Space button to examine your drives and to select a different disk to be used.

- When installation is complete, if a message appears telling you to reboot your workstation, you should do so. This ensures that any changes made to your CONFIG.SYS and/or AUTOEXEC.BAT files will take effect.

Refer to “Online books” on page 30 if you select the LANDP_books component.

Installing from diskette

You will first need to build the diskettes from the diskette images on the CD-ROM. The diskette build program (LOADDSKF) can be found in the DISKETTE subdirectory of the CD-ROM.

To build a set of diskettes for installing on to a DOS customization workstation, you need to create diskettes from the following images:

Path name of the file **(Replace d: with the drive letter of your CD-ROM drive)**	**Description of the file**
d:\diskette\dos\ldp5ddos.dsk	LANDP V5 installation diskette for a DOS customization workstation

Path name of the file (Replace d: with the drive letter of your CD-ROM drive)	Description of the file
d:\diskette\dos\landp01.dsk . . . d:\diskette\dos\landp08.dsk	LANDP V5 data files (LANDP for DOS, OS/2, and Windows NT) for a DOS customization workstation

To build a set of diskettes for installing on to an OS/2 customization workstation, you need to create diskettes from the following images:

Path name of the file (Replace d: with the drive letter of your CD-ROM drive)	Description of the file
d:\diskette\os2\ldp52os2.dsk	LANDP V5 - for installation on an OS/2 customization workstation
d:\diskette\os2\landp01.dsk . . . d:\diskette\os2\landp08.dsk	LANDP V5 data files (LANDP for DOS, OS/2, and Windows NT) for an OS/2 customization workstation

Instructions for creating diskettes

You will need nine 2 MB diskettes.

Use the LOADDSKF program to build each diskette:

```
LoadDskF [D:][path]filename.dsk d: [options]
```

Options:

/F The target diskette will be formatted.

/Y Bypasses the user input (y/n) before overwriting the target diskette.

/Q Omits the warning message before overwriting the target diskette.

/S Silent Mode (don't Beep after writing the diskette).

This program works on DOS and OS/2.

Example

To build the diskette for the LANDP for DOS installation diskette for a DOS customization workstation, type:

```
d:\LOADDSKF d:\diskette\dos\ldp5ddos.dsk a:
```

replacing d: with the drive letter of your CD-ROM drive and a: with the drive letter of your diskette drive.

Select the correct diskette to reflect the operating system of the customization workstation that you are using. For example, you should use the installation diskette for OS/2 if your customization workstation is running OS/2.

When you have built the installation diskettes you can install LANDP onto machines without a CD-ROM drive, as follows:

1. Insert the installation diskette for the operating system into the diskette drive and when prompted type:

   ```
   a:\install
   ```

2. You will then be asked for some information. The exact format of the prompts you will be shown will differ depending on which operating system you are using (DOS or OS/2).

   - DOS

     You will be prompted for:

     a. Installation directory

        Type the name of the required directory at the prompt. The default is C:\EHC

     b. Update CONFIG.SYS

        The default is YES. In addition to controlling whether or not the CONFIG.SYS is updated, this question also controls whether or not your AUTOEXEC.BAT is updated. If you choose not to update the CONFIG.SYS and AUTOEXEC.BAT files at install time (by typing No), the required changes are put into CONFIG.ADD and AUTOEXEC.ADD files in the C:\ directory. These changes need to be made and the machine rebooted before customization can be successfully performed.

- OS/2

  After selecting Continue from the initial installation panel, you will asked if you wish to update your CONFIG.SYS file and if you wish to continue. If you choose not to update the CONFIG.SYS file at install time, the required changes are put into a CONFIG.ADD file in the root directory of your boot directory. These changes need to be made and the machine rebooted before customization can be successfully performed.

  Next you will asked to select the LANDP components you wish to install and to select the installation directory. You should select the single component shown and enter the required target directory. The default installation directory is C:\EHC. You can use the Disk Space button to examine your drives and to select a different disk to be used.

  You will then be prompted to insert a new diskette. Follow the prompts for each subsequent diskette.

  When installation is complete, if a message appears telling you to reboot your workstation, you should do so. This ensures that any changes made to your CONFIG.SYS and/or AUTOEXEC.BAT files will take effect.

  Note: The installation program will ask only for those diskettes containing files for the requested component. In general, this will be one or more of the common diskettes and the specific component diskettes.

Installing from a fixed disk

Installation from a fixed disk can be automated using the Configuration, Installation, and Distribution (CID) method. This method is available when installing onto an OS/2 customization machine only.

This section describes how to create response file and command line parameters to install LANDP non-interactively from a fixed disk. A template response file and command file are provided on the CD-ROM and on the installation diskettes for OS/2. Their names are:

- ehc2500.rsp (template response file)
- ehc2500.cmd (template command file)

To use the CID method:

1. Copy the contents of the CD-ROM to a LAN location that the target machines have read/execute access to.

   For example, you could enter:

   ```
   XCOPY d:\* x:\landp\os2 /s
   ```

   This copies all the files from the CD-ROM in the d: drive to the target directory named `landp\os2` on the x: drive.

   Note: The /s parameter is used to ensure all the sub-directories are copied and is mandatory.

2. Edit the template response file to match your system setup. The response file contains the following keyword parameters:

Parameter	**Default**
FILE	C:\EHC
CFGUPDATE	AUTO
OVERWRITE	NO
COMP	LANDP_for_DOS
COMP	LANDP_for_OS/2
COMP	LANDP_for_Windows_NT
COMP	LANDP_books

where:

FILE Specifies the target directory in which the product will be installed.

CFGUPDATE Specifies whether the CONFIG.SYS file will be updated.

To update your CONFIG.SYS file automatically, enter:

```
CFGUPDATE=AUTO
```

To place the necessary updates in a file called CONFIG.ADD in the root directory of your boot drive, enter:

```
CFGUPDATE=MANUAL
```

The updates must be added to your CONFIG.SYS file before running LANDP customization.

OVERWRITE Specifies whether you want to overwrite files in the install directory. To overwrite files if they already exist, enter:

```
OVERWRITE=YES
```

To keep the original files, enter:

```
OVERWRITE=NO
```

COMP Specifies a LANDP component to be installed. It can be specified multiple times to install more than one component.

To install LANDP for DOS, enter:

```
COMP=LANDP_for_DOS
```

To install LANDP for OS/2, enter:

```
COMP=LANDP_for_OS/2
```

To install LANDP for Windows NT, enter:

```
COMP=LANDP_for_Windows_NT
```

To install the LANDP books, enter:

```
COMP=LANDP_books
```

3. Edit the template command file to match your system setup.

   The template command file is used to install LANDP. A sample command file follows:

   ```
   x:\landp\os2\install /A:I
          /L1:c:\ehc\error.log
          /L2:c:\ehc\history.log
          /O:DRIVE
          /R:x:\landp\os2\ehc2500.rsp
          /S:x:\landp\os2
          /C:x:\landp\os2\ehc2500.icf
          /P:LANDP_Family
          /X
   ```

   In the example:

   /A:I Specifies the type of operation (in this case, INSTALL). Another possibility is:

   ```
   /A:D
   ```

   which deletes all the LANDP files you have installed.

   /L1 Specifies the fully qualified name of the file to be used to store any errors encountered during the operation.

   In the example, the file error.log in the ehc directory of the c: drive will be used.

   /L2 Specifies the fully qualified name of the file to be used to store any messages send by the operation.

   In the example, the file history.log in the ehc directory of the c: drive will be used.

   /O:DRIVE Specifies that the product is being installed from a workstation or LAN drive.

   /R Specifies the fully qualified name of the response file.

   In the example, the response file is in the landp directory on the x: drive.

   /S Specifies the fully qualified location of the distribution copy of the LANDP product.

   In the example, the product file is in the landp\os2 directory on the x: drive.

   /C Specifies the fully qualified file name of the product catalog file. The catalog file name is ehc2500.icf.

   /P Specifies the product to be installed.

   /X Specifies a non-interactive operation. This parameter is mandatory, and is the last one specified.

4. Run the command file to install LANDP.

After the process completes, check the error log to ensure that the operation was successful and to find out if you need to shut down and restart the workstation. You may also have a software distribution manager that executes this command for you, for example, Tivoli NetView® DM on OS/2.

Deleting LANDP

Under OS/2, you delete LANDP for DOS, OS/2, or Windows NT by double-clicking on the **Delete LANDP for DOS**, **Delete LANDP for OS/2**, or **Delete LANDP for Windows NT** icon in the LANDP folder.

You can delete any one or all three products. In either case, any files that are shared between the three products are not deleted. So if you delete all three products, the common files and LANDP folder are not deleted; you must delete them manually.

Deleting using the CID method

You can delete from a fixed disk using the Configuration, Installation, and Distribution (CID) method. In order to do so, you need a delete response file (for example, ehc2500.rsp), as follows:

Parameter	**Default**
FILE	C:\EHC
CFGUPDATE	AUTO
OVERWRITE	NO
COMP	LANDP_for_DOS
COMP	LANDP_for_OS/2
COMP	LANDP_for_Windows_NT
COMP	LANDP_books
DELETEBACKUP	NO

The following file shows the required CID parameters:

```
c:\ehc\ehcinsts /A:D
        /L1:c:\ehc\error_log
        /L2:c:\ehc\history_log
        /O:DRIVE
        /C:c:\ehc\ehc2500.icf
        /P:LANDP_Family
        /R:c:\ehc\ehc2500.rsp
        /X
```

In this example:

`/A:D` Specifies the type of operation (in this case, DELETE).

`/L1` Is an optional parameter that specifies the fully qualified name of the file to be used to store any errors encountered during the delete operation.

In the example, the name used is `error_log` on the `ehc` directory on the `c:` drive.

/L2 Is an optional parameter that specifies the fully qualified name of the file to be used to store any messages sent by the delete operation.

In the example, the name used is `history_log` on the `ehc` directory on the `c:` drive.

/O:DRIVE
Specifies that the product is being deleted from a workstation environment.

/C Specifies the fully qualified file name of the product's catalog file. The following table lists the catalog file associated with each LANDP product.

Product	**Catalog file**
LANDP_Family	ehc2500.icf

In the example, the product was installed on the `c:\ehc` directory.

/P Identifies the product to be deleted (in this case, LANDP_Family).

/R Specifies the drive name and path for the delete response file.

/X Specifies a non-interactive operation. This parameter is mandatory, and is the last one specified.

Completing the LANDP for DOS installation

After you finished installing LANDP for DOS, you may need some of the information provided in this section if you want to use any of the following facilities:

- Magnetic stripe reader/encoder server
- PIN pad server
- Financial printer server
- 4748 printer server
- User servers
- 4733 teller assist unit
- NetBIOS support
- X.25 Data Link control

The EHCD500 directory mentioned in this section is located in the path where LANDP is being installed.

Note: Please ensure that you have the latest set of device driver diskettes, containing the files described in the rest of this section. Otherwise, the new functions may not be supported.

To get the latest LANDP device drivers, please contact your IBM representative who will send you the ones that are listed in the LANDP V5 announcement letter.

Magnetic stripe reader/encoder server

If the server supports a 4717 MSR/E, copy the MSRE2DD.SYS device driver from the 4700 Device Drivers diskette to the EHCD500 directory.

If the server supports a 4777 MSR/E or a 4778 MSR:

- If COM port attached, copy the IBM4777.SYS device driver to the EHCD500 directory
- If mouse attached, copy the MSRE2DD.SYS device driver to the EHCD500 directory.

PIN pad server

If the server supports a 4718 PIN pad, copy the PIN2DD.SYS device driver from the 4700 Device Drivers diskette to the EHCD500 directory.

If the server supports a 4778 PIN pad:

- If COM port attached, copy the IBM4778P.SYS device driver to the EHCD500 directory
- If mouse attached, copy the PIN2DD.SYS device driver to the EHCD500 directory.

If 4778 MSR capabilities are used, copy the IBM4777.SYS device driver to the EHCD500 directory.

Financial printer server

If only the parallel port is used, no device driver is required.

If serial ports are used and either 4712 printers or 4722 printers are supported, you must customize the printer device drivers by using the device driver customization program supplied with the 4700 Device Drivers diskette.

Version 3.0 (or later) of the device drivers is required to support the 4722 model 3 printer with the Read and Encode Magnetic Stripe (REMS) facility.

The same device drivers as those for the 4722 printer are used to support the 9055 model 2 and 9068-S01 printers.

The device driver customization program prompts you to enter a driver file name and a driver device name. Refer to Table 1 on page 22 and assign driver file names depending on the port number and model number, as well as whether the REMS facility is present. Driver device names depend only on the port number to which the printer will be connected.

Table 1. 4712 and 4722 Printer Device Driver Customization

Model	Port	Driver File Name	Driver Device Name
4712	1	FPRT121.SYS	FPRTDD1
4722	1	FPRT221.SYS	FPRTDD1
4722 REMS	1	FPRT221R.SYS	FPRTDD1
4712	2	FPRT122.SYS	FPRTDD2
4722	2	FPRT222.SYS	FPRTDD2
4722 REMS	2	FPRT222R.SYS	FPRTDD2
4712	3	FPRT123.SYS	FPRTDD3
4722	3	FPRT223.SYS	FPRTDD3
4722 REMS	3	FPRT223R.SYS	FPRTDD3
4712	4	FPRT124.SYS	FPRTDD4
4722	4	FPRT224.SYS	FPRTDD4
4722 REMS	4	FPRT224R.SYS	FPRTDD4

If 4772 printers are supported, the required device driver is:

4772SS.SYS

The customized device drivers must be copied to the EHCD500 directory.

4748 printer server

You must customize the 4748 printer device drivers by using the device driver customization program supplied with the 4748 printer device drivers diskette.

The 4748 Printer Server also provides support for the 9055 Model 1 printer, which supersedes the 4748 printer and can be set to emulate the 4748. (The 9055 Model 2 printer is supported by the 47X2 server.) The 9055 Printer provides more function, such as Reader/Encoder Magnetic Stripe (REMS) support, but needs its own device driver if the extra function is to be used under DOS.

Whichever printer you are using, you should copy its device driver files from the diskette supplied with it to the EHCD500 directory as shown below.

Table 2. 4748 Printer Server Device Driver Files under DOS

Printer	4748 emulation mode? (No REMS)	Driver File Name
4748	Not relevant	FPRT48x.SYS
9055 Model 1	Yes	FPRTSCPx.SYS or FPRT48x.SYS
9055 Model 1	No	FPRTSCPx.SYS
9068-S01	No	FPRTSCPx.SYS
Note: x is the port (1 or 2) to which the printer is attached.		

The device driver customization program prompts you to enter a driver file name and a driver device name. Refer to Table 3 on page 23 and assign driver file names depending on the port number. Driver device names depend only on the port number to which the printer will be connected.

Table 3. 4748 Printer Device Driver Customization

Model	**Port**	**Driver File Name**	**Driver Device Name**
4748	1	FPRT481.SYS	FPRTDD1
4748	2	FPRT482.SYS	FPRTDD2
9055 Model 1	1	FPRTSCP1.SYS	FPRTDD1
9055 Model 1	2	FPRTSCP2.SYS	FPRTDD2
9068-D01	1	FPRTSCP1.SYS	FPRTDD1
9068-D01	2	FPRTSCP2.SYS	FPRTDD2

User servers

If you have developed a server, the server program, the device driver it may require, and other needed files, should be copied to the directory specified in the SUBDIR keyword of the DEFSERV vector and, if it applies, of the DISTRIB vector. See Appendix B, “User servers” on page 275.

If you have developed a server that runs under LANDP for DOS, OS/2, and Windows NT, you must copy the files to the directories specified for LANDP for DOS, OS/2, and Windows NT. In this case, LANDP for DOS, OS/2, and Windows NT must already be installed.

4733 teller assist unit

If you are using a 4733 teller assist unit, you need to copy the TCD386D.SYS device driver to the EHCD500 directory.

NetBIOS support

If you select NetBIOS as the transport protocol, LANDP customization assumes that you are using the IBM LAN Support Program for a non-Network Driver Interface Specifications (NDIS) device driver.

Copy the following device drivers from the LAN Support Program installation diskette to the EHCD500 directory:

DXMA0MOD.SYS
DXMC0MOD.SYS
DXMT0MOD.SYS
DXMG0MOD.SYS

If the LAN Support Program version that you are using provides the message files DXMMSG.xxx, copy also the appropriate file as DXM.MSG.

If you are using IBM LAN Client, or if there are programs other than the LANDP NetBIOS transport using NetBIOS, you must install and configure the NetBIOS directly (see “Installation requirements for NetBIOS transport protocol” on page 95).

X.25 Data Link control

If you are using **IBM PC X.25 Communications Adapter**, copy the following files from the installation diskette of IBM PC X.25 Communication Support Version 1.11 to the EHCD500 directory:

X251.SYS
X25MENA.TXT
X25MENB.TXT
X25MENC.TXT
XL.EXE
CX1.LDA
CX2.LDA
CX3.LDA
CX4.LDA
CX5.LDA
CX6.LDA
CX7.LDA
CX8.LDA

If you are using **IBM X.25 Co-Processor/2**, create the FBSSC2X directory in the installation path.

Copy the following file from the IBM X.25 Co-Processor/2 option diskette to the FBSSC2X directory:

ICAAIM.COM

Copy the following files from the installation diskette of IBM X.25 Co-Processor/2 Support Program to the FBSSC2X directory:

GQKENGLI.TXT
GQKPTCOL.DEF
GQKCONF.EXE
GQKINIT.EXE
GQKLOAD.EXE
GQKSL1.EXE
GQKPTCOL.EXE
X25.EXE

Copy the following files from the installation diskette of IBM Realtime Interface Co-Processor DOS Support to the FBSSC2X directory:

ICAINTH.COM
ICAINTH.MSG
ICAINTH.SYS
ICALOAD.COM
ICALOAD.MSG

Finally, you must customize the IBM X.25 Co-Processor/2 in order to create the profiles to be used. For information about this process, refer to the corresponding manuals.

You have now finished the installation process for LANDP for DOS.

Completing the LANDP for OS/2 installation

After you finished installing LANDP for OS/2, you may need some of the information provided in this section if you want to use any of the following facilities:

- Magnetic stripe reader/encoder server
- PIN pad server
- Financial printer server
- 4748 printer server
- User servers
- 4733 teller assist unit

The EHCO500 directory mentioned in this section is located in the path where LANDP is being installed.

Note: Please ensure that you have the latest set of device driver diskettes, containing the .DLL files described in the rest of this section. If you have older .DLL files, the new functions may not be supported.

To get the latest LANDP device drivers, please contact your IBM representative who will send you the ones that are listed in the LANDP V5 announcement letter.

Magnetic stripe reader/encoder server

Copy the following files to the EHCO500 directory:

MAGCALLS.DLL

FIO*xxx*.MSG and FIO*xxx*H.MSG message files, renamed to FIO.MSG and FIOH.MSG files. Note that *xxx* stands for the language identifier.

If the server supports a 4717 MSR/E, or is mouse attached, copy the FIOAUXDD.SYS file from the 4700 OS/2 Device Drivers diskette to the EHCO500 directory.

If the server supports a 4777 MSR/E or a 4778 MSR, copy the FIOSERDD.SYS file to the EHCO500 directory.

PIN pad server

Copy the following files to the EHCO500 directory:

PINCALLS.DLL

FIO*xxx*.MSG and FIO*xxx*H.MSG message files, renamed to FIO.MSG and FIOH.MSG files. Note that *xxx* stands for the language identifier.

If the server supports a 4718 PIN pad, and a magnetic stripe reader/encoder server will not be loaded in the workstation, copy the FIOAUXDD.SYS file from the 4700 OS/2 Device Drivers diskette to the EHCO500 directory.

If the server supports a 4778 PIN pad:

- If COM port attached, copy the FIOSERDD.SYS file to the EHCO500 directory
- If mouse attached, copy the FIOAUXDD.SYS file to the EHCO500 directory.

If the server supports a 4778 PIN pad, and a magnetic stripe reader/encoder server will be loaded in the workstation, copy the PINMSR.DLL file to the EHCO500 directory.

Financial printer server

If only the parallel port is used, no device driver is required.

If serial ports are used and 4009, 4712, 4722, 4772, 9055-002, or 9068-S01 printers are supported, copy the following files from the 4700 OS/2 Device Drivers diskette to the EHCO500 directory:

PRTCALLS.DLL
FIOPRT.DCP
P4712.PCH
P4722.PCH
4772PDD.SYS
4772S.DCP
P4722-3.PCH
R4722-3.PCH
P4722-OP.PCH
P4722-1.PCH
R4722-1.PCH

You also have to copy the FIO*xxx*.MSG and FIO*xxx*H.MSG message files, renamed to FIO.MSG and FIOH.MSG files. Note that *xxx* stands for the language identifier; select the files according to the language of your choice.

4748 printer server

Copy the COM.SYS file to the EHCO500 directory. The COM.SYS file is provided with OS/2. For IBM PS/55 Models 5560 and 5580, copy the COMDMA.SYS file.

The 4748 Printer Server also provides support for the 9055 Model 1 printer, which supersedes the 4748 printer and can be set to emulate the 4748. (The 9055 Model 2 printer is supported by the 47X2 server.) The 9055 Printer provides more function, such as Reader/Encoder Magnetic Stripe (REMS) support, but needs its own device driver if the extra function is to be used under OS/2.

Whichever printer you are using, you should copy its dynamic link library files from the diskette supplied with it to the EHCO500 directory as shown below.

Table 4. 4748 Printer Server Device Driver Files under OS/2

Printer	4748 emulation mode? (No REMS)	DLL
4748	Not relevant	PBPCALLS.DLL
9055 Model 1	Yes	SCPCALLS.DLL or PBPCALLS.DLL
9055 Model 1	No	SCPCALLS.DLL
9068-D01	Yes	SCPCALLS.DLL or PBPCALLS.DLL
9068-D01	No	SCPCALLS.DLL

User servers

If you have developed a server, the server program, the device driver it may require, and other needed files, should be copied to the directory specified in the SUBDIR keyword of the DEFSERV vector and, if it applies, of the DISTRIB vector. See Appendix B, "User servers" on page 275.

If you have developed a server that runs under LANDP for DOS, OS/2, and Windows NT, you must copy the files to the directories specified for LANDP for DOS, OS/2, and Windows NT. In this case, LANDP for DOS, OS/2, and Windows NT must already be installed.

4733 teller assist unit

If you are using a 4733 teller assist unit, you need to copy the TCD3862.SYS device driver to the EHCO500 directory.

You have now finished the installation process for LANDP for OS/2.

Completing the LANDP for Windows NT installation

After you finish installing LANDP for Windows NT, you may need some of the information provided in this section if you want to use any of the following facilities:

- Magnetic stripe reader/encoder server
- PIN pad server
- Financial printer server
- User servers

The EHCN500 directory mentioned in this section is located in the path where LANDP is being installed.

Note: Ensure that you have the latest set of device driver diskettes, containing the .DLL files and other files described in the rest of this section. If you have older .DLL files, the new functions may not be supported.

To get the latest LANDP device drivers, please contact your IBM representative who will send you the ones that are listed in the LANDP V5 announcement letter.

Magnetic stripe reader/encoder server

Copy the following files to the EHCN500 directory:

- WNT77MAG.DLL
- WNT7XPDD.DLL

PIN pad server

Copy the following files to the EHCN500 directory:

- WNT78PIN.DLL
- WNT78MSR.DLL
- WNT7XPDD.DLL

Financial printer server

If only the parallel port is used, no device driver is required.

If serial ports are used and 4712, 4722, 4772, 9055-002, 9069 or 9068-S01 printers are supported, copy the following files from the Device Drivers diskette to the EHCN500 directory:

- WNT47X2.DLL
- WNTSFPRT.DLL
- P4712.PCH
- P4722.PCH
- P4722-1.PCH
- P4722-3.PCH
- P4722-OP.PCH

- R4772-1.PCH
- R4772-3.PCH
- 4772S.DCP
- FIOPRT.DCP

4748 printer server

Copy the file `WNTDFPRT.DLL` to the EHCN500 directory.

Under Windows NT, the 4748 printer server does not support the 4748 printer itself, but supports a 9055-01 or 9068-D01 printer when its ID byte is set to 1E(hexadecimal).

User servers

If you have developed a server, the server program, the device driver it may require, and other needed files, should be copied to the directory specified in the SUBDIR keyword of the DEFSERV vector and, if it applies, of the DISTRIB vector. See Appendix B, “User servers” on page 275.

If you have developed a server that runs under LANDP for DOS, LANDP for OS/2, and LANDP for Windows NT, you have to copy the files to the directories specified for LANDP for DOS, OS/2, and Windows NT. In this case, LANDP for DOS, OS/2, and Windows NT must already be installed.

You have now finished the installation process for LANDP for Windows NT.

Note: If you have a problem when loading the required dynamic library EHCWINNT.DLL, refer to *LANDP Problem Determination*, Chapter 8.

Considerations if the installation path is changed

If the LANDP installation path is changed after installation has been completed and customization run (for example, by XCOPYing the entire LANDP directory structure to a different path), then subsequent operation of the GENRUN utility will produce GETTING.SPC files referring to the original installation path and not the new one.

This problem can be overcome by deleting the file EHCPATH.SYS from the EHCCUS directory. This file will then be recreated with the correct path statement next time customization is run.

Installing FBSI family and 47xx banking products

After installing LANDP, you can install the following products using the procedures provided for each of them.

If you installed LANDP to remote fixed disks, run the procedures making sure that you use the same drive and path as for LANDP installation.

PC/Integrator and PC Integrator/2

When LANDP has been installed:

1. Insert the first PC/Integrator or PC Integrator/2 diskette (labeled Diskette 1) in the diskette drive.
2. From the EHCCUS directory, created by the installation program, enter:

   ```
   INSTFBSI
   ```

   Note: The INSTFBSI procedure can be run only in a DOS environment.

Complete the installation process using the PC/Integrator or PC Integrator/2 manuals. See "Bibliography" on page 571.

To use BIWP in OS/2 MVDM, copy the BIWP.EXE file distributed with PC/Integrator to the EHCO500 directory, renamed to VBIWP.EXE file.

47XX Banking Products

To install the software required for:

- IBM 4737 Self-Service Transaction Station
- IBM 4731 Personal Banking Machine, Model Pxx
- IBM 4738 Personal Banking Machine, Model Pxx
- IBM 4739 Personal Banking Machine, Model Pxx

refer to the corresponding manuals. See "Bibliography" on page 571.

To start the customization process for the banking machines software, from the EHCCUS directory enter:

```
FBSSBDV
```

Note: The FBSSBDV procedure can be run only in a DOS environment.

Complete the customization process by following the information in the Personal Banking Machines manuals.

Online books

The LANDP V5 books are on the product CD-ROM in the following formats:

- .INF files, for viewing under OS/2
- .PDF files, for viewing and printing under Adobe Acrobat.

  The Adobe Acrobat Reader is available on Windows (3.1 and NT), DOS, and OS/2 (plus many other platforms!). Get a free copy of the Adobe Acrobat Reader for your platform from the Internet by following the details on Adobe's home page:

  ```
  http://www.adobe.com
  ```

In addition, the LANDP CD-ROM also contains a file called GG243842.BOO. The file is an online version of the *FBSS Problem Determination Guidelines* "Red Book", GG24-3842, that was produced by the IBM International Technical Support Center (ITSC). The first part of this book has a description and examples of FBSS traces; the information is equally relevant to traces for LANDP for DOS and OS/2.

Use one of the IBM BookManager® READ products to view this file.

Put the online book files on a shared drive or distribute them to the machine of each user who will need them. Don't forget to tell your users that the online books are available for them to use.

BookManager BOOK files

BookManager BOOK files of the LANDP V5 books are provided on the *Transaction Processing and Data* collection kit CD-ROM (order number SK2T-0730).

Part 3. Customization

This part describes how to customize your LANDP installation to satisfy the requirements of your organization.

It contains the following chapters:

Chapter 3. The customization process

OS/2 DOS NT

After you have installed LANDP for DOS, OS/2, or Windows NT, you design a data processing network to meet your organization's requirements. This is called *customizing*.

LANDP customization files are automatically installed when you install LANDP for DOS, OS/2, or Windows NT.

The complete customization process involves the following steps:

1. Identifying the requirements for your environment. This is part of the planning process, and is described in *LANDP Introduction and Planning*.
2. Running the customization utility programs to define the characteristics of the various workstations in your production sites. These programs are covered in more detail in "The customization steps" on page 40.
3. Using the customization utility to create the diskettes for each workstation (obtaining software for distribution). This is described in Chapter 8, "Obtaining software for distribution—GENRUN" on page 71.

If you have customization data generated earlier by the FBSS customization program, you can use the *migration* utility. This utility will migrate all the previously defined data to your LANDP workgroup and adds the information to the internal repository. This process is described in "Migrating customization data from FBSS" on page 43 and in Chapter 16, "Migrating and generating customization data" on page 239.

You can see and perform the customization steps by following the examples described in Appendix A, "A hands-on example" on page 267. These examples are also provided on the product CD-ROM and may be useful if you want to get a hands-on feeling for the customization procedures.

Notes:

1. LANDP for Windows NT users can customize in any of the following ways:
   - On DOS
   - On OS/2 , running in an OS/2 window (but not in a DOS window)
   - On Windows NT using LANDP's DOS customization utilities running in an MS-DOS window To do this, modify the `files=` statement in CONFIG.NT to be at least 30. CONFIG.NT is in the SYSTEM32 subdirectory of the directory in which you installed Windows NT.

     You can start your MS-DOS window using the Windows NT command prompt `cmd.exe`. Alternatively, you can start your MS-DOS window using `command.com /E:2048`, with CONFIG.NT specifying DOSONLY.
2. When running LANDP customisation and related utilities on Windows NT, your customisation workstation must have an A: diskette drive, otherwise unavoidable system delays can occur. It is not necessary to have a diskette inserted. This

warning applies particularly to laptop computers, which may be shipped without an A: drive.

LANDP customization - things you need to know

Before starting the customization process, it is necessary to understand some fundamental concepts involved in the process.

Customization data structure

The customization data is specified in *vector* format. A vector is a set of statements defining configuration items. You define a series of vectors that are stored in files. These files are processed to generate an internal repository for retrieval purposes.

You define and process the customization data in different vector groups. Each vector group is included in a separate LANDP specification (SPC) file:

COMMON.SPC	Common data vectors
LANCONF.SPC	Workgroup configuration data vectors
MODELS.SPC	Model configuration data vectors

Common data—COMMON.SPC

Common data vectors are common for all or for several workstations or workgroups. These include defaults, tables, record definitions, and profiles.

The vectors that contain common data must be grouped in the COMMON.SPC file located in the EHCCUS directory, created by the installation program.

Workgroup configuration data—LANCONF.SPC

Workgroup configuration data vectors are specific for each workgroup.

The vectors that contain workgroup configuration data must be grouped in a LANCONF.SPC file located in a directory created for the workgroup. For more information see “Customization directory structure” on page 37.

Model configuration data—MODELS.SPC

The objective of modeling is to avoid specifying customization data more than once, when defining similar workgroup configurations.

A model configuration is a pattern designed *only* for reference purposes. Once you have defined a model configuration, you can use that model in a workgroup configuration just by referencing its name.

Models can be defined on three levels:

Workgroup level

A workgroup model configuration serves as a pattern for a workgroup configuration. You can make modifications to the same model in each workgroup configuration that refers to it.

Thus you specify the more general information only once in the model, and make workgroup specific changes only in the respective workgroup configuration definitions.

Workstation level

A workstation model configuration serves as a pattern for a workstation configuration. It can be referenced from either a workgroup model or a workgroup configuration, and contains information that is similar in many workstations in any workgroup.

Specific changes to a workstation model are made on the workgroup level where this model is referenced.

Server level

A server model configuration serves as a pattern for workstations which use similar server configurations. It can be referenced from either a workstation model or a workstation configuration.

Another useful way of using models is *nesting models*.

- Within a workgroup model configuration, you can refer to a workstation model when specifying the configurations of similar workstations in the workgroup. You can also reference a server model.
- On workstation level, you can refer to a particular server model either from a workstation configuration, or from a workstation model.

The vectors that contain model configuration data must be grouped in the MODELS.SPC file located in the EHCCUS directory, created by the installation program.

Customization directory structure

To locate LANDP information, the installation program creates a set of directories.

To locate customization information, you have to create some directories at workgroup level. The customization directory structure must be located in the first level subdirectory you define during installation.

The following figure shows the customization directory structure. The .SPC files are also included.

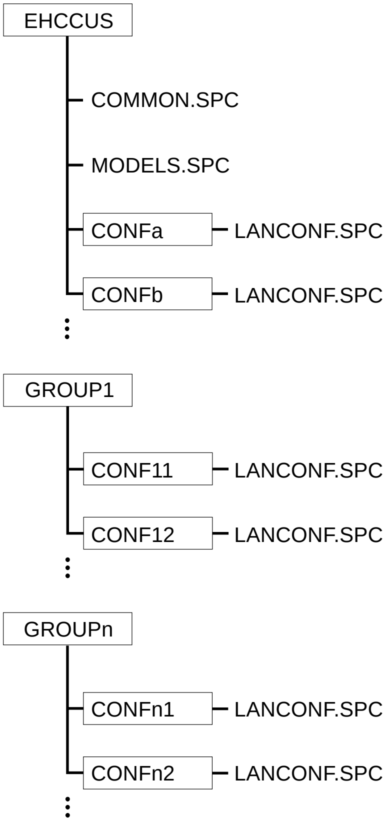


Figure 2. Customization Directory Structure

The COMMON.SPC and MODELS.SPC files must be located in the EHCCUS directory created by the installation program.

To locate the LANCONF.SPC files, containing workgroup configuration data, create a directory for each file.

These directories can be located either in the EHCCUS directory or in any directory you create at the same level as the EHCCUS directory. You can also create a directory for each group of workgroup configurations to be defined.

If you use the EDITSPC procedure provided by the customization program to edit .SPC files, you do not need to create those directories. The procedure creates them for you.

Techniques and considerations

Here are some more points you will need to know before you start to customize LANDP.

Remote customization

Remote customization, the process of customizing from a machine other than a designated customization workstation (for example via a LAN server), is supported. However, customization from two remote workstations at the same time is not allowed. If you try to customize, and customization is already taking place, an error message is displayed on the screen.

Notes:

1. You cannot run the customization program under OS/2 while FBSS/2 or LANDP for OS/2 are running in the same workstation.
2. The migration procedure in an OS/2 customization workstation will always run in MVDM, because it can only be run in DOS.

Embedded specification files

The objective of embedding files is to avoid working with large .SPC files.

Instead of storing all the common or model configuration data in the COMMON.SPC or MODELS.SPC file, some data can be stored in separate files. Later, these files can be embedded in the COMMON.SPC file or in the MODELS.SPC file, using INCLUDE statements.

The format of the INCLUDE statements is:

```
INCLUDE=D:\xxxxxxxx\yyyyyyyy\nnnnnnnn.eee
```

where

D:\xxxxxxxx\yyyyyyyy	is the path where the file is located.
nnnnnnnn	is the filename.
eee	is the extension of the separate file.

The number of directory levels is only limited by the operating system.

The INCLUDE keyword must start in the first position. No blanks are allowed between the INCLUDE keyword, the = sign, and the full file path.

When you retrieve customization data from the internal repository, only a single COMMON.SPC file and a single MODELS.SPC file are created, regardless of whether you embedded files to define the customization data.

Partial specification files

The objective of creating partial .SPC files is to avoid processing the whole COMMON.SPC or MODELS.SPC files to update the internal repository.

When processing a COMMON.SPC file or MODELS.SPC file, the common data or model configuration data in the internal repository is not deleted. The process result is that some data may remain as it is, some data may change according to the data in the .SPC file, or some data may be added to the repository.

This feature enables you to use a new profile (common data) without processing the whole COMMON.SPC file, or to use a new model of workstation (model configuration data) without processing the whole MODELS.SPC file.

The customization steps

Figure 3 on page 41 shows an overview of the customization steps.

The boxes in 3 represent the procedures you will be running during customization. Depending on your work environment, you will have to run all or just some of them.

The following sections describe, briefly, the start state for each procedure, the task being performed, and where to find more information.

Creating common data default definitions—CREATE

Start state
The LANDP installation is complete.

Task
To create common data default definitions, which are stored in the internal LANDP repository. If a COMMON.SPC file already exists, the CREATE procedure works like the VALSPEC procedure.

For more information
See Chapter 4, “Creating common data default definitions—CREATE” on page 45.

Converting common data to editable .SPC files—GENSPEC

Start state
There are two possible start states for running the GENSPEC procedure:

- The CREATE procedure has just been run
- The .SPC files have been edited and validated

Task
To convert the common, workgroup, and model data located in the internal repository to editable .SPC files.

For more information
See Chapter 5, “Converting common data to editable .SPC files—GENSPEC” on page 47.

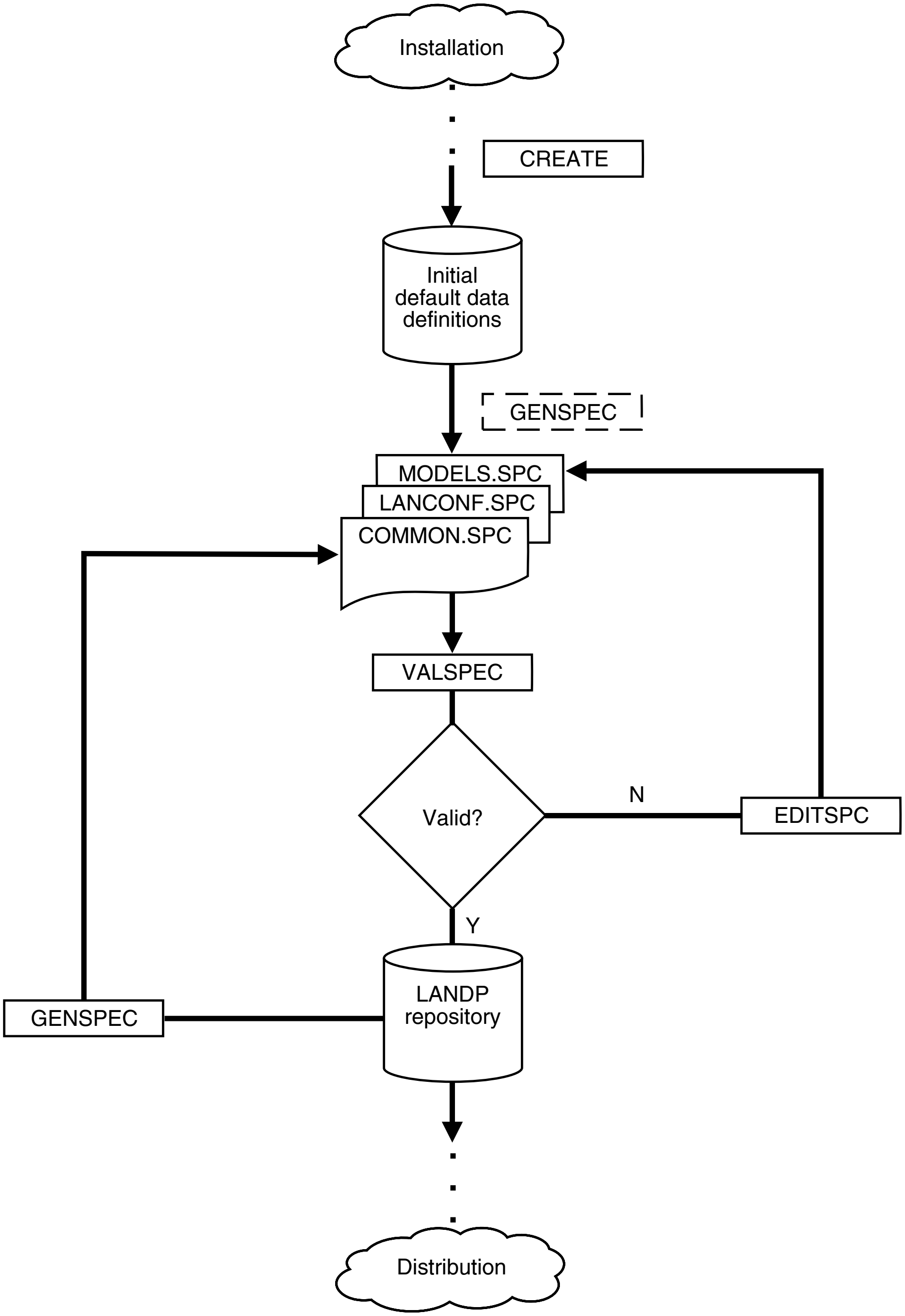


Figure 3. The Customization Steps

Validating customization data—VALSPEC

Start state
The editable .SPC files have been created using the GENSPEC or EDITSPC procedure.

Task
To validate the information created in the COMMON.SPC, LANCONF.SPC, and, if specified, MODELS.SPC files.

For more information
See Chapter 7, "Validating customization data—VALSPEC" on page 65.

Editing customization vectors—EDITSPC

Start state
There are two possible start states for running the EDITSPC procedure:

- The .SPC files have been created using the GENSPEC procedure. This may be after the default data definitions have been created, or at any point in time during customization of the existing configuration.
- The CREATE procedure has been run, but no .SPC files have been generated. In this case, you can use EDITSPC to create the files from scratch.

Task
To edit the customization vectors within the .SPC files.

For more information
See Chapter 6, "Editing customization vectors—EDITSPC" on page 51.

Note: You can edit the .SPC files using any text editor. Using EDITSPC has the advantage of providing a standard template for the vectors, and easy access to VALSPEC and the help files.

Migrating customization data from FBSS

You can migrate customization data from an earlier version of FBSS to LANDP. Figure 4 shows how the MIGRATE procedure fits into the customization process.

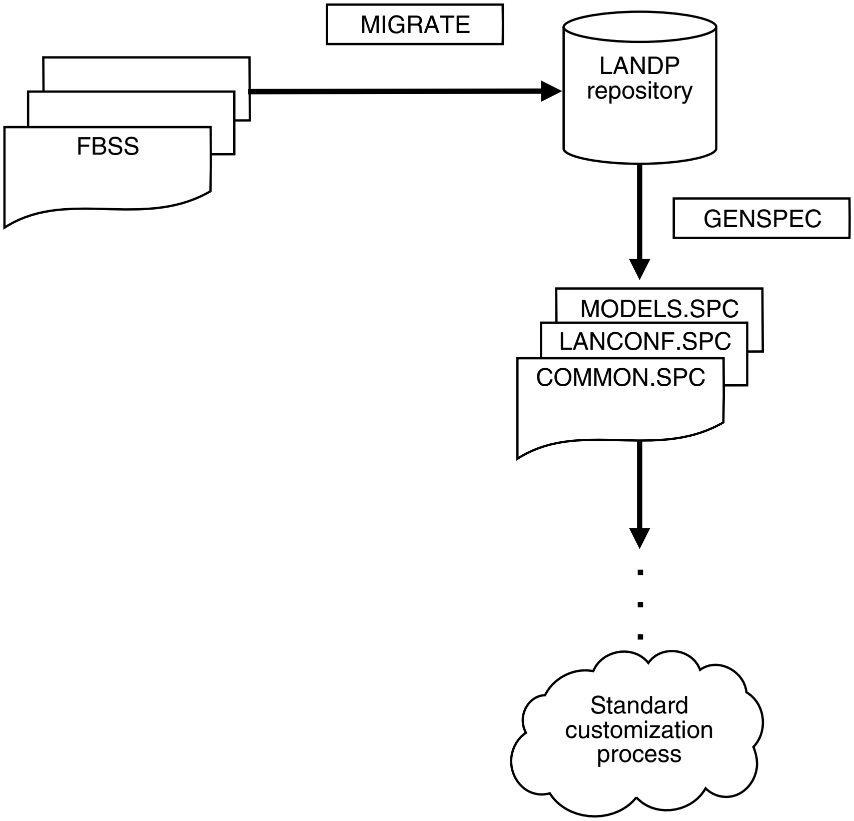


Figure 4. Migrating customization data from FBSS

Following the MIGRATE procedure, the customization process follows the same steps as for a new LANDP workgroup.

For more information see “Migrating” on page 240.

Chapter 4. Creating common data default definitions—CREATE

OS/2 DOS 6000 NT

The CREATE utility program is used to generate common data default definitions, held as vectors in the COMMON.SPC file.

Whether or not you need to specify common vectors depends on the servers in your workgroup. For a reference list of which vectors you have to specify at the common level, see “Vectors - a quick reference” on page 54. If you do not need to specify common vectors, you can omit the CREATE step.

Note: The CREATE utility runs only in DOS, OS/2, and Windows NT. The common data definitions can apply to all platforms.

Starting the CREATE procedure

You can display *online information* about the CREATE procedure. From the EHCCUS directory, enter:

```
CREATE ?
```

To start the CREATE procedure, from the EHCCUS directory, enter:

```
CREATE
```

The common data default definitions are created in the internal repository, and held as vectors on the COMMON.SPC file.

Note: If a COMMON.SPC file already exists when you run CREATE, the program performs a validation of the existing COMMON.SPC file.

Chapter 5. Converting common data to editable .SPC files—GENSPEC

OS/2 DOS 6000 NT

This chapter explains how to perform the vector generation step. It shows how to start the generation procedure, and the data generated by the procedure.

The generation procedure reads data in the internal repository, created either by the CREATE procedure, the migration procedure, or by the validation procedure, and generates vectors in the .SPC files.

When LANDP is first installed, it is advisable to run CREATE followed by GENSPEC to create a COMMON.SPC file containing the source for all the default tables. This is useful if you later want to create or change items such as keyboard definitions and translation tables.

If a .SPC file with the same name and in the same path already exists, the procedure creates a backup of the existing .SPC file, with the same file name and extension BAK. If a .BAK file corresponding to the .SPC file already exists, the procedure erases the existing .BAK file.

If a .SPC file is lost, you can recover it by running the generation procedure and using the current data in the internal repository.

Note: The GENSPEC utility runs only in DOS, OS/2, and Windows NT. The vectors generated can apply to all platforms.

Starting the generation procedure

You can display *online information* about the GENSPEC procedure. From the EHCCUS directory, enter:

```
GENSPEC ?
```

To start the generation procedure, from the EHCCUS directory, enter:

```
GENSPEC [parm1] [parm2] [parm3] [parm4]
```

where:

parm1 Is an optional parameter to specify the type of data to be processed. The parameter value can be:

COMMON To process common data, such as defaults, records, profiles, and tables.

LAN To process workgroup configuration data.

MODELS To process model configuration data.

The default is LAN.

parm2 Is an optional parameter to specify the workgroup configuration to be processed. It applies when you process only one workgroup configuration.

If the parameter is used, only the workgroup configuration specified will be processed, no matter the LANLIMIT.SPC file contents.

The parameter format is:

```
langroup\lanname
```

Replace *langroup* with the name of the directory on which the workgroup configuration resides. If you omit *langroup*, EHCCUS is assumed.

Replace *lanname* with the name of the workgroup configuration.

parm3 Is an optional parameter to specify the highest severity of return codes allowed. The parameter value can be:

1 To admit only informative return codes. Higher severity results in execution ending.

2 To admit informative and attention return codes. Higher severity results in execution ending.

3 To admit informative, attention, and error return codes. Higher severity results in execution ending.

The default is 1. For further information, refer to “Return codes” on page 240.

parm4 Is an optional parameter with only one possible value: DELETE.

If the parameter is specified, the customization data processed is automatically removed from the internal repository. Thus, after running the generation program, the data is stored *only* on the .SPC files.

If the parameter is not specified, the customization data processed also remains stored in the internal repository.

The parameters can be specified in any order. Return codes generated by the generation procedure are displayed on the screen.

To avoid lengthy processing, it is strongly recommended that you run the validation procedure specifying first the COMMON parameter value. Then, if it applies, run the procedure specifying the MODELS parameter value. Finally, run the validation procedure again specifying the LAN parameter value.

In addition, when processing workgroup configuration data, you can select the workgroup configurations affected by the process. See “Selecting workgroup configurations” on page 239.

Data in vector format

After generation of .SPC files, the customization data is stored in vector format in those files. The common data is stored in the COMMON.SPC file, in the EHCCUS directory.

The workgroup configuration data is stored in the LANCONF.SPC files, in the *yyyyyyyy*\xxxxxxxx directories, where:

yyyyyyyy Is the name of a group of workgroup configurations. The customization program default is EHCCUS.

xxxxxxxx Is the name of a workgroup configuration.

The vectors in a LANCONF.SPC file are:

LANCONF
LWSCONF

The model configuration data is stored in the MODELS.SPC file, in the EHCCUS directory. The vectors in that file are:

LANMODEL
LWSCONF
SVRMODEL
WSMODEL

Chapter 6. Editing customization vectors—EDITSPC

OS/2 DOS 6000 NT

Customization is done by editing customization vectors.

A *vector* is a set of keyword/parameter statements that define configuration items. These items correspond to common, workgroup, and model data definitions, which are written on the COMMON.SPC, LANCONF.SPC, and MODELS.SPC files.

Vectors can be created and written according to a set of rules.

Keywords are the names or symbols that identify parameters, or ordered strings of parameters.

Note: The EDITSPC utility runs only in DOS and OS/2, although the customized vectors can apply to all platforms. For information on how to use EDITSPC, refer to “Creating and editing common vectors” on page 287, “Creating and editing configuration vectors” on page 408, and “Editing model configuration data” on page 517.

Rules for using your own editor

Customization vectors can be created or edited with any *text editor* which adheres to the specifications below. Most commonly used editors already comply with these standards:

- Each line must start with a blank character (ASCII 20).
- Each line must end with a line separator character (ASCII 0A).
- Each line may or may not have a carriage return character (ASCII 0D) before the line separator (ASCII 0A).
- A blank character and a Tab character must be either ASCII 20 or ASCII 09. Both will be treated as blank characters (ASCII 20) and henceforth be called ‘blanks’.
- Line length can be fixed or variable. Maximum line length is 140 characters, including ASCII 0A and ASCII 0D.

Note: Your editor must neither add control nor format characters to the text. Therefore, you cannot use a word processor.

The EDITSPC customization program provides a procedure that, if the .SPC file or the path does not exist, creates the necessary path and opens the new .SPC file for input.

A file is provided for each operating system:

EDITSPC.BAT For DOS. This command calls the appropriate editor for your version of DOS (either E.EXE or EDIT.COM).

EDITSPC.CMD For OS/2. This command calls the OS/2 Enhanced Editor (EPM.EXE).

To have the OS/2 Enhanced Editor available, you must install that option from the OS/2 installation diskettes.

The command also enables the LANDP customization editing tool and adds the **LANDP** choice to the action bar of the editor window. For information about that tool, select the **View doc** option in the pull-down that appears when you choose **LANDP**.

Both files are located in the EHCCUS directory. If you want to use your own personal editor, modify the respective file accordingly.

For more information about using EDITSPC.CMD, refer to Appendix C, “Editing common data” on page 287 and “Creating and editing configuration vectors” on page 408.

Rules for specifying vectors

When creating or editing customization vectors in the COMMON.SPC, LANCONF.SPC, and MODELS.SPC files, you have to keep to the following syntax rules:

1. You can write vectors in upper case, lower case, or mixed case. However the validation program is not case sensitive. For example, if you named one workstation `A1` and another `a1`, they would be considered to be duplicates.

   This rule does not apply when text is enclosed within single quotes (').

   Note that the generating program always transforms vectors into upper case.

2. Any text starting with /* and ending with */ is a comment. For example:

   ```
   /* VECTOR1 defines the xx server properties */
   ```

   You can also use these characters to comment out vector lines.

3. At the beginning of a line (after any number of blank lines), you must have one or more blank characters before you start with the vector name. The vector definition never starts at the first position of a line. For example:

   ```
   /* VECTOR1 defines the xx server properties */

         VECTOR1
   ```

4. The vector must be separated from the first keyword by one or more blanks.

   The keyword is followed by an equal sign (=) and by its parameter specifications. For example:

   ```
   /* VECTOR1 defines the xx server properties */

         VECTOR1  KEYWORD1=PAR1
   ```

   You can have one or more blank separation characters before and after the equal sign.

5. A comma (,) is used as a continuation character. When there is no comma after a parameter, the vector specification is considered to be complete and the program looks for the next vector.

   A comma followed by a blank skips the rest of the line. For example:

   ```
   /* VECTOR1 defines the xx server properties */

         VECTOR1  KEYWORD1=PAR1,
                  KEYWORD2=PAR2,
                  KEYWORD3=PAR3,
                  KEYWORD4=PAR4
   ```

   Any text that you put in the same line after a comma followed by a blank is disregarded.

6. If a parameter contains blanks or special characters, it must be enclosed within single quotes (').

   Special characters are blanks, commas (,), single and double quotes (' and "), and parentheses (()). For example:

   ```
   /* VECTOR1 defines the xx server properties */

         VECTOR1  KEYWORD1=PAR1,
                  KEYWORD2=PAR2,
                  KEYWORD3=PAR3,
                  KEYWORD4=PAR4,
                  KEYWORD5='field description'
   ```

7. If a keyword has multiple parameters that are enclosed between parentheses, it may be split over several lines. The parameter values are separated by commas. You can split a line only after a comma. For example:

   ```
   /* VECTOR1 defines the xx server properties */

         VECTOR1  KEYWORD1=PAR1,
                  KEYWORD2=PAR2,
                  KEYWORD3=PAR3,
                  KEYWORD4=PAR4,
                  KEYWORD5='field description',
                  KEYWORD6=(PAR6a,PAR6b),
                  KEYWORD7=(PAR7a,PAR7b,
                            PAR7c,PAR7d,
                            PAR7e,PAR7f)
   ```

   Note that no blanks are allowed within parentheses.

8. All parameter values are positional. If you want to omit some parameter values in order to take the defaults, you must clearly point out their position by indicating all the commas. For example:

```
/* VECTOR1 defines the xx server properties */

     VECTOR1  KEYWORD1=PAR1,
              KEYWORD2=PAR2,
              KEYWORD3=PAR3,
              KEYWORD4=PAR4,
              KEYWORD5='field description',
              KEYWORD6=(PAR6a,PAR6b),
              KEYWORD7=(PAR7a,PAR7b,
                        PAR7c,PAR7d,
                        PAR7e,PAR7f),
              KEYWORD8=(PAR8a,,PAR8c,,,PAR8f)
```

   However, if the last parameter values of a keyword are defaulted, `KEYWORDx=(a,b,,,)`, they can be omitted. The format will then be `KEYWORDx=(a,b)`.

9. Keywords are not positional. They can appear in any order.

Vectors - a quick reference

The use of vectors is generally related to specific servers and functional areas.

Following is a list of LANDP functional areas and the vectors which have to be specified for each of them, both on common as well as on workgroup level. If you are using models, you will also have to specify the WSMODEL or the SVRMODEL vectors. They are not mentioned in the list.

Note that depending on your own configuration, you may have to specify other vectors as well. This list is only intended to give a quick overview. For detailed information, see Appendix C, “Editing common data” on page 287 and Appendix D, “Editing configuration data” on page 407.

Some LANDP functional areas in the list have a special mark:

(*) Apply to PC/Integrator

(**) Apply to both PC/Integrator and PC Integrator/2

Functional Area	On common level	Go to page:	On workgroup level	Go to page:
ASCII-EBCDIC translation server (EHCDBTR)	XLAT2TBL	402	LWSCONF DBCSXLAT	417 422
(**) Banking Interactive Workstation Program (BIWP)	DEFAULTS DISPLATT KBDBIWP KSCCBIWP KSTRBIWP LUPOOL MSRINTBL MSROUTBL PINPTBL XLATETBL	294 300 355 360 361 363 364 366 370 400	LWSCONF SERVER PAR&BIWP	417 422 428
(**) Banking Printer Program (BPP)	BPPPARM DEFAULTS FORM4710 FORM4720 LUPOOL XLATETBL	290 294 309 312 363 400	LWSCONF SERVER SES&BPP	417 422 431
Batch Machine Loader Server (BMLS)			LWSCONF SERVER PAR&BMLS	417 422 434
Batch Machine Operator (BMOP)			LWSCONF SERVER	417 422
CICS interface server (EHCTRAN)			LWSCONF SERVER LWSCONF CLIENT	417 422 417 424
Compression Server (EHCCOMP)			LWSCONF SERVER LWSCONF CLIENT	417 422 417 424
DDE Access Server (EHCLAD)			LWSCONF SERVER LWSCONF CLIENT	417 422 417 424
(*) Device Cluster Attachment DLC (DCADLC)			LWSCONF SERVER PAR&DCA	417 422 435

Functional Area	On common level	Go to page:	On workgroup level	Go to page:
Electronic journal server (ELECJO##)	EJOUPRF EJOUREC RECDEF RECFIELD	304 308 374 377	LWSCONF SERVER PAR&EJOU LWSCONF CLIENT	417 422 436 417 424
Financial printer server (PR47X2##)	FORM47X2	318	LWSCONF SERVER PAR&47X2 LWSCONF CLIENT SES&47X2	417 422 497 417 424 501
Forwarding server (FORWARD)	FORWPRF FORWDS LUPOOL XLATETBL	344 341 363 400	LWSCONF SERVER PAR&FORW SES&FORW	417 422 437 438
IBM 4731, 8, 9 Personal Banking Machine (SS#####)			LWSCONF SERVER PAR&4731 LWSCONF CLIENT	417 422 506 417 424
IBM 4733 Teller Assist Unit (DTAU4733)			LWSCONF SERVER PAR&4733 LWSCONF CLIENT	417 422 508 417 424
IBM 4737 Self-Service Transaction Station (SS######)			LWSCONF SERVER PAR&4737 PAR&PBMS LWSCONF CLIENT	417 422 509 509 417 424
Local resource manager server (EHCLRMGR)			LWSCONF SERVER	417 422
(**) Logical device address (LDA) 7 program (LDA7)	LUPOOL	363	LWSCONF SERVER PAR&LDA7	417 422 440

Functional Area	On common level	Go to page:	On workgroup level	Go to page:
Magnetic stripe reader/encoder server (MSRE47##)			LWSCONF SERVER PAR&MSRE LWSCONF CLIENT	417 422 443 417 424
MQSeries Link server (EHCMQ##)			LWSCONF SERVER PAR&MQ LWSCONF CLIENT	417 422 441 417 424
Multiple Virtual DOS Machine Relay (EHCVDMGR)			LWSCONF SERVER	417 422
Native X.25 (X25NAT##)			LWSCONF SERVER PAR&XNAT LWSCONF CLIENT SES&NSVC	417 422 444 417 424 445
Object Post Box Server (OPBS)			LWSCONF SERVER PAR&OPBS LWSCONF CLIENT	417 422 447 417 424
ODBC Query Server (EHCODB##)			LWSCONF SERVER PAR&ODB LWSCONF CLIENT	417 422 447 417 424
Operator Interface (OPER)			LWSCONF SERVER	417 422
PIN pad server (PINP47##)			LWSCONF SERVER PAR&PINP LWSCONF CLIENT	417 422 449 417 424
Printer manager server (PRTMGR)			LWSCONF SERVER	417 422

Functional Area	On common level	Go to page:	On workgroup level	Go to page:
Program-to-program communication (PPC) server (LU 6.2) (PPC) if LANDP for OS/2 or Windows NT (PPC####) if LANDP for AIX			LWSCONF SERVER LWSCONF CLIENT	417 422 417 424
Query server (EHCSQL##)			LWSCONF SERVER PAR&SQL LWSCONF CLIENT	417 422 450 417 424
Remote Change Management Services (RMCS)	LUPOOL RCMSLNF XLATETBL	363 373 400	LWSCONF SERVER PAR&RCMS	417 422 451
Searcher for electronic journal server and store-for-forwarding server (SFQUERY)			LWSCONF SERVER	417 422
Service Availability Manager. (EHCSAM)			LWSCONF SERVER	417 422
Shared DOS Directory (SHRDIR)			LWSCONF SERVER PAR&SHDR LWSCONF CLIENT SES&SHDR	417 422 454 417 424 454
Shared-file distributor (EHCSFD##)			LWSCONF SERVER PAR&SFD LWSCONF CLIENT	417 422 455 417 424
Shared-file replicator (EHCSFR##)			LWSCONF SERVER PAR&SFR LWSCONF CLIENT	417 422 457 417 424

Functional Area	**On common level**	**Go to page:**	**On workgroup level**	**Go to page:**
Shared-file server (SHFILE##)	COLSQTBL SHFLDBD SHFLPCB RECDEF RECFIELD SHFLSGM	292 383 388 374 377 389	LWSCONF SERVER PAR&SHFL LWSCONF CLIENT	417 422 459 417 424
SNA server (SNA##)	LUPOOL X25DIR	363 404	LWSCONF SERVER PAR&SNA LWSCONF CLIENT SES&SNA SES&SPVC SES&SSVC SBSX25	417 422 463 417 424 464 465 466 468
Store-for-forwarding server (SFORFORW)	SFORWPRF SFORWREC RECDEF RECFIELD	379 382 374 377	LWSCONF SERVER PAR&SFOR LWSCONF CLIENT	417 422 470 417 424
Synchronous data link control server (SDLC)	LUPOOL	363	LWSCONF SERVER PAR&SDLC	417 422 471
System manager (SMGR)	SMGRPRF SMGRUSER RECDEF RECFIELD	392 396 374 377	LWSCONF SERVER PAR&SMGR LWSCONF CLIENT	417 422 473 417 424
System manager operator (SMOP)			LWSCONF SERVER PAR&SMOP	417 422 472
Distribution (SOFTPACK)	SOFTPACK	399	LWSCONF SOFTPACK	417 420
TCP/IP wide area communication server (EHCTCP)			LWSCONF SERVER PAR&TCP SES&TCP LWSCONF CLIENT	417 422 476 477 417 424

Functional Area	On common level	Go to page:	On workgroup level	Go to page:
Token-ring data link control server (TRDLC)	LUPOOL	363	LWSCONF SERVER PAR&TKR	417 422 481
Trace tools For LANDP for DOS (DDT)			LWSCONF SERVER	417 422
Trace tools For LANDP for OS/2 and Windows NT (EHCTRACW)			LWSCONF SERVER	417 422
X.25 data link control server For IBM X.25 Co-Processor Adapters (X25DLC)			LWSCONF SERVER PAR&X25D	417 422 486
X.25 data link control server For IBM X.25 Interface Co-Processor/2 (X25DLC2)			LWSCONF SERVER PAR&X252	417 422 488
3270 Emulator (EMU3270)	DEFAULTS DISPLATT KBD3270 LUPOOL XLATETBL	294 300 348 363 400	LWSCONF SERVER PAR&3270 SES&3270	417 422 490 491
3287 Printer emulator (EMU3287)	DEFAULTS LUPOOL P3287ATT XLATETBL	294 363 368 400	LWSCONF SERVER PAR&3287 SES&3287	417 422 494 495
(*) 4700 Virtual file support (VFILE)	LUPOOL	363	LWSCONF SERVER PAR&VFIL	417 422 483
(**) 4700 Virtual volume support (RDVVOLS)	LUPOOL	363	LWSCONF SERVER PAR&VVOL	417 422 484
4721 Printer integrator (PT4721)	LUPOOL	363	LWSCONF SERVER PAR&PT21 LWSCONF CLIENT	417 422 503 417 424
4721 Self service printer server (SP4721##)			LWSCONF SERVER PAR&SP21 LWSCONF CLIENT	417 422 503 417 424

Functional Area	On common level	Go to page:	On workgroup level	Go to page:
4748 printer server (PR4748##)	FORM4748	329	LWSCONF SERVER PAR&4748 LWSCONF CLIENT SES&4748	417 422 511 417 424 513
4770 printer server (PR4770##)	FORM4770	337	LWSCONF SERVER PAR&4770 LWSCONF CLIENT SES&4770	417 422 515 417 424 516

Server requirements

Following is a definition of the prerequisites for a workstation in order to load a server or a functional area.

Server	Workstation Requirements
BIWP	• Must be client of SNA## • If loaded in MVDM, the EHCVDMGR server must be in the same workstation.
BMOP	• Must be client of SMGR • Must be client of OPBS.
BPP	• Must be client of SNA## • The PR47X2## server must be in the same workstation.
BMLS	• Must be client of OPBS.
EHCSFD##	• Must be client of the SHFILE## servers, the EHCSFD## distributors, and the EHCSFR## replicators that are owned by the EHCSFD## distributor you define. • At least one SHFILE## server must be in the same workgroup. • At least one of the following conditions must be fulfilled: – A SHFILE## server in the workgroup is owned by the EHCSFD## distributor you define, through PAR&SHFL specifications. – A EHCSFD## distributor in the workgroup is owned by the EHCSFD## distributor you define, through PAR&SFD specifications. – A EHCSFR## replicator in the workgroup is owned by the EHCSFD## distributor you define, through PAR&SFR specifications.
EHCSFR##	• Must be client of the SHFILE## servers, the EHCSFD## distributors, and the EHCSFR## replicators that are owned by the EHCSFR## replicator you define. • At least one SHFILE## server must be in the same workgroup. • At least one of the following conditions must be fulfilled: – A SHFILE## server in the workgroup is owned by the EHCSFR## replicator you define, through PAR&SHFL specifications. – A EHCSFD## distributor in the workgroup is owned by the EHCSFR## replicator you define, through PAR&SFD specifications. – A EHCSFR## replicator in the workgroup is owned by the EHCSFR## replicator you define, through PAR&SFR specifications.
EHCTCP	• If being used to emulate the SNA## server, the SNA## server must be defined to run in the same workstation. • If being used to emulate the PPC server, the PPC server must be defined to run in the same workstation.
ELECJO##	• Must be client of SMGR • The SFQUERY server must be in the same workstation • The SHFILE## server must be in the same workstation. For LANDP for AIX • The EHCSQL## server must be in the same workstation.
EMU3270	• Must be client of SNA## • If loaded in MVDM, the EHCVDMGR server must be in the same workstation. • In DBCS mode, the EHCDBTR server must be in the same workstation.
EMU3287	• Must be client of SNA## • If loaded in MVDM, the EHCVDMGR server must be in the same workstation. • In DBCS mode, the EHCDBTR server must be in the same workstation.

Server	Workstation Requirements
FORWARD	• Must be client of SMGR • Must be client of SNA## • Must be client of SFORFORW. • In DBCS mode, the EHCDBTR server must be in the same workstation.
LDA7	• Must be client of SNA## • If loaded in MVDM, the EHCVDMGR server must be in the same workstation.
OPBS	• Must be client of SMGR • The SMGR profile must have, at least, one user defined • The SHFILE## server must be in the same workstation.
PBMS	• Must be client of SNA## • The SS###### server must be in the same workstation.
PT4721	• Must be client of SNA## • The SP4721## server must be in the same workstation.
RCMS	• Must be client of SNA##. • In DBCS mode, the EHCDBTR server must be in the same workstation.
RDVVOLS	• Must be client of SNA##, or • The DCADLC server must be in the same workstation.
SDLC	• The SNA## server must be in the same workstation.
SFORFORW	For LANDP for DOS, OS/2, and Windows NT • Must be client of SMGR • The SFQUERY server must be in the same workstation • The SHFILE## server must be in the same workstation. For LANDP for AIX • The EHCSQL## server must be in the same workstation.
SMOP	• Must be client of SMGR.
SNA##	• In DOS, it must have a DLC server in the same workstation (DCADLC, SDLC, TRDLC, X25DLC, or X25DLC2).
TRDLC	• The SNA## server must be in the same workstation.
VFILE	• Must be client of SNA## in the same workstation.

Server information

Online help, including an example, is available for each LANDP functional area.

You can use any *text editor* to call this online help. In the EHCHELP sub-directory of the EHCCUS directory, open with your editor:

xxxxxxxx

where *xxxxxxxx* stands for the name of the functional area. A list of the names follows.

BIWP
BMOP
BPP
DCADLC
DTAU4733
BMLS
EHCTRAN
EHCLAD
EHCLRMGR
EHCMQ##
EHCODB##
EHCSAM
EHCSFD##
EHCSFR##
EHCSQL##
EHCTCP
EHCVDMGR
ELECJO##
EMU3270
EMU3287
FORWARD
LDA7
MSRE47##
OPBS
OPER
PBMS
PINP47##
PPC
PR47X2##
PR4748##
PR4770##
PRTMGR
PT4721
RCMS
RDVVOLS
SDLC
SFORFORW
SFQUERY
SHFILE##
SHRDIR
SMGR
SMOP
SNA##
SP4721##
SS#####
SS######
TRDLC
VFILE
X25DLC
X25DLC2
X25NAT##

Chapter 7. Validating customization data—VALSPEC

OS/2 DOS 6000 NT

You have to run the validation procedure, VALSPEC to validate customization data. This procedure reads customization data from the .SPC files, and checks the consistency of the data. The validation procedure generates data in the internal repository.

Note: The VALSPEC utility runs in DOS, OS/2, and Windows NT. The data that is being validated can apply to all platforms.

Selecting workgroup configurations

This section applies when you want to process more than one workgroup configuration at the same time.

The LANLIMIT.SPC file is located in the EHCCUS directory. You can edit the LANLIMIT.SPC file using any text editor, and specify the workgroup configurations to be processed.

- To *select* workgroup configurations, use INCLUDE statements. Those workgroup configurations not specified, explicitly or implicitly, are not processed. The format of the INCLUDE statement is:

  ```
  INCLUDE = langroup\lanname
  ```

- To *exclude* workgroup configurations, use OMIT statements. Those workgroup configurations not specified, explicitly or implicitly, are processed. The format of the OMIT statement is:

  ```
  OMIT = langroup\lanname
  ```

Replace *langroup* with the name of the directory on which the workgroup configuration resides. If you omit *langroup*, EHCCUS is assumed.

Replace *lanname* with the name of the workgroup configuration.

You cannot use both INCLUDE and OMIT statements at the same time.

Comments must start with /* and end with */.

The LANLIMIT.SPC file provided with the customization program has the following contents, and specifies that all the workgroup configurations will be processed.

```
INCLUDE = *
```

You can modify the LANLIMIT.SPC file to meet your requirements.

Examples

Example 1:

```
/* LANLIMIT.SPC Example 1*/
   INCLUDE = GROUP2\*
   INCLUDE = GROUP4\CONF47
```

Only the workgroup configurations located in the GROUP2 directory, and the workgroup configuration named CONF47 and located in the GROUP4 directory, will be processed.

Example 2:

```
/* LANLIMIT.SPC Example 2 */
   OMIT = GROUP6\*
```

All the workgroup configurations, except for those located in the GROUP6 directory, will be processed.

Example 3:

```
/* LANLIMIT.SPC Example 3*/
   INCLUDE = GROUP8\*
   INCLUDE = \CONF13
```

Only the workgroup configurations located in the GROUP8 directory, and the workgroup configuration named CONF13 and located in the EHCCUS directory, will be processed.

Return codes

The return codes generated by the validation procedure are classified into four types. The following list shows the types of return codes, starting with the least severe; an identifier for each type appears in parenthesis.

1. *Informative* (I) : Some input may be missing or incorrectly specified.
2. *Attention* or *Warning* (W) : Some secondary functions may not work properly at the production sites.
3. *Error* (E) : Some LANDP functional areas will not work properly.
4. *Severe* (S) : The execution will be ended.

The identifier of the type of return code is displayed as the last character of the return code. For example, the following is displayed on the screen:

```
01159 I Host identification default for session not defined.
```

This means that the validation procedure has ended with an informative return code.

When running the validation procedure, you can specify the highest severity allowed.

Starting the validation procedure

You can display *online information* about the VALSPEC procedure. From the EHCCUS directory, enter:

```
VALSPEC ?
```

To start the validation procedure, from the EHCCUS directory, enter:

```
VALSPEC [parm1] [parm2] [parm3]
```

where:

parm1 Is an optional parameter to specify the type of data to be processed. The parameter value can be:

`COMMON` To process common data such as defaults, records, profiles, and tables.

`LAN` To process workgroup configuration data.

`MODELS` To process model configuration data.

When you specify `LAN` or `MODELS`, if there is no common data in the internal repository, `COMMON` is also assumed. In this case, if a COMMON.SPC file is available, this file is processed to generate common data. If not available, a COMMON.SPC file containing only the DEFAULT vector with no keyword is generated and processed.

parm2 Is an optional parameter to specify the workgroup configuration to be processed. It applies when you process only one workgroup configuration.

If the parameter is used, only the workgroup configuration specified is processed, regardless of the LANLIMIT.SPC file contents.

The parameter format is:

```
langroup\lanname
```

Replace *langroup* with the name of the directory on which the workgroup configuration resides. If you omit *langroup*, EHCCUS is assumed.

Replace *lanname* with the name of the workgroup configuration.

parm3 Is an optional parameter to specify the highest severity of return codes allowed. The parameter value can be:

1 To admit only informative return codes. Higher severity results in execution ending.

2 To admit informative and attention return codes. Higher severity results in execution ending.

3 To admit informative, attention, and error return codes. Higher severity results in execution ending.

The default is 1. For further information, refer to “Return codes” on page 66.

The parameters can be specified in any order. The CUSPARM.LST file, which is generated in the EHCCUS subdirectory, contains the vectors and pointers to any problems that may arise.

If you use the OS/2 Enhanced Editor with the LANDP customization editing tool, you can also start the validation procedure using that tool.

Select the **LANDP** choice from the action bar in the editor window, and the **Validate** option in the pull-down that appears when you choose **LANDP**.

For further information, select the **View doc** option in that pull-down.

To avoid long processing, it is strongly recommended to run the validation procedure specifying first the COMMON parameter value. Then, if it applies, run the procedure specifying the MODELS parameter value. Finally, run again the validation procedure specifying the LAN parameter value.

In addition, when processing workgroup configuration data, you can select the workgroup configurations affected by the process. See “Selecting workgroup configurations” on page 65.

Example

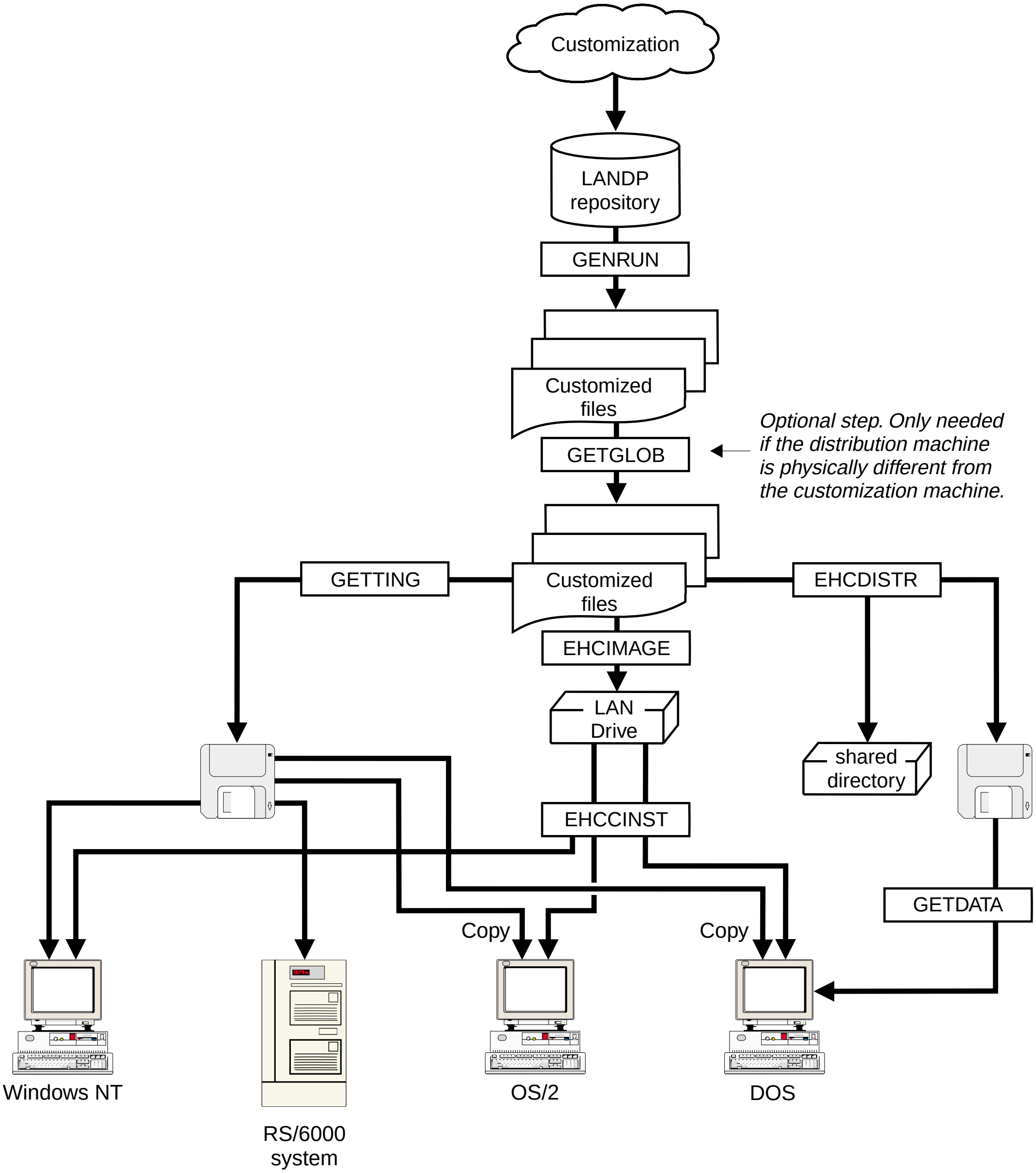
```
VALSPEC LAN \Sample1
```

This command validates the Sample1 workgroup configuration on the EHCCUS directory. Only informative return codes are allowed; higher severity return codes result in execution ending.

Part 4. Distribution

This part describes how to distribute your LANDP configuration software to the workstations in your LANDP workgroup.

A diagram showing the stages of the distribution process follows.


Customization
LANDP repository
GENRUN
Customized files
GETGLOB
Optional step. Only needed if the distribution machine is physically different from the customization machine.
Customized files
GETTING
EHCDISTR
EHCIMAGE
LAN Drive
shared directory
EHCCINST
GETDATA
Copy
Copy
Windows NT
RS/6000 system
OS/2
DOS

Chapter 8. Obtaining software for distribution—GENRUN

OS/2 DOS 6000 NT

The customization process creates software to be distributed to the production workstations.

This chapter contains information describing how to obtain that software, and copy it to the distribution workstation.

When the software is ready for distribution:

- To distribute updates to DOS, OS/2, or Windows NT workstations in an existing workgroup configuration, you can directly use RCMS and NetView DM. For more information, see *LANDP Servers and System Management*.
- To distribute a new workgroup configuration, refer to Chapter 9, “Distributing software” on page 77.

To obtain the software needed by the workstations:

1. **Generate run-time files.**

   This step is always required, and is performed using the GENRUN procedure on the customization workstation.

   The GENRUN procedure reads data in the internal repository, created either by the migration procedure or the validation procedure, and generates run-time files. For each defined workstation, it generates a GETTING.SPC file that contains information about the software to be installed. The file is located in a directory where the directory name is the workstation ID.

   For information on how to run the procedure, refer to “Starting the run-time generation procedure—GENRUN” on page 74.

2. **Copy run-time files to the distribution workstation.**

   This step is optional, and can be skipped if the customization workstation will be used for distribution purposes.

   If you have carried out the customization process on a unique customization workstation located in the development site, you need to copy the run-time files to the distribution workstation using the GETGLOB procedure on the customization workstation.

   The GETGLOB procedure creates, on a specified drive, an image of the customization directory structure corresponding to a specified workgroup configuration. Only one copy of the run-time files needed by the workgroup is made.

   If the distribution workstation cannot be accessed remotely, you can specify a diskette drive and create a diskette set with the customization image.

   For information on how to run the procedure, refer to “Starting the procedure for getting customization data—GETGLOB” on page 75.

Note: The GENRUN and GETGLOB utilities run only in DOS, OS/2, and Windows NT. The software generated can be distributed to all platforms.

Selecting workgroup configurations

This section applies when you want to process more than one workgroup configuration at the same time.

The LANLIMIT.SPC file is located in the EHCCUS directory. You can edit the LANLIMIT.SPC file using any text editor, and specify the workgroup configurations to be processed.

- To *select* workgroup configurations, use INCLUDE statements. Those workgroup configurations not specified, explicitly or implicitly, are not processed. The format of the INCLUDE statement is:

  ```
  INCLUDE = langroup\lanname
  ```

- To *exclude* workgroup configurations, use OMIT statements. Those workgroup configurations not specified, explicitly or implicitly, are processed. The format of the OMIT statement is:

  ```
  OMIT = langroup\lanname
  ```

Replace *langroup* with the name of the directory on which the workgroup configuration resides. If you omit *langroup*, EHCCUS is assumed.

Replace *lanname* with the name of the workgroup configuration.

You cannot use both INCLUDE and OMIT statements at the same time.

Comments must start with /* and end with */.

The LANLIMIT.SPC file provided with the customization program has the following contents, and specifies that all the workgroup configurations will be processed:

```
INCLUDE = *
```

You can modify the LANLIMIT.SPC file to meet your requirements.

Examples

Example 1:

```
/* LANLIMIT.SPC Example 1*/
   INCLUDE = GROUP2\*
   INCLUDE = GROUP4\CONF47
```

Only the workgroup configurations located in the GROUP2 directory, and the workgroup configuration named CONF47 and located in the GROUP4 directory, will be processed.

Example 2:

```
/* LANLIMIT.SPC Example 2 */
   OMIT = GROUP6\*
```

All the workgroup configurations, except for those located in the GROUP6 directory, will be processed.

Example 3:

```
/* LANLIMIT.SPC Example 3*/
   INCLUDE = GROUP8\*
   INCLUDE = \CONF13
```

Only the workgroup configurations located in the GROUP8 directory, and the workgroup configuration named CONF13 and located in the EHCCUS directory, will be processed.

Return codes

The return codes generated by both the run-time generation procedure and the getting procedure are classified into four types. The following list shows the types of return codes, starting with the least severe; an identifier for each type appears in parenthesis.

1. *Informative* (I) : Some input may be missing or incorrectly specified.
2. *Attention* or *Warning* (W) : Some secondary functions may not work properly at the production sites.
3. *Error* (E) : Some LANDP functional areas will not work properly.
4. *Severe* (S) : The execution will be ended.

The identifier of the type of return code is displayed as the last character of the return code. For example, the following is displayed on the screen:

```
00011 S COMMON specifications must be processed
        before RUN TIME generation
```

This means that the run-time generation procedure has ended with a severe return code.

When running both the run-time generation and getting procedure, you can specify the highest severity allowed.

Starting the run-time generation procedure—GENRUN

You can display *online information* about the GENRUN procedure. From the EHCCUS directory, enter:

```
GENRUN ?
```

To start the run-time generation procedure, from the EHCCUS directory, enter:

```
GENRUN [parm1] [parm2] [parm3]
```

where:

parm1 Is an optional parameter to specify the type of data to be processed. The parameter value can be:

COMMON To process common data, such as defaults, records, profiles, and tables.

LAN To process workgroup configuration data.

The default is LAN.

When you change existing data, or add new data, to the COMMON.SPC file, you have to reprocess the workgroup configurations affected by the modifications.

parm2 Is an optional parameter to specify the workgroup configuration to be processed. It applies when you process only one workgroup configuration.

If the parameter is used, only the workgroup configuration specified will be processed, no matter the LANLIMIT.SPC file contents.

The parameter format is:

```
langroup\lanname
```

Replace *langroup* with the name of the directory on which the workgroup configuration resides. If you omit *langroup*, EHCCUS is assumed.

Replace *lanname* with the name of the workgroup configuration.

parm3 Is an optional parameter to specify the highest severity of return codes allowed. The parameter value can be:

1 To admit only informative return codes. Higher severity results in execution ending.

2 To admit informative and attention return codes. Higher severity results in execution ending.

3 To admit informative, attention, and error return codes. Higher severity results in execution ending.

The default is 1. For further information, refer to “Return codes” on page 73.

To process data created by the validation procedure or by the migration procedure, with return code 2 or 3, you have to specify *parm3* with a parameter value equal to or higher than the return code obtained.

To avoid lengthy processing, it is strongly recommended that you run the run-time generation procedure specifying first the COMMON parameter value. Then, run the procedure again specifying the LAN parameter value.

In addition, when processing workgroup configuration data, you can select the workgroup configurations affected by the process. See “Selecting workgroup configurations” on page 72.

Example

```
GENRUN \Sample1
```

This command generates the distributable workgroup configuration data files for the Sample1 workgroup on the EHCCUS directory. Only informative return codes are allowed; higher severity return codes result in execution ending.

Starting the procedure for getting customization data—GETGLOB

You can display *online information* about the GETGLOB procedure. From the EHCCUS directory, enter:

```
  GETGLOB ?
```

To start the procedure for getting customization data, from the EHCCUS directory, enter:

```
  GETGLOB parm1 [parm2] [parm3]
```

where:

parm1 Is a required parameter to specify the workgroup configuration to be processed.

The parameter format is:

```
  langroup\lanname
```

Replace *langroup* with the name of the directory on which the workgroup configuration resides. If you omit *langroup*, EHCCUS is assumed.

Replace *lanname* with the name of the workgroup configuration.

parm2 Is an optional parameter to specify the highest severity of return codes allowed. The parameter value can be:

1 To admit only informative return codes. Higher severity results in ending execution.

2 To admit informative and attention return codes. Higher severity results in ending execution.

3 To admit informative, attention, and error return codes. Higher severity results in ending execution.

The default is 1. For further information, refer to “Return codes” on page 73.

parm3 Is an optional parameter to specify the path where the run-time files are to be copied. The parameter format is:

path

Replace *path* with the required path, for example `D:`. The default path is `A:`.

The GETGLOB procedure creates on the specified drive an image of the directory structure generated during customization.

Example:

```
GETGLOB \Sample1
```

This command gets the customization data for the Sample1 workgroup configuration on the EHCCUS directory, and copies it to the A: drive. Only informative return codes are allowed; higher severity return codes result in execution ending.

Chapter 9. Distributing software

OS/2 DOS
6000 NT

After obtaining software for distribution, the run-time files are located in the distribution workstation. The distribution workstation can be the same as the customization workstation.

From the distribution workstation, you distribute the files to the individual workstations.

Note: The software must be distributed **from** a DOS, OS/2, or Windows NT workstation, although it can be distributed **to** all platforms.

You can accomplish this task in three ways:

- **Using the GETTING procedure.**

  This procedure reads information from the GETTING.SPC files created by the run-time generation procedure, and copies the software for each workstation to the diskette drive or path specified.

  For information on how to run the procedure, refer to “Starting the getting procedure—GETTING” on page 78.

  If the software is copied to diskettes, you should format the diskettes before writing to them. The customization program does not erase the contents of a diskette before writing on it.

  For LANDP for DOS workstations that need only one diskette, you can format the diskette with the system files included, in order to perform the IPL from that diskette. Including the system files enables you to use the diskette for testing, or as backup.

  If the drive on the distribution workstation that contains the customization data is *shared* by the production DOS, OS/2, or Windows NT workstations, you can run the GETTING procedure from each individual workstation. The procedure can be run simultaneously on all the workstations, using a specific parameter.

  Note: The LANDP GETTING process uses a 16-bit implementation that can be run under DOS or OS/2, or from a Windows NT command prompt in an MS-DOS window.

- **Using the shared DOS directory server.**

  This option requires that the distribution workstation must be integrated in the workgroup.

  If you plan to use the shared DOS directory server for distribution purposes, see “Using the shared DOS directory server for distribution” on page 80.

  To meet your requirements, the software may need to be modified. For details of a utility program to modify files, see “Modifying file contents” on page 83.

  For information on how to set up the LANDP workstations in your workgroups, refer to:

  - Chapter 11, “Preparing DOS workstations” on page 95
  - Chapter 12, “Preparing OS/2 workstations” on page 127
  - Chapter 13, “Preparing Windows NT workstations” on page 167.
  - Chapter 14, “Preparing RS/6000 workstations” on page 203.
- **Using a distribution server**.

  This option uses a common disk (such as a LAN drive) shared between the distribution workstation and the individual workstation. See Chapter 10, “Distributing software using a distribution server” on page 85 for more information.

Return codes

The return codes generated by both the run-time generation procedure and the getting procedure are classified into four types. The following list shows the types of return codes, starting with the least severe; an identifier for each type appears in parenthesis.

1. *Informative* (I) : some input may be missing or incorrectly specified.
2. *Attention* or *Warning* (W) : some secondary functions may not work properly at the production sites.
3. *Error* (E) : some LANDP functional areas will not work properly.
4. *Severe* (S) : the execution will be ended.

The identifier of the type of return code is displayed as the last character of the return code. For example, the following is displayed on the screen:

```
00042I A file is missing. Correct the error and run GETTING again.
```

This means that the GETTING procedure has ended with an informative return code.

When running both the run-time generation and getting procedure, you can specify the highest severity allowed.

Starting the getting procedure—GETTING

You can display *online information* about the GETTING procedure. From the EHCCUS directory, enter:

```
GETTING ?
```

To start the getting procedure, from the EHCCUS directory, enter:

```
GETTING parm1 [WS=parm2] [parm3] [parm4] [/W:parm5]
```

where:

parm1 Is a required parameter to specify the workgroup configuration.

The parameter format is:

```
langroup\lanname
```

Replace *langroup* with the name of the directory on which the workgroup configuration resides. If you omit *langroup*, EHCCUS is assumed.

Replace *lanname* with the name of the workgroup configuration.

parm2 Is an optional parameter to specify the workstation in the workgroup configuration.

The parameter format is:

```
workstation name
```

Replace *workstation name* with the name of your workstation, for example `AA`.

The default is all workstations (*WS=ALL*).

parm3 Is an optional parameter to specify the highest severity of return codes allowed. The parameter value can be:

1 To admit only informative return codes. Higher severity results in execution ending.

2 To admit informative and attention return codes. Higher severity results in execution ending.

3 To admit informative, attention, and error return codes. Higher severity results in execution ending.

The default is 1. For further information, refer to “Return codes” on page 78.

Note that the only check the GETTING program does is to ensure that all the software that is being copied to the specified path is available.

parm4 Is an optional parameter to specify the path where the run-time files are to be copied. The parameter format is either the drive name, for example `A:`, or a full path name ending with a backslash, for example `B:\PLACE\`. A maximum of three directory levels is allowed. The default is `A:`.

If the target drive is demountable, and `WS=ALL` is specified (or defaulted) the copy must be targeted at the root directory, for example:

```
getting \sample ws=all 1 a:\
```

When running GETTING on Windows NT, the length of each directory name must be 8 bytes or less.

If, in the path you specify, a directory already exists with the workstation ID as the directory name, and the directory is not empty, the program can overwrite the contents before copying the run-time files to that directory.

parm5 Is an optional parameter to specify a working directory. The CUSPARM.LST file is to be generated in the EHCCUS directory that is created under the path you specify. The default is the current path.

The parameter is required, if you run the GETTING procedure on different workstations simultaneously.

The parameter format is:

```
drive:[\directory\]
```

Replace *drive* with the required drive name. Replace *directory* with any required directory name. Always include the final \ after the directory name.

Example:

```
GETTING \Sample1 WS=AA 1 A:\
```

This command reads the information from the GETTING.SPC files created by the GENRUN procedure in the EHCCUS directory. This information is held in the Sample1 workgroup configuration. The files for workstation AA are copied to the A: drive, home directory.

Only information return codes are allowed. Higher severity return codes result in execution ending.

Using the shared DOS directory server for distribution

To distribute the files from the distribution workstation to the individual workstations in the workgroup, you can use the shared DOS directory server.

Because this is a DOS process, all involved workstations must run DOS. This means that you must boot OS/2 and Windows NT workstations to DOS. The required version of the operating system is IBM DOS Version 7.1.

The process is as follows:

1. Create a diskette to be loaded in the distribution workstation. This workstation must be integrated in the workgroup, and will become the *server workstation*.
2. Create diskettes to be loaded in the other workstations in the workgroup. These will become *requester workstations*.

You must run the procedure to create diskettes (EHCDISTR) on the customization workstation. If you have installed only LANDP for OS/2 on your customization workstation, copy the LAN Support Program device drivers to the EHCCUS\EHCD500 directory before running the EHCDISTR procedure.

3. Distribute software to the server workstation as well as to the requester workstations.

   See “Distributing software to the workstations—GETDATA” on page 82.

Creating diskettes for the workstations—EHCDISTR

If you plan to boot from the diskette, you must format the diskette with the system files included, by using the /S option on the FORMAT command. This can only be done from a DOS workstation.

To create the diskette, from the EHCCUS directory, enter:

```
EHCDISTR parm1 WS=parm2 [parm3] [parm4] [parm5] [parm6] [parm7]
```

where:

parm1 Is a required parameter to specify the workgroup configuration.

The parameter format is:

```
langroup\lanname
```

Replace *langroup* with the name of the directory on which the workgroup configuration resides. If you omit *langroup*, EHCCUS is assumed.

Replace *lanname* with the name of the workgroup configuration.

parm2 Is a required parameter to specify the workstation in the workgroup that becomes the server workstation.

The parameter format is:

```
workstation name
```

Replace *workstation name* with the name of your workstation, for example `AA`.

parm3 Is an optional parameter to specify the highest severity of return codes allowed. The parameter value can be:

1 To admit only informative return codes. Higher severity results in execution ending.

2 To admit informative and attention return codes. Higher severity results in execution ending.

3 To admit informative, attention, and error return codes. Higher severity results in execution ending.

The default is 1. For further information, refer to “Return codes” on page 78.

parm4 Is an optional parameter to specify the type of diskette to be created. The parameter value can be:

`SVR` Diskette for the distribution workstation (server)

`REQ` Diskette for other workstations (requester)

The default is `REQ`.

parm5 Is an optional parameter to specify the target diskette drive. The parameter format is:

```
path
```

Replace *path* with the path you require. The default is `A:`.

parm6 Is a parameter to specify the *shared directory* to be used by the shared DOS directory server.

The parameter applies only when you create the diskette for the distribution workstation. In this case, it is required.

The parameter value must be the path on the distribution workstation where LANDP is installed. The parameter format is:

```
/S:path
```

Replace *path* with the path you require.

parm7 Is an optional parameter to specify a working directory. The CUSPARM.LST file is to be generated in the EHCCUS directory that is created under the path you specify.

The parameter format is:

```
/W:path
```

Replace *path* with the path you require. The default is the current directory.

Example:

```
EHCDISTR \Sample1 WS=AA
```

This command creates the distribution diskettes for the Sample1 workgroup configuration. Workstation AA will become the server workstation.

Only informative return codes are allowed; higher severity return codes result in execution ending.

Distributing software to the workstations—GETDATA

To distribute software to each workstation in the workgroup, insert the created diskette in the server or requester workstation and, if it applies, boot from that diskette.

Then, on the server workstation, run the AUTOFBSS.BAT procedure.

Finally, on the server and requester workstations, from the diskette drive, enter:

```
GETDATA parm1 parm2
```

where:

parm1 Is a required parameter to specify the workstation ID.

The parameter format is:

```
workstation name
```

Replace *workstation name* with the name of your workstation, for example AA. It must be specified in upper case.

parm2 Is a required parameter to specify the path where the files will be located. The parameter format is:

```
path
```

Replace *path* with the path you require.

Example:

```
GETDATA CC D:\LANDP
```

This command is run on workstation CC. It gets the workstation configuration data from a distribution diskette, and places it on the LANDP directory on the D: drive. The remaining files are copied from the server.

Modifying file contents

To meet your own requirements, you may need to modify some files created by the customization program, or files such as CONFIG.SYS on the production workstation.

Use the EHCADD utility program to add a new statement.

Note: EHCADD is supported only on OS/2 and DOS workstations.

The format of the command is:

```
  EHCADD path\nnnnnnnn.eee 'statement'
```

where:

path Is the path where the file to be modified is located.

nnnnnnnn.eee Is the name of the file (filename plus extension).

statement Is the statement to be added to the file.

The statement is added at the end of the file. If the file does not exist, the utility program creates it.

Example

```
EHCADD C:\CONFIG.SYS 'REM updated'
```

Use the EHCAPPEN utility program to append a new path to an existing statement.

Note: EHCAPPEN is supported only on OS/2 and DOS workstations.

The format of the command is:

```
  EHCAPPEN path\nnnnnnnn.eee 'identifier' 'entry'
```

where:

path Is the path where the file to be modified is located.

nnnnnnnn.eee Is the name of the file (filename plus extension).

identifier Identifies the statement that is to be modified.

The parameter value can be:

`PATH` To add the path at the end of a PATH or SET PATH statement

`LIBPATH` To add the path at the end of a LIBPATH statement

`DPATH` To add the path at the end of a SET DPATH statement

`HELP` To add the path at the end of a SET HELP statement

entry Is the path to be added to the statement.

If the path specified as the *entry* parameter already exists in the statement, the command is ignored. If the statement specified in the *identifier* parameter does not exist, the utility program creates it.

Example:

```
EHCAPPEN C:\CONFIG.SYS 'PATH' '.'
```

You can create a BAT file (DOS environment) or a CMD file (OS/2 environment) that contains EHCADD and EHCAPPEN statements. The EHCADD and EHCAPPEN utility programs are located in the EHCO000 directory.

To move a file from one directory to another, you can also include COPY statements, which copy files from one directory to another, and DEL statements, which delete files from a directory.

To start the getting procedure with the BAT file or the CMD file, you must include a GETTING statement as shown in the following example:

```
  GETTING WS=%1 %2
  COPY %2\PGM1.EXE X:\aaa\PGM1.EXE
  DEL %2\PGM1.EXE
  EHCADD C:\CONFIG.SYS ‘LASTDRIVE=G’
  EHCAPPEN C:\CONFIG.SYS ‘LIBPATH’ ‘C:\aaa\xxx;’
```

Chapter 10. Distributing software using a distribution server

OS/2 DOS NT

You can distribute files from an OS/2 distribution workstation to DOS, OS/2, and Windows NT workstations using a disk shared between the distribution and the individual workstations. Typically, the shared disk will be a LAN drive but you can use other shared mechanisms. You can then use a distribution server, such as NetView DM/2, to distribute the workstation files to each workstation. This is sometimes referred to as *distribution using CID*.

The task of distributing using CID is accomplished in two steps:

1. **Copy the workstation files onto the shared disk.**

   This step is performed using the EHCIMAGE procedure on the distribution workstation. The distribution workstation must be running OS/2 to perform this step.

   The EHCIMAGE procedure uses the run-time files generated by GENRUN, copies all the necessary files onto the shared disk, and generates response files for each individual workstation.

   For more information on GENRUN, see “Starting the run-time generation procedure—GENRUN” on page 74.

2. **Copy the workstation files from the shared disk.**

   This step is performed using the EHCCINST procedure and can run on DOS, OS/2, or Windows NT workstations.

   The EHCCINST procedure reads the response file for the workstation and copies the files for the individual workstation from the shared disk.

Selecting workgroup configuration

This section applies when you want to process more than one workgroup configuration at the same time.

The LANLIMIT.SPC file is located in the EHCCUS directory. You can edit the LANLIMIT.SPC file using any text editor, and specify the workgroup configurations to be processed.

- To *select* workgroup configurations, use INCLUDE statements. Those workgroup configurations not specified, explicitly or implicitly, are not processed. The format of the INCLUDE statement is:

  ```
  INCLUDE = langroup\lanname
  ```

- To *exclude* workgroup configurations, use OMIT statements. Those workgroup configurations not specified, explicitly or implicitly, are processed. The format of the OMIT statement is:

  ```
  OMIT = langroup\lanname
  ```

Replace *langroup* with the name of the directory on which the workgroup configuration resides. If you omit *langroup*, EHCCUS is assumed.

Replace *lanname* with the name of the workgroup configuration.

You cannot use both INCLUDE and OMIT statements at the same time.

Comments must start with /* and end with */.

The LANLIMIT.SPC file provided with the customization program has the following contents, and specifies that all the workgroup configurations will be processed.

```
INCLUDE = *
```

You can modify the LANLIMIT.SPC file to meet your requirements.

Examples

Example 1:

```
/* LANLIMIT.SPC Example 1*/
   INCLUDE = GROUP2\*
   INCLUDE = GROUP4\CONF47
```

Only the workgroup configurations located in the GROUP2 directory, and the workgroup configuration named CONF47 and located in the GROUP4 directory, will be processed.

Example 2:

```
/* LANLIMIT.SPC Example 2 */
   OMIT = GROUP6\*
```

All the workgroup configurations, except for those located in the GROUP6 directory, will be processed.

Example 3:

```
/* LANLIMIT.SPC Example 3*/
   INCLUDE = GROUP8\*
   INCLUDE = \CONF13
```

Only the workgroup configurations located in the GROUP8 directory, and the workgroup configuration named CONF13 and located in the EHCCUS directory, will be processed.

Copying files onto the shared disk—EHCIMAGE

You can display *online information* about the EHCIMAGE procedure. From the EHCCUS directory enter:

```
EHCIMAGE ?
```

To start copying the run-time files onto the shared disk, from the EHCCUS directory, enter:

```
EHCIMAGE [parm1] parm2 parm3
```

where:

parm1 Is an optional parameter to specify the workgroup to be processed. It applies when you wish to copy only one workgroup to the shared disk.

If the parameter is used, only the workgroup specified will be copied, regardless of the contents of the LANLIMIT.SPC file.

The parameter format is:

```
langroup\lanname
```

Replace *langroup* with the name of the directory on which the workgroup configuration resides. If you omit *langroup*, EHCCUS is assumed.

Replace *lanname* with the name of the workgroup configuration.

parm2 Is a required parameter to specify the path where the directory structure for the LANDP program files will be kept.

parm3 Is a required parameter to specify the path where the directory containing the response files for each workstation in the workgroups will be placed. One response file is generated for each workstation in every workgroup.

The names of the response files created include the NetView/DM workstation name, as specified by the WSID keyword on the LWSCONF vector for the workstation. The default value for this name comprises the first six characters of the workgroup name and the two character workstation name.

Example

```
EHCIMAGE \DJV W:\DJV\IMG\EHC W:\DJV\RSP\EHC
```

This command copies files from the LANDP installation directory to a directory tree that is similar to:

```
W:\DJV\IMG\EHC                 - Root directory for this CID directory tree
W:\DJV\IMG\EHC\EHCCUS          - Customization root directory
W:\DJV\IMG\EHC\EHCCUS\DJV      - Root directory for the DJV workgroup
W:\DJV\IMG\EHC\EHCCUS\DJV\AA   - Information for the AA workstation
W:\DJV\IMG\EHC\EHCCUS\DJV\BB   - Information for the BB workstation
W:\DJV\IMG\EHC\EHCCUS\DJV\CC   - Information for the CC workstation
W:\DJV\IMG\EHC\EHCCUS\EHCD100  - Customization files for DOS
W:\DJV\IMG\EHC\EHCCUS\EHCO100  - Customization files for OS/2
W:\DJV\IMG\EHC\EHCD500         - Common files for LANDP for DOS
W:\DJV\IMG\EHC\EHCO500         - Common files for LANDP for OS/2
W:\DJV\IMG\EHC\EHCN500         - Common files for LANDP for Windows NT
```

This might create response files with the following names:

```
W:\DJV\RSP\EHC\DJVAA.RSP       - Response file for workstation AA
W:\DJV\RSP\EHC\DJVBB.RSP       - Response file for workstation BB
W:\DJV\RSP\EHC\MYFILE.RSP      - Response file for workstation CC
                                 ('MYFILE' specified on the WSID on the
                                 LWSCONF vector for this workstation)
```

EHCIMAGE checks that all the response files that are generated have unique names before performing the copy operation.

Note: NetView DM/2 can only handle response file names of 6 characters or less. However, because the default response file created by EHCIMAGE uses the LAN name suffixed by the workstation name, the response file name can be greater than 6 characters. If this is the case, EHCIMAGE truncates the response file name after 6 characters, and a duplicate response may result.

It is recommended that you only use the default response file names created by EHCIMAGE, since this reduces user error situations where the same WSID is specified by different workgroups.

If you specify the WSID vector in a LANCONF.SPC, it overrides the default name created by EHCIMAGE only for that workstation.

Copying the workstation files from the shared disk—EHCCINST

You can display *online information* about the EHCCINST procedure. From the EHCCUS directory, enter:

```
EHCCINST ?
```

To start the EHCCINST procedure, from the distribution directory, enter:

```
EHCCINST [/S:parm1] /T:parm2 /R:parm3 /L1:parm4 /L2:parm5
```

where:

parm1 Is an optional parameter to specify the path where the files to be distributed are kept. The default is **.** (current directory), since EHCCINST is copied to the top of the distribution directory structure by EHCIMAGE.

parm2 Is a required parameter to specify the path where the files are to be copied.

parm3 Is a required parameter to specify the response file to be used for distribution to this workstation.

parm4 Is a required parameter to specify the location of the error log file. If you specify only a path (indicated by use of a trailing backslash on this parameter), the error log file is placed in that directory which has a filename matching that of the response file, and an extension of `L1`.

parm5 Is a required parameter to specify the location of the history log file. If you specify only a path (indicated by use of a trailing backslash on this parameter), the error log file is placed in that directory which has a filename matching that of the response file, and an extension of `L2`.

Example

```
EHCCINST /S:X:\DJV\IMG\EHC /T:D:\EHC /R:X:\DJV\RSP\EHC\MYFILE.RSP
         /L1:W:\ERRS\MYFILE.ERR /L2:W:\HIST
```

This command installs from the CID image directory X:\DJV\IMG\EHC, using the response file X:\DJV\RSP\EHC\MYFILE.RSP. The error log file is W:\ERRS\MYFILE.ERR and the history log file is W:\HIST\MYFILE.L2.

EHCCINST does not update any system files (such as CONFIG.SYS) on the individual workstation. If such an update is required, you can update the response file for the workstation to invoke a user exit program.

EHCCINST response files

Although response files are generated by EHCIMAGE, you may want to perform additional tasks before or after EHCCINST transfers the LANDP files to the individual workstations. This can be done by enhancing the response file provided by EHCIMAGE.

The response file read by EHCCINST can contain records which use the following keywords. They are processed in the order they are encountered.

COPY This keyword takes two values, a source and a target. The source file is copied to the target file specification.

INCLUDE This keyword takes one value, the file specification for another response file. This response is read and acted upon.

USEREXIT This keyword takes one value, the file specification of a user exit to be invoked. The user exit program is passed one value, the target directory.

LANDP.GETTING
This keyword takes two values, the path of the directory used on the distribution machine to hold the run time files, and the file specification of the GETTING.SPC file for the individual workstation. When the LANDP.GETTING keyword is processed, the run-time files for the individual workstation are copied into the target path.

The file specifications used by the COPY, INCLUDE, and USEREXIT keywords can start with one of the following substitution variables:

`$(Sourcedrive)` The drive where the source files can be found.
`$(Sourcedir)` The source directory specified on EHCCINST.
`$(Targetdrive)` The target drive to which the files will be copied.
`$(Targetdir)` The target directory to which the files will be copied.
`$(Responsedrive)`
The drive where the response file can be found.
`$(Responsedir)` The directory where the response file can be found.

Example

```
USEREXIT=$(Sourcedir)\EXITS\PREPWS.CMD
```

Using response files to update CONFIG.SYS

Together with a user exit program, you can use user-written response files to update the CONFIG.SYS file for individual workstations. To do this, create new response files for each workstation of the form:

```
USEREXIT=W:\USEREXIT\PREPWS.CMD
INCLUDE=W:\DJV\RSP\EHC\DJVAA.RSP
USEREXIT=W:\USEREXIT\FINISHWS.CMD
REBOOT=YES
```

The first user exit prepares the workstation for run-time files by ensuring that a copy of the workstation's original CONFIG.SYS has been saved.

The include statement then includes the response file created by EHCIMAGE. This allows EHCIMAGE to be rerun at a later date without losing the new responses that specify the user exits.

The final user exit then appends the CONFIG.ADD file created for the workstation to the original CONFIG.SYS and saves that as the workstation's new CONFIG.SYS.

An example of the first user exit program, PREPWS.CMD, is:

```
@MKDIR C:\ORIGINAL
@IF NOT EXIST C:\ORIGINAL\CONFIG.SYS COPY C:\CONFIG.SYS C:\ORIGINAL
```

An example of the second user exit program, FINISHWS.CMD, is:

```
/* In REXX, since we need to be able to give a return code */

/* Create the CONFIG.SYS from the original file and */
/* the additions from LANDP                         */
"@COPY C:\ORIGINAL\CONFIG.SYS+%1\CONFIG.ADD C:\CONFIG.SYS"

/* Return to EHCCINST indicating that a reboot is required. */
exit x2d("FE 00")
```

Sample NetView DM change file profile

A sample change file profile for NetView DM Change Distribution Manager (CDM) is shown below:

```
TargetDir = D:\EHC

Section Catalog
Begin
   GlobalName  = LANDP2.DISTRIBUTION.REF.5.0
   ObjectType  = SOFTWARE
   Description = 'LANDP Version 5.0 Distribution files'
End

Section Install
Begin
   Program             = $(SourceDir)\EHCCINST.EXE
   Parms               = /S:$(SourceDir) /T:$(TargetDir)
                         /R:$(ResponseFile) /L1:$(LogFile1)
                         /L2:$(LogFile2)
   SourceDir           = SA:\IMG\EHC
   WorkingDir          = $(TargetDir)
   ResponseFile        = SA:\RSP\EHC\$(WorkStatName).RSP
   GeneralResponsePath = SA:\RSP\EHC\$(WorkStatName).RSP
   Logfile1            = SB:\ERRS\$(WorkStatName).ERR
   Logfile2            = SB:\HIST\
End
```

It is best to use variables as much as possible, as this allows you to have just one change profile for all workstations in your workgroup. The $(WorkStatName) is the NetView DM workstation name, and may not be the same as your LANDP workstation name.

It is helpful to create your NetView DM workstations with the same name as your LANDP workstation names, remembering to prefix the name with the workgroup name.

Part 5. Preparing LANDP platforms

LANDP can run on five platforms:

- DOS Version 7.1
- OS/2 Warp Version 4.0 (or higher)
- Microsoft Windows NT Version 4.0, or Windows 2000
- AIX Version 3.2, running on an RS/6000® system

This part describes how to prepare these five platforms for the installation of LANDP. It contains the following chapters:

Chapter 11. Preparing DOS workstations

The first part of the chapter, “Installing and configuring IBM DOS workstations,” describes the requirements to install the LANDP for DOS run-time files on the DOS workstations. Some requirements are related to a specific server.

The second part of the chapter, “Modifying run-time files” on page 100, describes the process of modifying the run-time files created by the customization program, according to your needs.

The third part of the chapter, “Installing run-time files” on page 125, describes how to check for a proper installation of the run-time files.

Installing and configuring IBM DOS workstations

If the diskettes created for the DOS workstations do not contain the DOS system files, make sure that IBM DOS V7.1 is installed according to the existing recommendations, before installing the LANDP for DOS operational diskettes into your DOS workstation.

When working in DBCS mode, the operating system can be one of the following, or later versions, depending on your national language:

Language	**Operating system**	**Supported codepage**
Traditional Chinese	IBM DOS T2000	938, 950
Korean	IBM DOS H2000	949
Simplified Chinese	IBM DOS P2000	1381

Installation requirements for NetBIOS transport protocol

This section applies to LANDP for DOS workstations integrated in a LANDP workgroup that uses NetBIOS as the transport protocol.

To use NetBIOS as the internal communications protocol for a LANDP workgroup, a NetBIOS implementation must be installed and configured in all the workstations. This can be the IBM LAN Support Program, IBM LAN Client, or another compatible implementation.

If you select NetBIOS as the transport protocol, LANDP customization assumes that you are using the IBM LAN Support Program for a non-Network Driver Interface Specifications (NDIS) device driver. It generates a CONFIG.SYS file containing statements to load the NetBIOS device drivers. The software distribution procedure (GETTING) copies these device drivers to the distribution media. However, if you are not using the IBM LAN Support Program for a non-NDIS device driver, or if there are programs other than the LANDP NetBIOS transport using NetBIOS, you will have to configure the NetBIOS support directly.

NetBIOS resources

Before configuring some NetBIOS implementations you need to know what NetBIOS resources will be required. The NetBIOS resources to note are NAMES, SESSIONS, and COMMANDS. Each program that uses NetBIOS will require a number of these. Find out the programs that use NetBIOS and the resource that they require and add these resource requirements together to find the total number of NAMES, SESSIONS, and COMMANDS required for each workstation in the workgroup.

You should take into account:

The LANDP NetBIOS transport

This requires 1 NAME, n SESSIONS and n+3 COMMANDS, where n is the number of other workstations that the subject workstation is configured to communicate with. Look on the DXMT0MOD line in the generated CONFIG.SYS file and you will see S, the number of sessions, and C, the number of commands.

Servers for NetBIOS attached devices

If the subject workstation is to run any of DTAU4733, PT4721, SP4721##, SS#####, or SS###### servers, additional NetBIOS resources will be required. LANDP customization adds 1 SESSION and 2 COMMANDS to the values in the DXMT0MOD line in the generated CONFIG.SYS file if any of these servers are configured. However, more resources may be required if more than one of them is configured, or if more than one device is assigned. Refer to the documentation of these servers for more details.

IBM LAN Server and other products

Refer to the documentation of these products for more details.

IBM LAN Support program

When using non-NDIS adapters the statements in the generated CONFIG.SYS file are usually adequate. If n1 NAMES, n2 SESSIONS and n3 COMMANDS are required, the DXMT0MOD line should be:

```
DEVICE=DXMT0MOD N=n1 ST=n2 S=n2 C=n3
```

When using an NDIS adapter there is a choice of drivers depending upon whether or not the IEEE 802.2 interface is required in addition to the NetBIOS interface. The LANDP TRDLC server requires the IEEE 802.2 interface.

DXME0MOD

If both interfaces are required the DXME0MOD driver must be used. It is easiest to configure this by changing DXMC0MOD to DXME0MOD in the customization generated CONFIG.SYS file. In addition O=N must be specified on the DXMT0MOD line, giving the following CONFIG.SYS file lines for NetBIOS:

```
DEVICE=DXMA0MOD
DEVICE=DXME0MOD
DEVICE=DXMT0MOD N=n1 ST=n2 S=n2 C=n3 O=N
```

DXMJ0MOD

If only the NetBIOS interface is required then you may use DXMJ0MOD. This replaces both the DXMC0MOD and DXMT0MOD lines in the generated CONFIG.SYS file. The lines for NetBIOS are as follows:

```
DEVICE=DXMA0MOD
DEVICE=DXMJ0MOD
```

It has no parameters as NetBIOS resource information is contained in the PROTOCOL.INI file. For more information, see “PROTOCOL.INI contents” on page 107.

For either of these device drivers LANDP customization will not generate the correct software distribution statements in the GETTING.SPC files. You can either alter these or install and configure the IBM LAN Support Program separately on each workstation.

When using the device drivers for PC Network Adapters (DXMGnMOD) or NDIS adapters (DXME0MOD) the workspace may not be large enough if there are a large number of sessions or a large volume of data traffic. A message may be generated, the adapter may fail to open, or the workstation may simply hang at start-up. The workspace is the second positional parameter for these drivers. For example, to set it to the maximum of 64K, change the DXMG0MOD line to:

```
DEVICE=DXMG0MOD ,64
```

See the IBM LAN Support Program User's Guide for more details.

IBM LAN Client

LANDP customization will not generate the correct software distribution statements in the GETTING.SPC files. You can either alter these or install and configure the IBM LAN Client separately on each workstation.

Discard the IBM LAN Support Program configuration statements in the generated CONFIG.SYS file and install IBM LAN Client in the required workstations. There is no requirement to enter NetBIOS resources.

Installation requirements for TCP/IP transport protocol

This section applies to LANDP for DOS workstations integrated in a LANDP workgroup that uses TCP/IP as transport protocol.

To use TCP/IP as the internal communications protocol for a LANDP workgroup, TCP/IP must be installed and configured in all the workstations.

LANDP operates with either IBM TCP/IP for DOS version 2.1.1.4 or with PC/TCP® Network Software version 5.0 from NetManage Inc.®.

PC/TCP

The datagram length used by LANDP must not exceed the PC/TCP maximum segment size (MSS). The MSS values for different types of network are listed in the *PC/TCP Advanced Users Guide*. The datagram length used by LANDP can be set by the /D parameter of the LANDP Internet Protocol (LIP) LOADER statement described in “LANDP Internet Protocol” on page 111.

When large volumes of data are being transmitted, the PC/TCP kernel may require additional packet buffers for optimal performance. See the *PC/TCP Advanced Users Guide* for instructions on adjusting the number of packet buffers.

For more information about the PC/TCP transport protocol, see Appendix E, “Using TCP/IP for internal communication” on page 525.

The default configuration of PC/TCP has 4 TCP connections and 2 UDP connections. This may not be sufficient when using the TCP/IP wide area communications server. The connections required are:

udp-connections
The LANDP Internet Protocol requires 1 UDP connection and the TCP/IP wide area communications server requires 2 UDP connections, so if both are used add:

```
udp-connections=3
```

to the [PCTCP KERNEL] section of the PCTCP.INI file.

tcp-connections
The TCP/IP wide area communications server requires the following numbers of TCP connections:

Listening sessions.
One TCP connection is required for every listening session.

TELNET sessions.
One TCP connection is required for every TELNET session.

MPTN (LU6.2) sessions.
One TCP connection is required for every LU-LU combination in session, plus one TCP connection for every session between the LUs.

MPTN (LU0/1/2) sessions.
Two TCP connections are required for every dependent LU server (DLUS) in use plus one TCP connection for every dependent LU session.

To change the number of TCP connections add:

```
tcp-connections=number
```

to the [PCTCP KERNEL] section of the PCTCP.INI file.

When the TCP/IP wide area communications server is being used for MPTN (AnyNet) sessions, it is necessary to set the TCP window size parameter of PC/TCP to a value of at least 254 bytes more than the maximum RU size sent by the primary LU. The PC/TCP documentation recommends setting this parameter to a multiple of the maximum segment size of the network interface.

To set the TCP window size, where *value* is the required size, add:

```
window=value
```

to the [PCTCP KERNEL] section of the PCTCP.INI file.

IBM TCP/IP

To prepare TCP/IP software for coexisting with LANDP, update the `IntExclude` parameter in the TCPDOS.INI file, located in the ETC subdirectory in the path where TCP/IP is installed. The default specifies:

```
IntExclude = 61-63, 67
```

That means that TCP/IP will not use interrupts 61, 62, 63, and 67. Because interrupt 64 is used by LANDP, to avoid unpredictable results, specify also 64:

```
IntExclude = 61-64, 67
```

For detailed information, refer to the TCP/IP manuals. See “Bibliography” on page 571.

For more information about the TCP/IP transport protocol, see Appendix E, “Using TCP/IP for internal communication” on page 525.

Installation requirements for TCP/IP wide area communications server

LANDP for DOS workstations configured to use the TCP/IP wide area communications server must have PC/TCP installed. see details, see “Installation requirements for TCP/IP transport protocol” on page 97.

There must also be an EHCTCP.INI file to provide the information required to map LANDP sessions with SNA and PPC servers to TCP/IP protocols, ports and internet addresses. This file can be created with any text editor, but it is best practice to use the file as generated by LANDP customisation. See SES&TCP on page 477 for a description of the file contents.

SNA over TCP/IP dependent LUs

When using the TCP/IP wide area communications server the dependent LU names must have a common alphanumeric prefix and a numeric suffix with a constant difference between the suffix and the LOCADDR. For example:

```
IYCKT300 LU LOCADDR=2
IYCKT301 LU LOCADDR=3
IYCKT327 LU LOCADDR=29
IYCKT481 LU LOCADDR=183
```

Installation requirements for TRDLC server

This section applies to LANDP for DOS workstations configured to use the TRDLC server for SNA communications.

To use the TRDLC server on a workstation an IEEE 802.2 interface to the LAN adapter must be installed and configured on the workstation. This can be provided by IBM LAN Support Program, IBM LAN Client, or another compatible implementation.

If you select NetBIOS as the transport protocol and specify the TRDLC server, LANDP customization assumes that you are using IBM LAN Support Program for a non-NDIS device driver. It generates a CONFIG.SYS file containing statements to load the IEEE 802.2 device drivers. The software distribution procedure (GETTING) copies these device drivers to the distribution media. However, if you're not using IBM LAN Support Program for a non-NDIS device driver, then you will have to configure the NetBIOS support directly.

IBM LAN Support program

When using non-NDIS adapters the statements in the generated CONFIG.SYS file should be correct. For example:

```
DEVICE=DXMA0MOD
DEVICE=DXMC0MOD 4000xxxxxxxx
DEVICE=DXMT0MOD N=n1 ST=n2 S=n2 C=n3 ES=1 EST=1 DS=600
```

Where 4000xxxxxxxx is the locally administered address for the adapter and 'ES=1 EST=1 DS=600' are additional required parameters for the NetBIOS driver when using the IEEE 802.2 interface.

When using an NDIS adapter simply change DXMC0MOD to DXME0MOD in the customization generated CONFIG.SYS file. In addition O=N must be specified on the DXMT0MOD line, giving the following CONFIG.SYS file lines:

```
DEVICE=DXMA0MOD
DEVICE=DXME0MOD 4000xxxxxxxx
DEVICE=DXMT0MOD N=n1 ST=n2 S=n2 C=n3 O=N ES=1 EST=1 DS=600
```

IBM LAN Client

Discard the IBM LAN Support Program configuration statements in the generated CONFIG.SYS file.

During IBM LAN Client installation make sure that the IEEE 802.2 interface is configured and that the locally administered address is set to the required value.

TCP/IP and TRDLC

LANDP customization will not permit the configuration of the TCP/IP transport together with the TRDLC server because it is not possible to run them both through the same non-NDIS adapter with IBM LAN Support program. However, with multiple adapters or with an NDIS adapter it is possible to use them together. To get around the customization restriction, customize LANDP to use the NetBIOS transport. Substitute EHCLIP.EXE for LAN.EXE in the distributed files and in the loading statement in AUTOFBSS.BAT. To arrive at the correct adapter driver configuration see the IBM TCP/IP Version 2.1.1.4 for DOS Installation and Administration manual.

Modifying run-time files

The main areas of modification are:

- Emulator considerations
- Windows 3.1 considerations
- CONFIG.SYS contents

- AUTOFBSS.BAT and AUTOUSER.BAT contents

For a utility program to modify files, see “Modifying file contents” on page 83.

Depending on the applications and servers you have developed and the devices you have installed, additional files may need to be copied.

Emulator considerations

There are some considerations for the CONFIG.SYS and AUTOFBSS.BAT files for the emulators.

In accordance with the keyboard and display code page selected at emulator customization time, additions may be needed in the CONFIG.SYS and AUTOFBSS.BAT files. If these are not included, you may get unexpected results.

Note: The following commands do not apply to DBCS mode.

To CONFIG.SYS, add the following DOS commands for the 3270 emulator:

```
COUNTRY=xxx,yyy,COUNTRY.SYS
DEVICE=DISPLAY.SYS CON:=(EGA,yyy,1)
```

To CONFIG.SYS, add the following DOS commands for the 3287 printer emulator:

```
COUNTRY=xxx,yyy,COUNTRY.SYS
DEVICE=PRINTER.SYS LPT1:=(4201,yyy,1)
```

To AUTOFBSS.BAT, add the following DOS commands for the 3270 emulator:

```
NLSFUNC
KEYB zz,yyy,KEYBOARD.SYS
MODE CON: CP PREP=((yyy) EGA.CPI)
CHCP 850  or  MODE CON CP SELECT=yyy
```

To AUTOFBSS.BAT, add the following DOS commands for the 3287 printer emulator:

```
NLSFUNC
KEYB zz,yyy,KEYBOARD.SYS
MODE LPT1: CP PREP=((yyy) 4201.CPI)
CHCP 850  or  MODE CON CP SELECT=yyy
```

where:

xxx Is the country code.
yyy Is the code page.
zz Is the keyboard code.

For more information about these commands, see the *DOS Reference Guide*.

Microsoft Windows 3.1 or 3.11 considerations

If Microsoft Windows 3.1 or 3.11 support was specified at customization time, the following files will be distributed to the workstation:

EHCPREV.BAT
WINSTART.BAT
EHCDOSVM.BAT
EHCAPLVM.BAT
SYSTEM.ADD
EHCDOSVM.PIF

The **EHCPREV.BAT** procedure should be run before starting Windows 3.1/3.11.

The first statement in the file must be:

```
EHCWGMDI /L:xx
```

where *xx* is an optional parameter to specify the maximum storage, in KB, required by Windows parameters and user data. The parameter value ranges from 4 to 56. The default is 4.

If the following servers were specified at customization time, the corresponding loading statements will appear after the EHCWGMDI statement:

- Token-ring data link control server
- Synchronous data link control server
- Financial printer server
- 4748 printer server

For information on those loading statements, refer to “Loading statements for LANDP for DOS servers” on page 109.

The **WINSTART.BAT** file contains the following statement:

```
EHCWVDMI
```

EHCWVDMI is accessed by the Windows 3.1/3.11 applications.

The **EHCDOSVM.BAT** procedure should be run under Windows 3.1/3.11 in a virtual machine.

The EHCDOSVM.BAT file contains the loading statements of the supervisor and the servers that are to reside on the workstation, except for those whose loading statement should be placed in the EHCPREV.BAT file.

Ensure that any modification of this file does not cause a server to be loaded after the supervisor. For information on the loading statements, refer to “Loading statements for LANDP for DOS servers” on page 109.

The **EHCAPLVM.BAT** procedure should be run under Windows 3.1/3.11 in the virtual machine where the LANDP application will be loaded. The default contents are:

```
EHCWVDMI
AUTOUSER
```

The **SYSTEM.ADD** file should be added to the SYSTEM.INI Windows 3.1/3.11 setting file. The default contents are:

```
[386Enh]
DEVICE=EHCVMSD.386
```

The **EHCDOSVM.PIF** file can be edited with the Windows 3.1/3.11 PIF editor. You can modify the contents according to the settings to be assigned to the virtual machines where LANDP or LANDP applications run.

CONFIG.SYS contents

You should compare the CONFIG.SYS file created by the customization program with the CONFIG.SYS file in your workstation, and make the necessary modifications to merge them into one.

You should specify the paths for the new device drivers and other needed files.

You should also take into account the following:

- The number of files (data + indexes) created by the shared-file server for the electronic journal and store-for-forwarding servers, or your applications.

  An *approximate* value is already set for the FILES parameter.

- Usage of hard disks greater than 32MB, when many files are in use.

  You must use the SHARE program. The default parameters may not be sufficient. In this case, load the SHARE program with the CONFIG.SYS file, by adding the following statement:

  ```
  INSTALL d:\path\share.exe /F:xxxx
  ```

  where *xxxx* is the value, in bytes, you need for file sharing. The default is 2048.

  The SHARE program is mandatory to have file locking available.

- Performance.

  You should add a CACHE program. In the workstations where you use a data management server such as the shared-file server, set **lazy writes** off, unless your system is equipped with an Uninterruptible Power Supply (UPS).

- Shared-file server usage.

  When the shared-file server starts up, it creates the necessary shared-file directories if they do not already exist. This removes the need (which existed in previous releases) to call CREADB.BAT to create these directories.

- Windows 3.1/3.11 considerations.

  You do not require the LAN Support Program (LSP) drivers, DXMA0MOD, DXMC0MOD, and DXMT0MOD, in your CONFIG.SYS file if you are running under Windows for Workgroups 3.11. They are required for Windows 3.1.

The following is an example of a CONFIG.SYS file for a workstation that contains the SNA server with X.25 data link control, the shared-file server, and a 3270 emulator.

The workstation is integrated in a LANDP workgroup that uses NetBIOS as transport protocol.

```
BUFFERS=20
FILES=40
SHELL = \COMMAND.COM /E:256 /P
REM             The environment parameter above may
REM             need to be increased based on other
REM             than LANDP for DOS program requirements.
DEVICE=DXMAOMOD.SYS
DEVICE=DXMCOMOD.SYS
DEVICE=DXMTOMOD.SYS  ST=26  S=26  C=30
DEVICE=X251.SYS
```

Magnetic stripe reader/encoder server

If you plan to load this server, the following statements are required in the CONFIG.SYS file.

If the server supports a 4717 MSR/E:

```
DEVICE=MSRE2DD.SYS /X
```

If the server supports a 4777 MSR/E:

```
DEVICE=IBM4777.SYS /X /Cp
```

The *p* parameter value corresponds to the COM port where the 4777 MSR/E will be attached.

If the server supports a 4778 MSR:

```
DEVICE=IBM4777.SYS /X /Cp /M
```

The *p* parameter value corresponds to the COM port where the 4778 PIN Pad MSR will be attached.

If a 4778 device is mouse attached:

```
DEVICE=MSRE2DD.SYS /M
```

If a 4777 device is mouse attached:

```
DEVICE=MSRE2DD.SYS
```

PIN pad server

If you plan to load this server, the following statements are required in the CONFIG.SYS file.

If the server supports a 4718 PIN pad:

```
DEVICE=PIN2DD.SYS /X
```

If the server supports a 4778 PIN pad:

```
DEVICE=IBM4778P.SYS /X /Cp
```

The *p* parameter value corresponds to the COM port where the 4778 PIN Pad MSR will be attached. The second statement should be specified only if MSR capabilities will be used.

If a 4778 device is mouse attached:

```
DEVICE=PIN2DD.SYS
```

TRDLC server

If you plan to load this server, refer to "Installation requirements for TRDLC server" on page 99.

X.25 DLC server for IBM PC X.25 communications adapter

If you plan to load this server, the following statement is required in the CONFIG.SYS file:

```
DEVICE=<drive> <path> xxxxxx.SYS
```

where:

drive Is the drive on which the device is located.

path Is the directory on which the device is located.

xxxxxx
Is the name of the device driver. This can be:

X25DD2 X.25 Communications Adapter uses interrupt level 2.
X251 X.25 Communications Adapter uses interrupt level 3.
X25DD4 X.25 Communications Adapter uses interrupt level 4.

Financial printer server

If you plan to load this server, for each printer connected to an RS-232 port, the following statement is required in the CONFIG.SYS file:

```
DEVICE=<drive> <path>FPRTxxxx.SYS [/B:bbbb][/A:a] [/R:rrr] [/X] [/M]
```

where:

drive Is the drive on which the device is located.

path Is the directory on which the device is located.

FPRT*xxxx*SYS
Is the driver file name. For a list of file names, refer to Table 1 on page 22.

bbbb The desired baud rate (in bits per second). Permissible values are: 9600, 4800, 2400, 1200, 0600, 0300, 0150. The default value is 9600 bps.

a The logical connector address of the printer. Permissible values are:

1 Asynchronous connector 1
2 Asynchronous connector 2
3 Asynchronous connector 3
4 Asynchronous connector 4

The default value is Asynchronous connector 1.

rrr The redirection assignment for parallel connectors 1, 2, and 3. The redirection of the connectors is as follows:

- 1 - Connector 1 - 100
- 2 - Connector 2 - 010
- 3 - Connector 3 - 001

All combinations 000 through 111 are valid. The default value is 000.

/X This option suppresses error messages to the screen.

/M To be used if REMS is used.

The same device drivers as those for the 4722 printer are used if you plan to use the 9055 Model 2 or 9068-S01 printers.

4748 printer server

If you plan to load this server, for each 4748 printer connected to an RS-232 port, the following statement is required in the CONFIG.SYS file:

```
DEVICE=<drive> <path>FPRTxxxx.SYS [/B:bbbb] [/A:a] [/X]
```

where:

drive Is the drive on which the device is located.

path Is the directory on which the device is located.

FPRT*xxxx*.SYS
Is the driver file name. For a list of file names, refer to Table 3 on page 23.

bbbb The desired baud rate (in bits per second). Permissible values are: 9600, 4800, 2400, 1200, 0600, 0300, 0150. The default value is 9600 bps.

a The logical connector address of the printer. Permissible values are:

1 Asynchronous connector 1
2 Asynchronous connector 2

The default value is Asynchronous connector 1.

/X This option suppresses error messages to the screen.

If you plan to use the 9055 Model 1 or 9068-D01 printer, you can use the same device driver as for the 4748 printer (but it will provide support only for 4748 emulation mode). If the extra features of these printers are to be used (for example, REMS), change the device driver name in your CONFIG.SYS file to:

```
DEVICE=<drive> <path>FPRTSCPx.SYS [/B:bbbb] [/X]
```

4733 teller assist unit

If you plan to use a 4733 teller assist unit, you must specify the following statement in your CONFIG.SYS file:

```
DEVICE=TCD386D.SYS
```

PROTOCOL.INI contents

If your workstation is using LAN Support Program device driver DXMJ0MOD.SYS, which requires LAN Support Program Version 1.38 or later, you need to update your PROTOCOL.INI file to include details that the LANDP configuration added to the CONFIG.SYS file. These include:

- Sessions (S)

  LANDP Customization will have added a sessions parameter to the DXMT0MOD.SYS device driver. You should use this value to set the SESSIONS and DATAGRAMPACKETS keywords in the DXMJ0MOD_MOD section of PROTOCOL.INI.

- Commands (C)

  LANDP Customization will have added a commands parameter to the DXMT0MOD.SYS device driver. You should use this value to set the NCBS keyword in the DXMJ0MOD_MOD section of PROTOCOL.INI.

For example, LANDP customization may have generated a CONFIG.SYS file that includes the lines:

```
DEVICE=DXMC0MOD.SYS
DEVICE=DXMT0MOD.SYS O=N ST=3 S=3 C=6
```

If, for example, you are using the IBMTOK.DOS MAC driver, PROTOCOL.INI would look something like this:

```
[DXMJ0MOD_NIF]
   Driver Name=NETBEUI$
   Bindings=...
   NCBS=6
   SESSIONS=3
   DATAGRAMPACKETS=6
```

⋮

Note: DATAGRAMPACKETS should be set to at least double the number of sessions. The maximum value for DATAGRAMPACKETS is 140, so if you have more than 70 sessions, the value should be set to 140.

AUTOFBSS.BAT and AUTOUSER.BAT contents

Note: This section does not apply if you use Windows 3.1/3.11 support. If you do use Windows 3.1/3.11, refer to “Microsoft Windows 3.1 or 3.11 considerations” on page 101.

The AUTOFBSS.BAT file contains the loading statements of the supervisor and the servers that are to reside on the workstation. The last program loaded is the supervisor.

Ensure that any modification of this file does not cause a server to be loaded after the supervisor. Application loading statements should not be placed in the AUTOFBSS.BAT file but in the AUTOUSER.BAT file.

If DOSKEY is used as well as LANDP, execute DOSKEY before AUTOFBSS to ensure that the settings are saved.

The following is an example of a AUTOFBSS.BAT file for a workstation that contains the SNA server with X.25 data link control, the shared-file server, and a 3270 emulator. The workstation is integrated in a LANDP workgroup that uses NetBIOS as transport protocol.

```
NEWCFG
IF ERRORLEVEL 1  GOTO END
VARDAT
IF ERRORLEVEL 1  GOTO END
LOADER SHFILE##.EXE /C:SFILE1 /B:12 /E /S:3 /L:2 /F:50
IF ERRORLEVEL 1  GOTO END
LOADER SNA##.EXE
IF ERRORLEVEL 1  GOTO END
XL.EXE
IF ERRORLEVEL 1  GOTO END
LOADER X25DLC.EXE
IF ERRORLEVEL 1  GOTO END
LOADER EMU3270.EXE /C:ATR/K:KBD/D:DIS/I:1
IF ERRORLEVEL 1  GOTO END
LOADER DDT.EXE
IF ERRORLEVEL 1  GOTO END
LOADER LAN.EXE/AB
IF ERRORLEVEL 1  GOTO END
LOADER SPV.EXE/AB /K:20
IF ERRORLEVEL 1  GOTO END
AUTOUSER
:END
FREE
```

If the workstation is integrated in a LANDP workgroup that uses TCP/IP as transport protocol, EHCLIP.EXE should be specified instead of LAN.EXE.

In order to get an information window about LANDP for DOS status, you should invoke the EHCCINFO utility program.

You should modify the AUTOUSER.BAT file to meet the requirements of your applications. They can be started with AUTOUSER.BAT.

Printer manager server

If you plan to use DOS PRINT together with this server and either the 3287 printer emulator, or the financial printer server, or your own printer server, the following statement is required in the AUTOFBSS.BAT file:

```
FBSSP <path>\PRINT<parameters>
```

where `FBSSP` is the loading command. You should provide the *path* and *parameters*.

Loading statements for LANDP for DOS servers

Most loading statements explained in this section are automatically created by the customization program, using the parameters you provided. LOADER is used by LANDP for DOS to load the functional areas into workstation memory.

For return codes during loading, refer to *LANDP Problem Determination*.

After loading LANDP, the functional areas become an extension of DOS. Control is transferred to the servers when loading is completed. Each server performs initialization, then goes into a wait state. Invocation is subsequently done by the supervisor as needed.

Some functional areas can be loaded into expanded memory, thus not occupying space in conventional memory. The loading command then is changed to LOADERE.

Both loading commands, LOADER and LOADERE, provide two optional parameters, `w:e` and `&:e`. These parameters can be included in the loading statements of some user servers, as follows:

```
LOADER (or LOADERE) [/w:e] [servername]
```

where *servername* is the name of the server being loaded.

The `w:e` parameter applies to user servers that do not call DOS functions at run-time, and therefore do not need specific internal storage for that purpose.

If the parameter is specified, the LOADER(E) program uses less memory. If specified, but the server uses DOS functions, the results are unpredictable.

```
LOADER (or LOADERE)  [/&:e] [servername]
```

where *servername* is the name of the server being loaded.

The `&:e` parameter applies to user servers that do not process connection functions (“&&”) and disconnection functions (“**”) corresponding to a specific process, and therefore do not need to be involved in the general process for that purpose. Both parameters, `w:e` and `&:e`, can be included in the same loading statement.

If the `&:e` parameter is specified in the loading statement of the supervisor, the applications calling the Wait for Asynchronous Events (WM) supervisor function do not receive “&&” and “**” events.

LANDP for DOS functional areas can be loaded into high memory. Add the LOADHIGH command to the loading statement, in front of LOADER or LOADERE:

```
LOADHIGH LOADER (or LOADERE) [servername]
```

where *servername* is the name of the server being loaded.

The following commands can be used when you load the LANDP for DOS functional areas.

ASCII-EBCDIC translation server

```
LOADER EHCDBTR.EXE
```

Compression server

```
LOADER EHCCOMP.EXE
```

Electronic journal server

```
LOADER ELECJO##.EXE [/K:y]
```

where:

y Is the size, in KBs, of the buffer used to insert the electronic journal records. It ranges from 1 to 4. The default is 1.

The size of the buffer must be large enough to hold the maximum of:

- Size of the longest electronic journal record plus 32B
- Split criteria plus 36B plus size of the selected keys. Note that you have to include the *hidden key*, which is 8 bytes long.

Forwarding server

```
LOADER FORWARD.EXE /O:vvvvvvvv.vvv [/T:wwww] [/S:xxxxxxxx.xxx] [/K:y] [/H:z]
```

where:

vvvvvvvv.vvv
Is the name of the file corresponding to the ASCII-to-EBCDIC translation table. It must follow the operating system rules.

wwww Is the number of time ticks after which the supervisor will dispatch the forwarding function. One time tick is roughly 0.05 seconds.

Values can range from 1 to 6000. The default is 20 (about 1 second).

xxxxxxxx.xxx
Is the name of the file corresponding to the sign-on feature. It must follow the operating system rules.

y Is the size, in KBs, of the buffer used to read the store-for-forwarding records. It can range from 1 to 4. The default is 1.

The parameter value must be the value assigned in the loading statement of the store-for-forwarding server.

z Specifies whether headers are included when sending host computer messages. The parameter value can be `Y`, to include headers, or `N`, not to include headers. The default is Y.

LAN server

```
LOADER LAN.EXE /pc-id [/N:n] [/B:bb] [/I:x] [/S:y]
```

where:

pc-id Is the one (or two) character alphanumeric identification of the workstation that was assigned during customization. The *pc-id* must be the same as the one for the supervisor.

n Is the adapter number used by the workstation.

The parameter value can be 0 or 1. The default is 0, or the value specified at workgroup level.

bb Is the number of KB that should be reserved for the internal buffer to be shared with the NetBIOS manager layer in order to receive data from remote workstations.

The parameter value can range from 1 to 56. The default is 4, or the value specified at workgroup level.

If an incoming message size is bigger than the buffer size, performance degrades.

x Is the time interval, in seconds, between attempts to establish the required NetBIOS sessions.

The parameter value can be 0 or in the range 5 through 3000. The default is 20 seconds. If a value of 0 is specified, only one attempt is made to establish each session, at startup, or after a session has been lost.

y Is the NetBIOS send timeout, in seconds.

The parameter value can be 0 or in the range 10 through 127. The default is 10 seconds. A value of 0 implies no timeout.

The LAN server is needed only when more than one workstation is present in a workgroup that uses NetBIOS as transport protocol. It is included during the process of creating diskettes for distribution.

LANDP Internet Protocol

```
LOADER EHCLIP.EXE /pc-id [/N:n] [/Y] [/R:r]  [/C[:ws-id]] [/P:p] [/J] [/D:d]
```

where:

pc-id Is the workstation ID that was specified during customization. The *pc-id* must be the same as that for the supervisor. This parameter must always be the first one.

n Specifies the TCP/IP port number used by LANDP Internet Protocol.

The parameter value can range from 1024 to 65535. The default is 52699.

Y Specifies that no availability probe datagram will be sent, when a session has no normal traffic.

r Specifies the storage, in bytes, to be allocated for the retransmission table.

This table is used by LANDP Internet Protocol to save information about datagrams sent, in case they should be retransmitted.

The parameter value ranges from 256 to 65000. The default value is calculated using the following formula:

```
910 * ((N * 40) / (N + 34))
```

where *N* is the number of related workstations.

`C` Requests LANDP Internet Protocol communications trace. All sessions are traced, except when the *ws-id* parameter is specified. In this case, only the session with the workstation specified in that parameter is traced.

p Specifies the number of trace pages to be displayed for LANDP Internet Protocol communications trace. Each trace page you add requires 836 bytes.

The parameter value can range from 3 to 70. The default value is 3.

`J` Specifies that no checking for related workstations with undefined LANDP Internet Protocol addresses will be carried out.

`d` Specifies the maximum datagram length in bytes. The parameter value can be in the range 688 through 32711. The default is 1472.

Local resource manager server

```
LOADER EHCLRMGR.EXE
```

Magnetic stripe reader/encoder server

```
LOADER MSRE47##.EXE
```

Native X.25 server

```
LOADER X25NAT##.EXE
```

Operator interface

```
LOADER OPER.EXE
```

PIN pad server

```
LOADER PINP47##.EXE /M
```

where `/M` indicates that 4778 magnetic stripe reader capabilities are to be used.

Printer manager server

```
LOADER PRTMGR.EXE
```

The loader statement for the printer manager server will be included automatically during the process of creating diskettes for distribution if the 3287 printer emulator or the financial printer server is using the parallel port.

Remote change management services

`LOADER RCMS.EXE /I:xxx /O:yyy [/T:ttt] [/A:mmm] [/L:n] [/P:q] [/C:x] [/R:r]`

where:

xxx Is the file extension of the EBCDIC-to-ASCII translation table (EARCMS.*xxx*). In DBCS mode the parameter does not apply.

yyy Is the file extension of the ASCII-to-EBCDIC translation table (AERCMS.*yyy*). In DBCS mode the parameter does not apply.

ttt Is the number of timer ticks after which RCMS receives control. One timer tick is roughly 0.05 seconds. The default is 10. The value must be a decimal number in the range 1 to 999. For example, `/T:1`, `/T:12`, or `/T:255`.

mmm Is the amount of memory reserved for COMMAND.COM (in KB). You can enter values in the range 48 to 256. If you do not enter a value for *mmm*, 47.5 KB is reserved. The default amount of memory reserved for COMMAND.COM is not sufficient for some versions of DOS. For example, PC-DOS 7.1 requires 53 KB.

n Specifies the number of lines in the EHCRCMS.LOG file. The parameter value ranges from 100 to 10000. The default is 1000.

***Attention:* If an EHCRCMS.LOG file with *n1* lines exists already, and you choose a value for the L parameter that is different from *n1*, your old file will be destroyed and a new one created. If you want to keep the old file, rename it or copy it to somewhere else before running the LOADER program.**

q Specifies the translation mode. The parameter applies only to DBCS mode. The parameter value can be:

S Standard ASCII-EBCDIC and EBCDIC-ASCII translation.

P 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.

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

x Specifies the reception mode for CLISTs. The parameter value can be:

B The CLIST is received as a binary file.
E The CLIST is received as an EBCDIC file.

The default is E.

r Is the interval, in minutes, before retrying a connect to SNA if there is a communications problem. The value must be an integer in the range 0 (which is taken to mean 30 seconds) and 8 (8 minutes). The default value is 0.

Searcher

`LOADER SFQUERY.EXE [/K:y]`

where:

y Is the size, in KBs, of the buffer used to read the electronic journal and store-for-forwarding records. It ranges from 1 to 4. The default is 1.

The parameter value must be the highest of the values assigned in the loading statements of the electronic journal and store-for-forwarding servers.

This program is required by the electronic journal and the store-for-forwarding servers.

Shared DOS directory server

```
LOADER SHRDIR.EXE /E:nn
```

where:

nn Is the maximum number of entries in the profile table (SHRDIR.PRO) that describe the relationship between the short name and the directories to be shared.

In the client workstations that have IBM DOS 7.1 or higher installed, the customization program includes the following statement in the AUTOFBSS.BAT file:

```
SHRDIRDD /K:mm
```

where:

mm Is the size of the request/reply data area used by the shared DOS directory server. The parameter value can range from 1 to 56. The default is 4.

Shared-file server

```
LOADER SHFILE##.EXE [/C:confname] [/B:nnn] [/E] [/S:xxx] [/L:y] [/F:zz]
```

where:

confname
Specifies the name of the shared file server profile. If you omit this parameter, the system uses the name CONFIGUR.

nnn Specifies the number of additional 1 KB index buffers to be allocated; that is, buffers over 15. More index buffers increase system throughput, but also reduce the amount of free storage available for the server workstation. A rule of thumb is that the number of buffers should be 10 per workstation using the shared file server simultaneously. A practical limit is approximately 100, depending on available storage size. The maximum value is 968.

Another factor that must be considered is that the more buffers you have, the greater is the probability of losing index data when the shared file server is abnormally ended. Thus, if many index buffers are allocated, and the server workstation is switched off with a transaction still in process, or if no RF function has been called, an automatic index rebuild is issued the next time the server is loaded.

/E Is an optional parameter to create a file for the statistics gathered during the session.

xxx Specifies the total number of additional sessions in the whole workgroup that the server can manage. The maximum is 245.

This number plus the number of workstations that receive services must not be higher than 245.

y Specifies the log management type. The parameter value can be:

`0`	Dynamic and static log with a unique log file
`1`	Dynamic log with a unique log file
`2`	Dynamic and static log with two log files
`3`	Dynamic log with two log files

The default is 0.

zz Is the number of files open at a time.

The parameter value can range from 10 to 245.

Note: 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. To overcome this problem, issue the GENLOG command. (For further information see *LANDP Servers and System Management*.)

SNA server

```
LOADER SNA##.EXE /P:nn
```

where:

nn Is the number of trace pages to be displayed for the SNA server.

The parameter value can range from 3 to 70. The default is 3.

Store-for-forwarding server

```
LOADER SFORFORW.EXE [/K:y]
```

where:

y Is the size, in KBs, of the buffer used to insert the store-for-forwarding records. It ranges from 1 to 4. The default is 1.

The size of the buffer must be large enough to hold the maximum of:

- Size of the longest store-for-forwarding record plus 32 KB
- Split criteria plus 36 KB plus size of the selected keys. Note that you have to include the *hidden key*, which is 8-bytes long.

Supervisor

```
LOADER SPV.EXE /pc-id [/D:yy] [/L:zz] /K:mmm
```

where:

pc-id Is the one (or two) character alphanumeric identification of the workstation that was assigned during customization. This parameter is required.

yy Is the percentage of dispatching cycles where control is maintained by the application and pending LANDP tasks are not processed. That is, *yy*% of dispatching cycles come back to the application immediately, and (100–*yy*)% of the cycles perform LANDP processes before returning to the application. Therefore, *yy* can be any number from 0 to 99. The default is 0.

zz Is the numeric value for the number of screen lines used to display the trace. Valid numbers are from 1 to 24. The default is 12.

mmm Is the number of KB that should be reserved for the internal buffer pool to allocate incoming/outgoing requests from/to remote workstations, and requests from servers loaded into expanded memory to servers that are also loaded into expanded memory.

The parameter value ranges from 7 to 512. It can be specified during customization through the POOLSIZE keyword in the LWSCONF vector.

The default calculated by the customization program is [1.25 x *N*], where *N* is the number of workstations that receive services from the workstation you are defining, or provide services to it.

The value is limited by the available memory. In addition, a low value may result in a poor performance. Thus, it is highly recommended to adjust this value accurately. On the panels of the trace tools you can obtain information about the percentage of buffer pool being used.

If the size of the buffer pool is not big enough, the requester may receive status X'01004C46' (no free space in the buffer pool), X'01004C47' (no free space in the remote buffer pool), or X'01004C48' (timeout, due to the fact that a reply is not possible because no free space is available). On the Routines Trace panel of the trace tools you can see the number of times that a requester received these statuses.

For workstations that provide or receive shared DOS directory services, the customization program recalculates the default accordingly.

If you are using the shared DOS directory, a situation may occur where there are insufficient Supervisor buffers. This may result in the following symptoms:

- The message: `Abort, Retry, Fail` may be displayed at the command line when using the remote accessed drive.
- The client workstation routines trace will show that “out of local buffer” conditions have occurred. LF and LA errors *may* be present in the client workstation traces.

Increasing the value of *mmm* will resolve the error.

Synchronous data link control server

```
LOADER SDLC.COM /C:x /T:nn /I:yy
```

where:

x Is a parameter used for SNA/SDLC switched communications. The parameter value can be:

- Y Data terminal ready (DTR) will be activated and deactivated by applications, using the Connect (CN) and Release (RL) functions of the SNA server.
- N DTR of the modem will be activated automatically when LANDP for DOS is ready.

The default is N.

nn Is the number of seconds of the *line inactivity timer*.

The parameter value can range from 10 to 65. The default is 65.

yy Is the interrupt request numbers to be used for the IBM Asynchronous/SDLC Communications Adapter.

The parameter value can be:

34	Use interrupt requests 3 and 4
54	Use interrupt requests 5 and 4
37	Use interrupt requests 3 and 7
57	Use interrupt requests 5 and 7

The default is 34.

The parameter values must match the jumper settings on the adapter card.

System manager server

```
LOADER SMGR.EXE /D:x /O:yyyyyyyy
```

where:

x Is the drive where the FBSS#GDT backup is located. If the parameter is omitted, the backup is not performed.

yyyyyyyy
Is the NetView operator ID. The default is OPER1.

System manager operator

```
LOADER SMOP.EXE
```

TCP/IP wide area communications server

```
LOADER xxx.EXE [/S:nn] [/T]
```

Where:

xxx Is either SNA## to emulate the SNA server or PPC to emulate the PPC server. EHCTCP.EXE has to be renamed to SNA##.EXE or PPC.EXE. It is not possible to emulate both SNA##.EXE and PPC.EXE on the same DOS workstation.

nn Is the maximum number of TCP/IP wide area communications server sessions that can be in use at any time. The default and minimum values are 5. The maximum value is 2048, but available memory forces a lower maximum.

/T Is an optional parameter to turn on internal tracing for problem determination.

Token-ring data link control server

```
LOADER TRDLC.EXE
```

Trace tools

```
LOADER DDT.EXE /I:x /R:y /P:nn
```

where:

x Is a parameter with two possible values:

- I Specifies that an internal function trace is included. The information refers to server-to-server requests.
- E Specifies that an internal function trace is excluded.

The default is I. For application debugging purposes, choose the E parameter value.

y Is a parameter with two possible values:

- I Remote requests trace is included. The information refers to incoming requests from remote workstations.
- E Remote requests trace is excluded.

The default is I. For application debugging purposes, choose the E parameter value.

nn Is the number of trace pages to be displayed for the supervisor functions. Each trace page you add requires 1216 bytes.

The parameter value can range from 3 to 40. The default is 3.

Note: If you use the LOADERE loading command, the parameter value ranges from 3 to 30.

X.25 DLC server for IBM PC X.25 communications adapter

```
LOADER X25DLC.EXE
```

X.25 DLC server for IBM X.25 interface co-processor/2 adapter

```
LOADER X25DLC2.EXE [/T]
```

If the X25DLC2 server is used, it is renamed to X25DLC.EXE and loaded accordingly.

If option /T is added to the loading statement, the X.25 trace will be started from the beginning.

Note: The two X.25 servers (one used with the IBM PC X.25 Communications Adapter and the other used with the IBM X.25 Interface Co-Processor Adapter/2) use the same LOADER statement. However, the /T parameter is valid only for the IBM X.25 Interface Co-Processor Adapter/2.

3270 emulator

```
LOADER EMU3270.EXE /C:atr /K:kbd /D:dis /I:n [/H:hh] [/W:www] [/B:y] [/S:xxxxxxxx]
[/Z:nnnn] [/P:a] [/L:l] [/T:tt]
```

where:

atr Is the name of the selected display color attributes table.

kbd Is the name of the selected keyboard ASCII-to-EBCDIC translation table.

dis Is the name of the selected display EBCDIC-to-ASCII translation table.

n Is the emulator identification number.

hh Specifies the alternate screen height (number of rows) of the 3270 display to be emulated. (The height specified should not include the operator information area line at the bottom of the emulator screen.) *nn* must be in the range 24 through 49. For 132-column screens, the maximum height might be limited by the capabilities of the PC video display adapter installed in your system.

Use this parameter, in conjunction with /W, to make the emulated alternate screen look like one of the following 3270 models:

3270 model	Alternate screen height	Alternate screen width
2	24	80
3	32	80
4	43	80
5	27	132

If this parameter is omitted, the default is 24.

www Specifies the alternate screen width (number of columns) of the 3270 display to be emulated. *nnn* must be either 80 or 132. Some PC video display adapters do not support 132-column mode.

If this parameter is omitted, the default is 80.

y Indicates whether blinking is supported. Specify Y for yes or N for no.

If this parameter is omitted, the default is N.

xxxxxxxx Specifies the long name of the 3270 emulator session (sometimes known as the “host session ID”), which is displayed in the operator information area on the screen. You can specify up to eight characters (with no imbedded blanks).

If this parameter is omitted, the default is a name of eight blanks.

nnnn Specifies the size (in bytes) of the buffer used to communicate with the host. *nnnn* can be any value in the range 2048 through 4096. Specifying a small buffer size minimizes memory requirements; using a large buffer can reduce the number of transmissions needed to send or receive a large data stream.

The parameter value specified must match the RU size detailed in the bind session.

If this parameter is omitted, the default is 2048.

a Indicates whether the 3270 emulator should handle the Print Screen key. Specify N for no or Y for yes.

If this parameter is omitted, the default is Y.

l Indicates whether the SNA session is connected at emulator load time rather than at ‘hot key’ time. Specify N for no or Y for yes. The default is N.

tt Specifies the minimum time in seconds between checks on ‘print screen’ key presses. The default is 3.

3287 printer emulator

```
LOADER EMU3287.EXE /x /E:prt [/T:nn] /P:HP [/N:n]
```

where:

x Can be `S` or `M`. Use `S` for single and `M` for multiple LU_1 support.

Note: This parameter and its values are no longer supported, though they will be accepted for compatibility purposes with earlier versions of LANDP. If specified, they will be ignored.

prt Is the name of the selected EBCDIC-to-ASCII translation table.

nn Is the frequency of polling.

The parameter value ranges from 1 to 60. The default is 15.

`/P:HP` Indicates that the 3287 printer emulator will use either an IBM 4019 Printer or an IBM 4029 Printer for output.

`/N:n` Specifies the range of logical printer numbers that can be used. The parameter value can be in the range 1 through 3. Specify:

- `/N:1` to use LPT1 only
- `/N:2` to use LPT1 and LPT2
- `/N:3` to use LPT1, LPT2, and LPT3

If the parameter is not specified, the emulator uses the number of parallel printer ports physically installed on the workstation.

Financial printer server

```
LOADER PR47X2##.EXE [/K:n]
```

where:

n Is the maximum number of KB to be printed at a time.

The parameter is optional. The parameter value can range from 1 to 4. The default is 1.

4748 printer server

```
LOADER PR4748##.EXE [/K:n] [/H] [/N] [/A:xxxx]
```

where:

n Is the maximum number of KB to be printed at a time.

The parameter is optional. The parameter value can range from 1 to 4. The default is 1.

H Shows that ALERT support is required.

N Shows that code conversion is not required.

xxxx Is the fully qualified filename of the User-Defined Character (UDC) file.

Unloading LANDP for DOS

The FREE.EXE utility is provided to unload LANDP for DOS. You can also unload LANDP for DOS by issuing a supervisor function call from an application program. For more information on supervisor function calls, see *LANDP Programming Reference*. The FREE.EXE, is called as follows:

```
d:\path\FREE
```

where:

d: Is the drive where the utility is located.

path Is the path where the utility is located.

Notes:

1. If you use the X.25 Co-processor, you must unload the support program and the interrupt handler using the X.25 Co-processor support program functions.
2. If you use TCP/IP internal communications, you must unload LANDP for DOS before stopping TCP/IP.

Unloading LANDP for DOS in Microsoft Windows 3.1/3.11

To unload LANDP for DOS, the FREE program must be run from the application VDM after EHCWVDMI. The FREE program unloads only the LANDP modules loaded after Microsoft Windows 3.1/3.11.

To unload EHCWGMDI and the servers that were loaded before Windows 3.1/3.11, you must exit Microsoft Windows and then enter:

```
FREE /W:1
```

Using expanded memory

The maximum amount of conventional memory (640 KB) is sometimes not enough for large programs or groups of programs. Lotus Development Corporation, Intel Corporation, and Microsoft Corporation created the Lotus/Intel/Microsoft (LIM) Expanded Memory Specification to enable programs to be loaded into expanded memory.

The LIM Expanded Memory Specification defines the software interface between the Expanded Memory Manager (EMM), a device driver that controls and manages expanded memory, and programs that use expanded memory.

To use the LIM Expanded Memory Specification for loading servers into expanded memory, a special loader program called LOADERE is included in the LANDP for DOS distribution package.

To load a server into expanded memory, you have to select expanded memory for the server during customization. The customization program places the correct statements in the CONFIG.SYS and AUTOFBSS.BAT files, depending on the version of DOS chosen, and the LOADERE program loads the servers into expanded memory. The LOADERE program itself is always loaded in conventional memory.

The following LANDP for DOS functional areas can be loaded in expanded memory. The user servers that conform to the rules listed in “Rules for user servers in expanded memory” on page 124 can also be loaded in expanded memory. Note that the corresponding definitions are provided during customization through the DEFSERV vectors.

ASCII-EBCDIC translation server	EHCDBTR.EXE
3270 emulator	EMU3270.EXE
3287 printer emulator	EMU3287.EXE
PIN pad server	PINP47##.EXE
Electronic Journal	ELECJO##.EXE
Store-for-Forwarding	SFORFORW.EXE
Searcher	SFQUERY.EXE
Forwarding	FORWARD.EXE
MSR/E server	MSRE47##.EXE
Financial printer server	PR47X2##.EXE
4748 printer server	PR4748##.EXE
Trace	DDT.EXE
SNA server	SNA##.EXE
Shared file server	SHFILE##.EXE
X.25 Co-Processor	X25DLC.EXE
Operator interface	OPER.EXE
System manager operator	SMOP.EXE
System manager	SMGR.EXE
Local resource manager server	EHCLRMGR.EXE
Remote Change Management Services	RCMS.EXE
LANDP Internet Protocol	EHCLIP.EXE

Notes:

1. If IBM PC 3270 Emulation LAN Management Program is installed, the system manager server cannot be loaded into expanded memory.
2. If you are using PC code page 950 (Taiwan) or host code page 1388 (People's Republic of China), the ASCII-EBCDIC translation server (EHCDBTR.EXE) cannot be loaded into expanded memory.

The following PC/Integrator software can also be loaded into expanded memory:

Banking interactive workstation program	BIWP.EXE
Banking printer program	BPP.EXE

Software requirements

Depending on the DOS version selected during customization, your CONFIG.SYS file is updated differently.

If you select DOS, the following statements are added to your CONFIG.SYS file:

```
DEVICE=HIMEM.SYS
DEVICE=EMM386.EXE [parameters]
```

The second statement is valid only for personal computer systems with a processor higher than 80286. If your systems have processors lower than 80386, you must change your CONFIG.SYS file and use the XMA2EMS.SYS device driver or another specific device driver related to your memory adapter.

Note that the FRAME value must range from C000 to E000, in increments of 400h.

For more information, refer to the DOS library.

LOADERE checks for the largest frame that is available consisting of contiguous pages.

Important: Memory problems may occur due to incorrect memory allocations. Make sure the values specified in your CONFIG.SYS file are not overlapping with addresses used by the adapters installed. You can verify or change the memory values either in the CONFIG.SYS file or in the workstation configuration diskette.

Following is an example of a configuration for a PS/2 Model 70 (80386) with Token-Ring (16Mb (bits) per second), X.25 Co-Processor, and DOS installed. The highlighted lines show a possible memory address distribution:

```
Slot 1 - Token Ring Network 16/4 Adapter/A
         Primary or Alternate Adapter........[Primary  ]
         Adapter Data Rate...................[16 Mps]
    ROM Address Range...................[DE000 - DFFFF]
   RAM Size and Address Range..........[16 KB / DA000 - DDFFF]
         Interrupt Level.....................[Interrupt 2]


Slot 2 - IBM Realtime Interface Co-Processor Multiport/2
         Physical Card number...........[Physical Card 0; 02A0H - 02A7H]
    Shared Storage Window Location
       and Size.......................[D8000 - D9FFF (8 KB Window)& rbrk.
         Interrupt Level.....................[Interrupt Level]
         Port 0 Transmit Clock Source...[DCE Sourced Clocking]
         Port 0 Receive Clock Source....[DCE Sourced Remote Clocking]
         Port 1 Transmit Clock Source...[DCE Sourced Clocking]
         Port 1 Receive Clock Source....[DCE Sourced Remote Clocking]
```

Possible CONFIG.SYS entries could be:

```
         BUFFERS=20
         FILES=40
         DOS=HIGH,UMB
         DEVICE=HIMEM.SYS
    DEVICE=EMM386.EXE RAM FRAME=C000 /X=D800-DFFF
         DEVICEHIGH=DXMAOMOD.SYS
         DEVICEHIGH=DXMCOMOD.SYS
         DEVICEHIGH=DXMTOMOD.SYS  ST=26  S=26  c=30
```

Rules for user servers in expanded memory

LIM EMS involves some restrictions related to loading servers in expanded memory. Besides, user servers to be loaded in expanded memory must conform to the following rules:

1. LOADERE uses the largest contiguous frame found for the EMM LIM. The servers that can be loaded must fit within this frame. Depending on the fragmentation of the frames, some servers that could be loaded on some configurations of the EMM LIM will fail in loading on other configurations.

2. The maximum contiguous frame possible is 64 KB. Any servers above that size, including any additional memory allocated at load time by using the SETBLOCK function under DOS INT 21H, are not loaded by LOADERE.
3. If the server must expand itself (allocate more memory), it must do it at load time using the DOS Allocate Memory Block or the DOS Set Block function calls.
4. The memory allocated through the DOS INT 21H Allocate Memory Block function is not in expanded memory. This memory is provided by DOS from its conventional memory (640 KB).
5. LOADERE keeps track of the interrupts that the server is chaining into. To do this, servers must chain into an interrupt vector at IN time, using the DOS INT 21H SET VECTOR function. A maximum of eight interrupt vectors can be set.
6. Servers that catch hardware interrupts, or software interrupts inside hardware interrupts, cannot be loaded with LOADERE.

Using high memory

To use high memory (between 640K and 1MB), Upper Memory Block (UMB) support is required. Set the CONFIG.SYS file to enable UMB support, following the standard DOS rules.

Because IBM TCP/IP uses UMB support, it is recommended not to load LANDP for DOS functional areas into high memory in IBM TCP/IP environments.

Installing run-time files

When you install the run-time files on a DOS workstation, take the following into account:

- If the workstation has a hard disk, install the run-time files on that disk.

  Note: The shared-file server requires a workstation with a hard disk.

- Load all the programs that change interrupt vectors, for example KEYB or PRINT, before loading the servers and the supervisor.

  If you load such programs after the servers or the supervisor, unpredictable results may occur.

- When using LANDP for DOS, the PC Local Area Network (LAN) Program, and the DOS LAN Requester, issue the NET START command before loading LANDP for DOS.

  Assign to the parameters SES and CMD values larger than those needed for the PC LAN Program.

- When NetBIOS is used as the transport protocol, if LANDP for DOS, the PC LAN Program, and the DOS LAN Requester are used together, they share the resources of NetBIOS.

- When TCP/IP is used as the transport protocol, TCP/IP must be started before loading LANDP for DOS, and LANDP for DOS must be unloaded before stopping TCP/IP.

The LANDP family provides a utility program to check the path where the run-time files are located. See “Installing and validating system files” on page 236 for more information about this program.

For other utility programs also provided to be used at run-time, refer to Chapter 15, “Run-time utility programs” on page 229.

Chapter 12. Preparing OS/2 workstations

The first part of the chapter, “Installing and configuring IBM OS/2 workstations,” describes the requirements to install the LANDP for OS/2 run-time files on the OS/2 workstations. Some requirements are related to a specific server.

The second part of the chapter, “Modifying run-time files” on page 139, describes the process of modifying the run-time files created by the customization program, according to your needs.

The third part of the chapter, “Installing run-time files” on page 165, describes how to check for a proper installation of the run-time files.

Appendix G, “LANDP implementation using WorkSpace On-Demand” on page 541 gives additional consideration for implementing LANDP for OS/2 on Workspace on Demand.

Installing and configuring IBM OS/2 workstations

The diskettes created for the OS/2 workstations do not contain any OS/2 system files. Therefore, before installing the LANDP for OS/2 operational diskettes into your OS/2 workstation, make sure that OS/2 is installed according to the existing recommendations.

The LAN adapter and protocol support (LAPS) can be provided by one of the following products, or later versions:

- IBM SecureWay Communications Server for OS/2 Warp, Version 6.0
- IBM SecureWay Personal Communications Version 4.21 for DOS/Windows, Windows 95, Windows NT, and OS/2
- IBM OS/2 LAN Server V. 2.0 or V. 3.0
- IBM LAN Enabler V. 2.0
- IBM NTS/2 V. 1.0
- IBM TCP/IP V. 1.21

When working in DBCS mode, the operating system can be one of the following, or later versions, depending on your national language:

Language	**Operating system**	**Supported codepage**
Traditional Chinese	IBM OS/2 T4.0	938, 950 (no defaults)
Korean	IBM OS/2 H4.0	949
Simplified Chinese	IBM OS/2 P4.0	1381, 1386 (no defaults)

Installation requirements for NetBIOS transport protocol

This section applies to LANDP for OS/2 workstations integrated in a LANDP workgroup that uses NetBIOS as the transport protocol.

Table 5. Maximum Requirements for LANDP for OS/2 Workstations in a LANDP Workgroup

Sessions	*nn*
Commands	MIN(*nn*+2, 70)
Names	1

Where *nn* is the number of workstations in the LANDP workgroup minus 1. This value corresponds to the case when every workstation in the workgroup needs a NetBIOS session to all the workstations in the LANDP workgroup.

To optimize the resources, for each workstation you can specify a lower value, which depends on the workstation requirements. This value refers to the number of workstations the workstation is related to - those to which it provides services, or from which it receives services.

The resources have to be specified in the current PROTOCOL.INI file that is used by LAPS. You should take into consideration the requirements of other programs installed in the same workstation. If OS/2 LAN Server or LAN Requester is installed, take into account the resources assigned through the IBMLAN.INI file.

Check the following statements in the PROTOCOL.INI file:

```
[NETBEUI]
        .
        .
        .
      SESSIONS = nn      + (number of sessions used by other NetBIOS programs)

      NCBS = nn+2   + (number of commands used by other NetBIOS programs)

      NAMES = 1          + (number of names used by other NetBIOS programs)
         .
         .
         .
```

In addition, check for the DEVICE statements that correspond to the following LAPS modules in the CONFIG.SYS file:

- NetBIOS protocol driver (NETBEUI.OS2)
- NetBIOS device driver for application programming interface support (NETBIOS.OS2)

Finally, add the DEVICE statement corresponding to the adapter used in the workstation, and any other DEVICE statement to meet your own requirements.

For information on those statements, refer to the manuals of the corresponding products. See “Bibliography” on page 571.

Installation requirements for TCP/IP transport protocol

This section applies to LANDP for OS/2 workstations integrated in a LANDP workgroup that uses TCP/IP as the transport protocol.

To use TCP/IP as the internal communications protocol for a LANDP workgroup, OS/2 TCP/IP must be installed and configured in all the workstations.

For detailed information, refer to the TCP/IP manuals. See “Bibliography” on page 571.

For more information about the TCP/IP transport protocol, see see Appendix E, “Using TCP/IP for internal communication” on page 525.

Using TCP/IP X.25 support

If you plan to use TCP/IP X.25 support for LANDP workstations internal communications, consider the limit of X.25 switched circuits supported by OS/2 TCP/IP, and the characteristics of this support.

It is recommended that for each workstation-to-workstation session a circuit is always available when there is data exchange. Otherwise, the response time and session lost conditions may increase dramatically.

Installation requirements for workstations with SNA servers

The LANDP for OS/2 SNA server uses the LUA interface that is provided by one of the following SNA communications providers :

- IBM SecureWay Communications Server for OS/2 Warp, Version 6.0
- IBM SecureWay Personal Communications Version 4.21 for DOS/Windows, Windows 95, Windows NT, and OS/2

The workstation containing the SNA server must be defined in the appropriate SNA communications provider as the workstation connected to the host. All logical units types 0, 1, and 2 receiving services from the LANDP SNA server must be defined in the appropriate SNA communications provider of the workstation in which the SNA server will be loaded. That is, the workstation that is connected to the host computer.

In the configuration files of the appropriate SNA communications provider, you define the workstation profile and the SNA profiles. Within the SNA profiles you update the following sections:

1. DLC profiles
2. SNA local node characteristics
3. SNA connections
4. SNA LUA APIs

If you do not use LU pooling support, you have to define as many LUA profiles as SNA sessions.

If you plan to use LU pooling support, you have to group LUA profiles in LU pools. In this case, the number of LUA profiles depends on your configuration requirements.

The profile name is a string of eight characters, which must comply with the following conventions:

```
EHCxxnnb
```

where:

xx Has two different meanings:

- If you do not use LU pooling support, *xx* stands for the workstation ID of the workstation that uses the session. The parameter value is a string of two alphanumeric characters.

  If the workstation ID has only one character followed by a blank, the profile name you have to define is `EHCx@`*nn*`ƀ`.

- If you use LU pooling support, *xx* stands for the pool ID assigned to the session during customization.

nn Is the SNA session identifier.

- If you do not use LU pooling support, see below for the correct values.
- If you use LU pooling support, you should number the identifiers from 01 to 99, sequentially. If you skip one number, only the identifiers specified before will be considered. For example, if you specify 01, 02, 03, 05, ... , 50, only the SNA sessions 01, 02, and 03 will be considered.

b Is one character you leave blank.

The variable *nn* takes the following values.

For the 3270 emulator:

You can have five sessions running simultaneously in each workstation, with the identifiers numbered from 51 (for the first session) through 55 (for the fifth session). That is, the value for *nn* for the 3270 emulator ranges from 51 to 55.

For the 3287 printer emulator:

You can have five sessions running simultaneously in each workstation, with the identifiers numbered from 76 (for the first session) through 80 (for the fifth session). That is, the value for *nn* for the 3287 printer emulator ranges from 76 to 80.

For RCMS (for LANDP for DOS, OS/2, and Windows NT):

The value for the variable *nn* for RCMS is 33.

For Forwarding (for LANDP for DOS, OS/2, and Windows NT):

You can have three sessions running simultaneously, with the identifiers numbered from 37 (for the first session) through 39 (for the third session).

For applications:

For DOS applications, the variable *nn* can take the values of 1 through 15. For DOS user servers, it can take the values of 16 through 30.

For OS/2 or Windows NT applications or user servers, the variable *nn* can take the values of 1 through 30. However, 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. The SNA session identifier part of the LUA profile name (defined under the appropriate SNA communications provider) may now be any two ASCII characters instead of just two decimal digits.

For 4731 and 4737 (for LANDP for DOS):

The variable *nn* can take the values 34 and 35 for the 4731 Personal Banking Machine (PBM), and the value 34 only for the 4737 PBM.

For BIWP (Running in OS/2 MVDM and PC/Integrator):

The variable *nn* takes the value 63. For the value to specify when BIWP runs in OS/2, refer to the PC Integrator/2 manuals.

For the LDA 7 Program (Running in OS/2 MVDM and PC/Integrator):

The variable *nn* takes the value 64. For the value to specify when the LDA 7 program runs in OS/2, refer to the PC Integrator/2 manuals.

Following is an example of communications provider definition of the SNA LUA profile names for a given workgroup, for which LU pooling is not used:

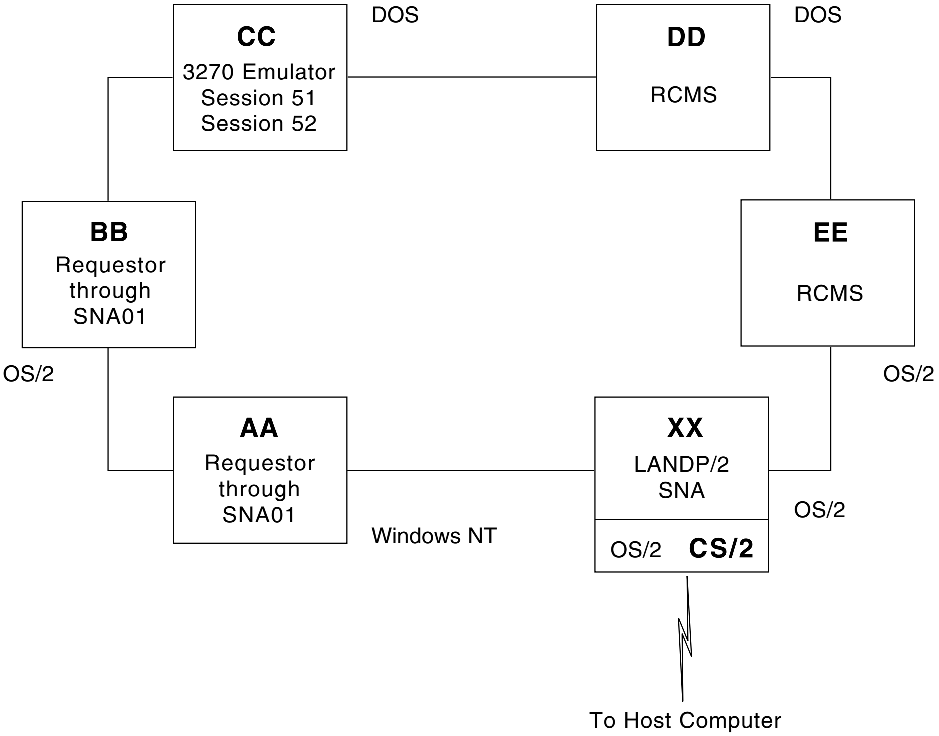


Figure 5. Example of SNA LUA profile names in a workgroup (OS/2)

These are the expected profile names in the communications provider for workstation XX (gateway):

EHCAA01	For the client workstation AA
EHCBB01	For the client workstation BB
EHCCC51	For the first emulator session in workstation CC
EHCCC52	For the second emulator session in workstation CC
EHCDD33	For the RCMS client workstation DD
EHCEE33	For the RCMS client workstation EE

Then you specify the following profile parameters:

Comment
This entry is optional

Local NAU address
The address of the valid host connections

DLC Type
The DLC profile customized earlier for the gateway

Note: If you use *correlation tables*, ensure the Communications Server LUA level supports the corresponding definitions.

Installation requirements to use the DC and QC functions

The LANDP for OS/2 SNA server provides the define connection (DC) and query connection (QC) functions.

These functions require that the communications provider is installed in the workstation where the SNA server is loaded.

To use the DC and QC functions, the communications provider requires further configuration:

1. Define as many X.25 directory entries as X.25 destinations you are to use.

   The following sample shows the most relevant fields in a X.25 directory entry definition:

   ```
   Directory Entry Name    ENTRY01
                           Directory Entry        Remote
                           Directory Type         SNA
                           Virtual Circuit Type   SVC
                           Link Name              PHYSLINK
                           Link Station Role      Secondary
                           Remote DTE Address     111111111
   ```

2. Define a SNA connection for one of the X.25 directory entries you have defined. The X.25 directory entry you choose becomes the default for all the LUs over that SNA connection.

The following sample shows the most relevant fields in a SNA connection definition:

```
Link Name    LINK0002
             X25 Directory Entry    ENTRY04
             Node ID                05D 80002
```

3. Define all dependent LUs (LUA and 3270 emulations). All the LUs in the same PU must be associated to the same SNA connection.

   The following sample shows the most relevant fields in a LU (LUA) definition:

```
LU Name   EHCAA01
          Host Link Name   LINK0002 (to X25 SNA Connection)
          NAU Address      001
```

All the subscriber numbers that become parameter values for the DC function must be defined as X.25 directory entries. These X.25 directory entries must not be referenced by any other SNA connection.

Installation requirements for cryptography management

The LANDP for OS/2 SNA server can manage cryptography.

If cryptography is managed by the SNA server, copy the following files to the D:\CMLIB\DLL path, where D is the drive where the communications provider is installed.

ACSRENCR.DLL
ACSRDECR.DLL

Installation requirements for workstations with native X.25 servers

To run the native X.25 server, you have to provide some definitions for the communications provider configuration.

You should provide the following definitions. The given values are mandatory.

For the X.25 feature profiles:

- Link profile name: FBSSLINK

  You must create this profile.

- Routing table entry name: FBSSROUT

  You must create this table for incoming calls expected to be received.

To ensure that all the communications provider files needed for X.25 management are located in the \CMLIB directory, reinstall the communications provider using a properly defined configuration file.

Installation requirements for workstations with PPC servers

To run the PPC server, you have to provide some definitions for the communications provider configuration.

The definitions that can be involved are shown in this section. However, you do not need to specify all the definitions in all the cases, due to the communications provider dynamic configuration.

Most definitions and the resulting configuration should be regarded as a pattern to define your own configuration.

For the workstation profile and automatic start options.

Translation table

The translation table specified in the workstation information option can be ACSGTAB.DAT.

Local node characteristics

- Network ID = *Name of the network*

  The parameter value can be, for example, FBSSNET.

- Local node name = *Name of the machine*

  The parameter value can be, for example, LAN1WS01.

- Node type =

  The default is `End Node - No Network Node Server`.

- Local node ID =

  The parameter is important when communicating to a subarea host computer via switched connection.

- Local node alias name = *Alias of the machine*

  It is recommended that you specify the same value as that specified as the name of the machine.

Additional SNA features

- Transaction program definitions
  - Conversation security required =

    Specify the parameter value that meets your requirements.

  - Transaction program (TP) name =

    The default assumed by the PPC server is FBSSTPNA.

  - OS/2 program path and file name = *path*\PPCRATP.EXE

    Specify the *path* according to your configuration.

  - Operation type = Non queued - Attach Manager started

This value corresponds to the only operation type to be used with LANDP. It allows for multiple occurrences of the same program to run concurrently, and no PID control is made by APPC. This enables the PPC server to issue calls for conversations remotely allocated, because APPC does not control the PID of the started TP file specification (for example, PPCRATP.EXE) and regards the server as the program previously started.

- Transaction program defaults
  - Default program directory =

    The parameter value is the OS/2 path and directory where the Attach Manager should search for programs called in *attach requests* and not defined in the Transaction Program Definitions.

- Partner LUs
  - Fully qualified LU name = *LU partner name in the network*

    The parameter value can be, for example, FBSSNET.CICS2.

  - Alias =

    The default assumed by the PPC server is FBSSLUPA.

- Local LUs
  - LU name = *LU local name in the network*

    Specify the parameter value according to your configuration.

  - Alias =

    The default assumed by the PPC server is FBSSLULO.

  - NAU address =

    The parameter value can be 00, for independent LUs, or HH for host computer dependent LUs.

- Conversation security
  - User ID and password =

    Specify the parameter value according to your choice.

- Modes
  - Mode name =

    The default assumed by the PPC server is FBSSMODE.

  - Class of service = #CONNECT
  - Mode session limit =

    The parameter value can be 1, for dependent LUs, or the value required for independent LUs (for example, 3).

  - Minimum contention winners =

The parameter value can be 0, for dependent LUs, or a higher value for independent LUs. It is recommended that the number of minimum contention winners is sufficiently high to enable all the expected Open Send requests to allocate a session.

– Receive pacing window =

  It is recommended to specify 0, unless pacing is needed.

– Maximum RU size =

  The parameter value can be, for example, 1920.

Connections, to peer node

- DLC type =

  Specify the parameter value according to your configuration.

- Link name =

  The parameter value is a name that identifies the connection.

- Partner network ID = *name of the partner network*

  The parameter value can be, for example, FBSSNET.

- Partner node name = *name of the partner machine*

  The parameter value can be, for example, FBSSHOST.

Installation requirements for TCP/IP wide area communications server

LANDP for OS/2 workstations configured to use the TCP/IP wide area communications server must have OS/2 TCP/IP installed and configured.

There must also be an EHCTCP.INI file to provide the information required to map LANDP sessions with SNA and PPC servers to TCP/IP protocols, ports and internet addresses. The file can be created with any text editor, but it is best practice to use the file as generated by LANDP customisation. See SES&TCP on page 477 for a description of the file contents.

SNA over TCP/IP dependent LUs

When using the TCP/IP wide area communications server the dependent LU names must have a common alphanumeric prefix and a numeric suffix with a constant difference between the suffix and the LOCADDR. For example:

```
IYCKT300 LU LOCADDR=2
IYCKT301 LU LOCADDR=3
IYCKT327 LU LOCADDR=29
IYCKT481 LU LOCADDR=183
```

Installation requirements for workstations with MQSeries Link servers

A significant amount of planning and configuration is required for each MQSeries Queue Manager. For more details on MQSeries administration please refer to the MQSeries documentation. For book numbers and titles, refer to “Bibliography” on page 571.

You can use the CRTMQM utility to define queue managers for use with the MQSeries Link server. When defining the queue manager, you should ensure that the queue manager has the same name as specified for the MQSeries Link server during customization.

In addition to MQSeries naming requirements, the MQSeries Link server can only accept MQSeries names that comply with the following.

- The special characters `%` and / should not be used.
- The MQSeries Queue Manager name has a maximum length of 24 characters. The characters must be upper case.
- Queue names have a maximum length of 24 characters.

The LANDP customizing process also imposes some restrictions on the naming of the MQSeries queue manager. For details, see the *MQM_name* parameter of the `LOADER EHCMQ##` statement on page 148.

A default queue manager can be defined on the MQSeries server when CRTMQM is run, by using the `-q` flag.

It is suggested that the MQSeries Queue Manager is defined to have a maximum message length similar to the LANDP MQ Server. This can be achieved by use of the MQSeries attribute **MaxMessageLength**.

The MQSeries Link server must be installed on an MQSeries Server so that the MQSeries Queue Manager is on the same workstation as the MQSeries Link server.

For security control the MQSeries Link server relies on the services of the 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 has been enabled in the LANDP API with the GQ option EHCGQ_CONVERT. When requested conversion will be attempted by the MQSeries Queue Manager.

Queues used by the MQSeries Link server should be predefined. Some sample queue definition files have been provided to aid application development. They can be found in `\EHC\EHCO500\SAMPLES`.

Installation requirements for workstations with query servers

You can use the IBM DB2 Universal Database Control Center to create and customize a database for the LANDP for OS/2 query server. The control center is part of the IBM DB2 Universal Database program suite. For information on creating and customizing a new database, refer to the *IBM DB2 Universal Database User's Guide*.

When defining the parameters for the new database, consider the following:

- **Database name**

As specified during database creation by using the control center.

The name of the new database must be the same as specified for the LANDP for OS/2 query server during customisation.

- **Maximum number of active applications**

  As customized using using the configuration database panel of the control center.

  This value should be greater than or equal to the number of processes customized in LANDP for OS/2 query server.

- **Time interval for checking deadlock (# of milliseconds)** control center.

  As customized using using the configuration database panel of the control center.

  This value should be set to 600000 milliseconds, which is the maximum number permitted by the database manager

For enhanced performance and performance tuning information, see “Tuning the LANDP for OS/2 query server” on page 252.

If you are migrating data definitions or Shared File data from a LANDP for DOS SHFILE## environment, refer to *LANDP Programming Reference*.

Installation requirements for workstations with Windows 3.1 support

If Microsoft Windows 3.1 support was specified at customization time, the EHCWIN.DLL file will be distributed to the workstation.

In addition, the multiple virtual DOS machine relay (EHCVDMGR) is required in the workstation.

Installation requirements for Java support

The LANDP Java support on OS/2 can be found in the following files located in the EHCO500 directory:

- LDPJAVA.JAR, Java Archive File containing the LANDP Java classes
- LDPJMAN.EXE, LANDP Java Manager
- LDPJDISP.EXE, LANDP Java Dispatcher
- LDPJAVA.DLL, DLL that implements the native methods of the LANDP Java classes

To support the running of Java user servers, applications, applets or servlets on a LANDP workstation, these files should be copied to the directory on the workstation containing the LANDP files.

Once copied, edit CONFIG.SYS and make the following changes:

- If not already present, add the directory containing the LANDP files to the LIBPATH statement.
- Add the following line to the CLASSPATH statement:

  ```
  <landp dir>\ldpjava.jar
  ```

where `landp dir` is the directory in which the LANDP files reside.

Modifying run-time files

The main areas of modification are:

- CONFIG.SYS contents
- AUTOFBSS.CMD contents

For a utility program to modify files, see “Modifying file contents” on page 83.

Depending on the applications and servers you have developed, and the devices you have installed, additional files may need to be copied.

When the shared-file server starts up, it creates the necessary shared-file directories if they do not already exist. This removes the need (which existed in earlier LANDP releases) to call CREADB.CMD to create these directories.

CONFIG.SYS contents

The customization program creates a CONFIG.ADD file for each workstation that requires it. You should compare the CONFIG.ADD file with the CONFIG.SYS file in your workstation, and make the necessary modifications to merge them into one.

You should specify the paths for the new device drivers and other files needed.

You should take into account that the default LIBPATH statement created during OS/2 installation is:

```
SET LIBPATH=.;C:\OS2\DLL
```

and DLLs in the current directory can be accessed.

Magnetic stripe reader/encoder server

If you plan to load this server, the following statements are typical of those required in the CONFIG.SYS file. Full details of the statements can be found in the *LANDP/DOS and LANDP/2 Support for Financial Magnetic stripe Readers/Encoders, SG24-4530*.

If the server supports a 4717 MSR/E:

```
DEVICE=FIOAUXDD.SYS /M /P
```

where `/P` should be specified only if a PIN pad server will also be loaded in the workstation.

If the server supports a 4777 MSR/E:

```
  DEVICE=FIOSERDD.SYS /Cp /M /P
```

where /P should be specified only if a PIN pad server will also be loaded in the workstation. The *p* parameter value corresponds to the COM port where the 4777 MSR/E will be attached.

If the server supports a 4778 MSR:

```
  DEVICE=FIOSERDD.SYS /Cp /P /S
```

where /P should be specified only if a PIN pad server will also be loaded in the workstation. The *p* parameter value corresponds to the COM port where the 4778 PIN pad MSR will be attached.

If a 4777 or 4778 device is mouse attached:

```
  DEVICE=FIOAUXDD.SYS /M
```

Multiple virtual DOS machine relay (MVDM)

If you plan to load this server, the following statement is required in the CONFIG.SYS file:

```
  DEVICE=<drive> <path> EHCVDMVD.SYS
```

where:

drive Is the drive on which the device is located.

path Is the directory on which the device is located.

PIN pad server

If you plan to load this server, the following statements are required in the CONFIG.SYS file.

If the server supports a 4718 PIN pad, and a magnetic stripe reader/encoder server will not be loaded in the workstation:

```
  DEVICE=FIOAUXDD.SYS /P
```

If the server supports a 4778 PIN pad, and a magnetic stripe reader/encoder server will not be loaded in the workstation:

```
  DEVICE=FIOSERDD.SYS /Cp /P
```

The *p* parameter value corresponds to the COM port where the 4778 PIN pad MSR will be attached.

If a 4778 device is mouse attached:

```
  DEVICE=FIOAUXDD.SYS /P
```

Financial printer server

If you plan to load this server, the following statement is required in the CONFIG.SYS file.

```
DEVICE=<drive> <path> OS2PRT.SYS /Dd /Bnm /Cnp
```

where:

drive Is the optional disk or diskette drive.

path Is the directory-search sequence to locate the OS2PRT.SYS file.

d Is the number of physical devices that are attached to the workstation.

The parameter value ranges from 1 to 8. The default is 1.

np Associates the logical device driver *n*. with the physical COM port *p*.

Valid values for both *n* and *p* range from 1 to 8, with a default of 1.

nm Associates the logical device driver *n* with the communication speed setting for its physical device *m*.

The parameter value for *n* ranges from 1 to 8, with a default of 1.

The parameter value for *m* is the baud rate of the device, in bits per second. Its value can be 9600, 4800, 2400, 1200, 600, 300, and 150. The default is 9600.

IBM 4712 and 4722 printers use the 4772PDD.SYS device driver (for more information see “4772 or 9068-S01 printer” on page 142).

Note: 4772 printer drivers supersede OS2PRT.SYS files. The information in the section above is retained for compatibility with older printers.

4770 printer server

If you plan to load this server, the following statement is required in the CONFIG.SYS file.

```
DEVICE=<drive> <path> COS2PRT.SYS /D:d /B:nm
```

where:

drive Is the optional disk or diskette drive.

path Is the directory-search sequence to locate the COS2PRT.SYS file.

d Is the number of physical devices that are attached to the workstation.

The parameter value ranges from 1 to 8. The default is 1.

np Associates the logical device driver *n* with the physical COM port *p*.

Valid values for both *n* and *p* range from 1 to 8, with a default of 1.

nm Associates the logical device driver *n* with the communication speed setting for its physical device *m*.

The parameter value for *n* ranges from 1 to 8, with a default of 1.

The parameter value for *m* is the baud rate of the device, in bits per second. Its value can be 9600, 4800, 2400, 1200, 600, 300, and 150. The default is 9600.

4772 or 9068-S01 printer

The 4772PDD.SYS driver must be loaded if you plan to use a 4772 or 9068-S01 printer. The 4772PDD.SYS driver can also be loaded for 4712 and 4722 printers.

The following statement is required in the CONFIG.SYS file:

```
  DEVICE=<drive> <path> 4772PDD.SYS [/Dd] [/Bnm] [/Cnp]
```

where:

drive Is the optional disk or diskette drive.

path Is the directory-search sequence to locate the 4772PDD.SYS file.

d Is the number of 4772 printers attached.

The parameter value ranges from 1 to 3. The default is 1.

nm Associates the logical device driver with the communication speed setting for its physical device:

n Is the logical device driver. The parameter value ranges from 1 to 3.

m Is the baud rate of the device, in bits per second.

The parameter value can be 9600, 4800, 2400, 1200, or 600. The default is 9600.

np Associates the logical device driver *n* with the physical COM port *p*.

The driver has to be customized if a 4770 device, and a 47X2 or a 9068-S01 printer, is to be included. The customization program 4772OCUS.EXE is on the 4772 device driver diskette, along with a detailed description in its READ.ME file, describing how and why the customization is done.

4748 printer server

The following statement is required in the CONFIG.SYS file if you plan to load this server for each 4748, 9055 model 1, or 9068-D01 printer connected to an RS-232 port:

```
  DEVICE=<drive> <path>COM.SYS
```

where:

d: Is the drive on which the device is located.

path Is the directory path of the asynchronous device driver

Note: COM.SYS is provided in OS/2 Warp Version 4 onwards as a standard device driver. (It is provided as COMDMA.SYS for PS/55, 5560, 5580, and PS/2 models 57, 90, and 95 workstations.)

If you plan to use the 9055 Model 1 or 9068-D01 printer, you can use the same device driver as for the 4748 printer (but it will provide support only for 4748 emulation mode).

If the extra features of these printers are to be used (for example, REMS), change the device driver name in your CONFIG.SYS file to:

```
DEVICE=<drive> <path>FPRTSCPx.SYS [/B:bbbb] [/X]
```

4733 teller assist unit

If you plan to use a 4733 teller assist unit, you must specify the following statement in your CONFIG.SYS file:

```
DEVICE=TCD3862.SYS
```

AUTOFBSS.CMD contents

The AUTOFBSS.CMD file contains the loading statements of the supervisor and the servers that are to reside on the workstation. The first program loaded is the supervisor.

Ensure that any modification of this file does not cause a server to be loaded before the supervisor. Application loading statements should be placed at the end of the server loading statements.

Note: To get trace and log information about the loading process, you have to load the LANDP trace tool (EHCTRACW) immediately after the supervisor.

The following is an example of a AUTOFBSS.CMD file for a workstation that contains the SNA server. The name of the workstation is AB.

```
LOADER SPV.EXE /AB
IF ERRORLEVEL 1  GOTO END
LOADER LAN.EXE
IF ERRORLEVEL 1  GOTO END
LOADER SNA##.EXE
IF ERRORLEVEL 1  GOTO END
GOTO OK
:END
EHCFREE SPV /F
EXIT
:OK
```

If the workstation is integrated in a LANDP workgroup that uses TCP/IP as transport protocol, EHCLIP.EXE should be specified instead of LAN.EXE.

You can specify the EXIT command to close the OS/2 session, and leave LANDP for OS/2 running in the background. After the :OK label, you can specify an OS/2 command to invoke a LANDP for OS/2 application.

In order to get an information window about LANDP for OS/2 status, and a LANDP for OS/2 icon, you should invoke the EHCINFO utility program. EHCINFO is a LANDP utility program; it is described in *LANDP Servers and System Management*. Alternatively, a command line version, EHCCINFO, is available.

Loading statements for LANDP for OS/2 servers

Most loading statements explained in this section are automatically created by the customization program, using the parameters you provided. The following commands appear in the AUTOFBSS.CMD provided by customization. Manual modifications can be done if need be. The LOADER command is used to run the servers and the supervisor, in the background. You can use any OS/2 command (such as DETACH or START) to run these programs since they are regular OS/2 programs. Note that using the LOADER command displays the return codes corresponding to the initialization process.

The LOADER program provides the optional parameter T to specify the timeout for a specific server load:

```
  LOADER [/T:xxxx] servername serverparms
```

The timeout value *xxxx* can be up to four characters long. The minimum is 1, the maximum 3600 seconds. If a timeout is not specified, the default used when loading the server is 30 seconds.

If the initialization works without errors, the server performs initialization, then goes into a wait state until it receives a request from an application. If an initialization error occurs, it is logged in the error log file. This file is named EHCLOG*xx*.DAT, where *xx* stands for the workstation ID.

To check the type of errors, use IF ERRORLEVEL statements. If a LANDP error occurs, the corresponding return code is provided. If an OS/2 error occurs, the value 1 is provided.

For return codes during loading, refer to *LANDP Problem Determination*.

The following commands are used when you load the LANDP for OS/2 functional areas.

Banking printer program (BPP)

The banking printer program (BPP), part of the IBM Financial Branch System Integrator/2 program, provides financial printer support that enables IBM 4700 application programs to use IBM 4712 and IBM 4722 printers (or IBM 9068 and 9069 printers in 4722 or 4712 emulator mode respectively) that are attached to the workstation.

The loading statements for BPP are described in *IBM Financial Branch System Integrator/2 Programmer’s Reference*. LANDP for OS/2 provides an extra parameter for BPP that allows printer sharing between the 4700 and local applications, as described below.

```
[/Y:nnn]
```

where:

nnn Is the interval, in seconds, for which BPP holds the printer after a write from the 4700 application has completed. The printer is then released for use by local applications. When the next 4700 request is received either to write to or set the parameters for the printer, the printer is reacquired. If the printer is in local mode and a write request is received, the previous set of device parameters is restored first.

nnn must be in the range 001 to 255. All three digits must be specified. A value of 000 means that BPP will hold the device indefinitely (as is the case without this new parameter). BPP reacquires the device automatically when a print request is received from the 4700 application.

The PRTMON.EXE utility must be installed to enable OS/2 printing to a serial-attached printer. This utility is shipped with the IBM printer drivers.

Batch machine loader server

```
LOADER BMLS.EXE /N:bmlname /P:progname [/D:workdir] [/T:yyy]
```

where:

bmlname Is the public user ID used by the object post box server.

progname Is the name of the program called by the batch machine loader server when a message is pending in the message queue.

workdir Is the working directory for the program. The default is the current directory.

yyy Is the priority for batch machine loader server processor. The parameter value can be:

1	Idle
2	Normal
3	High
4	Very high

The default is 3.

Batch machine operator

```
START BMOP [/U:userid] [/P:password] [/J:jobname]
```

where:

userid Is the user ID used to logon to the system manager. It can be a maximum of eight characters.

password Is the password used to logon to the system manager. It can be a maximum of eight characters.

jobname Requests loading of the job definition stored in the file jobname. The name and path can be a maximum of 80 characters.

CICS interface server

```
LOADER EHCTRAN.EXE
```

DDE access server

```
START EHCLAD.EXE
```

Electronic journal server

```
LOADER ELECJO##.EXE [/K:y]
```

where:

y Is the size, in KBs, of the buffer used to insert the electronic journal records. It ranges from 1 to 4. The default is 1.

The size of the buffer must be large enough to hold the maximum:

- Size of the longest electronic journal record plus 32B
- Split criteria plus 36B plus size of the selected keys. Note that you have to include the *hidden key*, which is 8-bytes long.

Forwarding server

```
LOADER FORWARD.EXE /O:vvvvvvvv.vvv [/T:wwww] [/S:xxxxxxxx.xxx] [/K:y] [/H:z] [/Z:nnnn]
```

where:

vvvvvvvv.vvv
Is the name of the file corresponding to the ASCII-to-EBCDIC translation table. It must follow the operating system rules.

wwww Is the number of time ticks after which the supervisor will dispatch the forwarding function. One time tick is roughly 0.05 seconds.

It can range from 1 to 6000. The default is 20 (about 1 second).

xxxxxxxx.xxx
Is the name of the file corresponding to the sign-on feature. It must follow the operating system rules.

y Is the size, in KBs, of the buffer used to read the store-for-forwarding records. It ranges from 1 to 4. The default is 1.

The parameter value must be the value assigned in the loading statement of the store-for-forwarding server.

z Specifies whether headers are included when sending host computer messages. The parameter value can be Y, to include headers, or N, not to include headers. The default is Y.

nnnn Is the host code page identifier (DBCS countries only). The permitted values are:

933	Korea
935 or 1388	People’s Republic of China (the default value is 935)
937	Taiwan

LAN server

```
LOADER LAN.EXE [/N:n[,n,...]] [/I:x] [/S:y] [/U:s[,s,...]]
```

where:

n[,*n*,...]

Represents the logical adapter numbers used by the LAN.EXE program. These must match the values defined for the NetBIOS protocol driver during NetBIOS configuration.

The parameter value can range from 0 to 3. The default is 0.

You can specify a series of up to four adapter numbers. This enables you to configure a server which provides services to workstations on two or more otherwise unconnected LANs. For example, `/N:0,1,2,3`, tells LAN.EXE to use all four adapters.

x

Is the time interval, in seconds, between attempts to establish the required NetBIOS sessions.

The parameter value can be 0 or in the range 5 through 3000. The default is 20 seconds. If a value of 0 is specified, only one attempt is made to establish each session, at startup, or after a session has been lost.

y

Is the NetBIOS send timeout, in seconds.

The parameter value can be 0 or in the range 10 through 127. The default is 10 seconds. A value of 0 implies no timeout.

s[,*s*,...]

Represents the number of NetBIOS sessions to be made via each adapter. The *s* values correspond to the adapters specified on the /N parameter. For example, the first *s* value after /U corresponds to the first *n* value after /N, and so on.

The parameter values can range from 0 to 254, provided sufficient sessions have been defined during NetBIOS configuration. If the parameter is omitted, or specified as 0, then enough sessions for all the related workstations are used. The total number of sessions specified must be not less than the number of related workstations.

The LAN server is needed only when more than one workstation is present in a LAN. It is included during the process of creating diskettes for distribution.

LANDP Internet Protocol

```
LOADER EHCLIP.EXE [/N:n] [/Y] [/T] [/C[:ws-id]] [/O:o[filename]] [/J]
```

where:

n

Specifies the TCP/IP port number used by LANDP Internet Protocol.

The parameter value ranges from 1024 to 65535. The default is 52699.

`Y`

Specifies that no availability probe datagram will be sent, when a session has no normal traffic.

`T` Requests LANDP Internet Protocol internal routines trace.

`C` Requests LANDP Internet Protocol communications trace. All sessions are traced, except when the *ws-id* parameter is specified. In this case, only the session with the workstation specified in that parameter is traced.

o Specifies the destination of trace data. The parameter value can be:

1 Standard output.

Note that the screen is the usual output, and requires LANDP Internet Protocol be loaded in foreground, with the DETACH or START command.

2 ASCII file. If you do not specify a name of file, the trace data is stored in the LIPTRACE.TRC file. The target file is initialized each time. Thus, if an existing file is used, the file contents get lost.

3 EHCTRACW (the default).

`J` Specifies that no checking for related workstations with undefined LANDP Internet Protocol addresses will be carried out.

Magnetic stripe reader/encoder server

```
LOADER MSRE47##.EXE
```

MQSeries Link server

```
LOADER EHCMQ## [/Q:MQM_name] [/L:lllll] [/M:m] [/P:[d:[path]]] [/T:tt] [/S:ss]
```

where:

MQM_name
Is the MQSeries Queue Manager Name.

This parameter specifies the name of the queue manager to which the server is to connect. The parameter can be a string that conforms to the rules defined for MQSeries object naming.

Alternatively, the parameter can define an environment variable, for example, '%MQMNAME%', which is resolved when the AUTOFBSS file is processed. The special characters '%' and '/' should not be used in the queue manager name.

If this parameter is omitted or entered as spaces, EHCMQ## connects to the default MQSeries queue manager.

lllll Specifies the maximum message length permitted by the MQSeries Link server. This is used to restrict the maximum message length to something less than the default of 57,000 bytes. The variable *lllll* can take a value between 1000 and 57,000.

m Specifies the message detail level to be written to the log file. The variable *m* can have the values E|W|I where E=errors, W=warnings and I=info. I includes levels W and E, W includes level E. If *m* is omitted, logging is switched off, which is the default.

d Is the drive where the log files are created. If this parameter is omitted, the drive defaults to the drive of the current working directory for the EHCMQ## process.

path Is the path where the log files are created, which defaults to `'EHCMQLOG'`. If a path is specified, a drive should also be defined. If both drive and path are omitted, `'EHCMQLOG'` is created as a sub- directory of the current working directory for the EHCMQ## process.

A log file is created for each workstation that initiates a session. The log file is named as `XXhhmmss.ddd`, where

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

The maximum length of the combined `/P` ***arguments,*** *d* ***and*** *path****, must not exceed 128 bytes***

tt Is the total number of permitted sessions.

This parameter specifies the maximum number of different sessions that the server can process at the same time. The parameter is optional, and can be in the range 1 to 64. The default is 32.

ss Is the number of MQ connections that should be made at startup. Every session requires a MQ connection, which can be made at startup or as session requests are received. The parameter is optional, the variable *ss* can be in the range 0 to *tt*. The default value is 0. When this option is non-zero, the LOADER server's timeout probably needs to be increased. Refer to "Loading statements for LANDP for OS/2 servers" on page 144.

Multiple virtual DOS machine relay

```
LOADER EHCVDMGR.EXE [path\name]
```

where:

path Is the directory where the configuration file created for the MVDM relay is located.

name Is the name of the configuration file.

The configuration file is optional. The default is EHCBOXS.CFG, if it exists.

Native X.25 server

```
LOADER X25NAT##.EXE
```

Object post box server

```
LOADER OPBS.EXE
```

PIN pad server

```
LOADER PINP47##.EXE /M
```

where `/M` indicates that 4778 magnetic stripe reader capabilities are to be used.

Program-to-program communication server

```
LOADER PPC.EXE [/FA] [/FS] [/D:nn]
```

where:

`FA` Specifies MC_FLUSH after ALLOCATE, as a result of an Open Services (OP) function.

`FS` Specifies MC_FLUSH after a Send Data (SD) function.

When the application sends data to the partner application program, the data is stored in buffers that are automatically flushed, and is immediately sent to the partner application. Note that this option may affect the workstations and the network performance.

nn Is the buffer pool that the PPC server initializes to send data to or receive data from the communications provider The default is 16.

Query server

```
LOADER EHCSQL##.EXE [/C:s] [/P:y] [/S:z [/T:x] [/F] [/W:q] [/EL]
```

where:

s Is the Database Manager database name the server will work with.

y Is the number of processes. This is the maximum number of simultaneous processes that the server will support at a given moment. It must be big enough to satisfy the application requirements. The default value is 4.

z Is the number of sessions. This is the maximum number of working sessions that a workstation is allowed to open by issuing the open session function (OS). The default value is 10. This includes the current number of different applications connected to the server at a given time (one session per application).

x Is the number of threads. This is the maximum number of requests that can be processed simultaneously. The default value is 5.

x should be slightly larger than *y* The defaults of 5 and 4 are suitable in most installations. If you are using Distributed Database Connection Services/2® (DDCS/2) to communicate with a remote DB2® database (for example, DB2 on a mainframe), it is advisable to increase the /T and /P parameters. Suggested values are 8 and 7.

`/F` Stands for fast cursor operation. If you do not include this parameter, the index search is forced by the server when issuing functions and working in shared file mode. This switch can be added if you want improved throughput without an index search in some cases.

q Specifies (in seconds) the timeout value to wait for an SQL response. The default value is 15. If a request is not satisfied within the specified time, the sentence is interrupted, and an error code appears:

RL Resources locked (shared file mode)
TE Timeout error (query mode)

If you use Query Server in Shared File mode, set the /W parameter high enough for a call to complete before it is timed out. When increasing this parameter, keep it less than the LAN timeout value. If the LAN times out first, the result of the query call is lost.

`/EL` Stands for logging enabled. If you do not include this parameter, the server does not enable logging.

At load time it is not necessary to have Database Management Services started, but if the selected database is protected with a logon password, a logon procedure must have been run previously. This can be done interactively by accessing User Profile Management, or by using the LOGON command. For example:

```
LOGON USERID / P=PASSWORD.
```

If you want to load this server, and no logon has been performed previously, the loading will not be successful and you will get a return code of RC=152.

For information on enhanced performance and performance tuning, see “Tuning the LANDP for OS/2 query server” on page 252.

Remote change management services

```
LOADER RCMS.EXE /I:xxx /O:yyy [/T:ttt] [/L:n] [/P:q] [/C:x] [/R:r]  [/Z:nnnn]
```

where:

xxx Is the file extension of the EBCDIC-to-ASCII translation table (EARCMS.*xxx*). In DBCS mode the parameter does not apply.

yyy Is the file extension of the ASCII-to-EBCDIC translation table (AERCMS.*yyy*). In DBCS mode the parameter does not apply.

ttt Is the number of timer ticks after which RCMS receives control. One timer tick is roughly 0.05 seconds. The default is 10. The value must be a decimal number in the range 1 to 999, For example:

```
/T:1, /T:2, or /T:55
```

n Specifies the number of lines in the EHCRCMS.LOG file. The parameter value can range from 100 to 10000. The default is 1000.

***Attention:* If an EHCRCMS.LOG file with *n1* lines exists already, and you choose a value for the L parameter that is different from *n1*, your old file will be destroyed and a new one created. If you want to keep the old file, rename it or copy it to somewhere else before running the LOADER program.**

q Specifies the translation mode. The parameter applies only to DBCS mode. The parameter value can be:

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

x Specifies the reception mode for CLISTs. The parameter value can be:

- `B` The CLIST is received as a binary file.
- `E` The CLIST is received as an EBCDIC file.

The default is `E`.

r Is the interval, in minutes, before retrying a connect to SNA and communications provider after a communications problem. The value must be an integer in the range 0 (which is taken to mean 30 seconds) and 1440 (24 hours). The default value is 0.

nnnn Is the host code page identifier (DBCS countries only). The permitted values are:

933	Korea
935 or 1388	People's Republic of China (the default value is 935)
937	Taiwan

Searcher

```
LOADER SFQUERY.EXE [/K:y]
```

where *y* is the size, in KBs, of the buffer used to read the electronic journal and store-for-forwarding records. It ranges from 1 to 4. The default is 1.

The parameter value must be the highest of the values assigned in the loading statements of the electronic journal and store-for-forwarding servers.

This loading statement does not correspond to a separate functional area. The SFQUERY.EXE program is required by the electronic journal and the store-for-forwarding servers.

Service Availability Manager

This loading statement does not correspond to a separate functional area. The EHCSAM.EXE program is required by shared-file servers when using the XLR (external logging replicator) facility. For details, see *LANDP Servers and System Management*.

```
LOADER EHCSAM.EXE
```

Shared-file distributor

```
LOADER EHCSFD##.EXE [/T:nnn] [/E]
```

where:

nnn Specifies the number of threads to attend and process requests in parallel.

The parameter is optional. The parameter value can range from 1 to 252. The default is 2.

/E Is an optional parameter to create a file for the statistics gathered during the session.

Shared-file replicator

```
LOADER EHCSFR##.EXE /C:confname [/T:nnn] [/E]
```

where:

confname
Specifies the name of the PCB profile to be used. The parameter is required.

nnn Specifies the number of threads to attend and process requests in parallel.

The parameter is optional. The parameter value ranges from 1 to 252. The default is 2.

/E Is an optional parameter to create a file for the statistics gathered during the session.

Shared-file server

(**Note** In this statement, the last parameter uses the letter O, not the digit 0.)

```
LOADER SHFILE##.EXE [/C:confname] [/B:nnn] [/E] [/R] [/S:xxx] [/D] [/L:y] [/F:zz]
[/X:ss] [/A:ttt] [/O]
```

where:

confname
Specifies the name of the profile that defines the shared file. If you omit this parameter, the server uses the name CONFIGUR.

nnn Specifies the number of additional 1 KB index buffers to be allocated; that is, buffers over 15. More index buffers increase system throughput, but also reduce the amount of free storage available for the server workstation. A rule of thumb is that the number of buffers should be 10 per workstation using the shared file server simultaneously. A practical limit is approximately 100, depending on available storage size. The maximum value is 968.

Another factor that must be considered is that the more buffers you have, the greater is the probability of losing index file data when the shared file server is abnormally ended. Thus, if many index buffers are allocated, and the server workstation is switched off with a transaction still in process, or if no RF function has been called, an automatic index rebuild is issued the next time the server is loaded.

/E Is an optional parameter to create a file for the statistics gathered during the session.

/R
Is an optional parameter to rebuild FREECHAIN. Use this parameter after receiving a X'A7' loading error.

xxx
Specifies the total number of additional sessions in the whole workgroup that the server can manage. The maximum is 243.

This number plus the number of workstations that receive services must not be higher than 243.

/D
Is an optional parameter to use the OS/2 default collated table. If not specified, the collated keys as defined by the configuration are used.

y
Specifies the log management type. The parameter value can be:

0 Dynamic and static log with a unique log file
1 Dynamic log with a unique log file
2 Dynamic and static log with two log files
3 Dynamic log with two log files

The default is 0.

zz
Is the number of files open at a time.

The parameter value can range from 10 to 245.

ss
Is the suffix assigned to the shared-file server. During shared-file data replication, the external logging replicator (XLR) uses this suffix to match associated active and backup servers. For example, for SHFILE01/BKFILE01, /X:01 would be specified.

ttt
Is used during shared-file data replication by the external logging replicator (XLR). It is the time delay in seconds before an automatic takeover by the backup is attempted when the active terminates. A value of zero means that no automatic takeover is attempted. This parameter is valid only for XLR servers.

Range: 0-999

/O
Overrides normal XLR initialization if only one EHCSAM server is available. This parameter has no default. Use it when only one EHCSAM server is available and then only ***when you are certain that up-to-date databases are available to that server. Do not use this parameter until you have read the following section.*** (**Note** This parameter uses the letter O, not the digit 0.)

Use of /O parameter: During XLR server initialization, LANDP:

- Determines how many EHCSAM servers are available.
- Ensures that state information (which workstation should be the active and which should be the backup) is reliable by polling all the servers.

If a majority vote of at least two SAMs cannot be achieved, LANDP issues the following messages, where *xlr* can be SH or BK.

```
EHC0587: Waiting for XLR state confirmation (Last XLR state on WS N1:
xlrFILE01).
EHC0588: Start other XLR WS, or use /O parameter on SHFILE## to override.
```

If one of the XLR workstations is permanently unavailable and no other SAM is active, the server cannot complete initialization. If you are sure that the state given in the first message is correct, and that the issuing server has up-to-date copies of the databases, you can force confirmation by restarting SHFILE## with the /O parameter. Do ***not*** use this parameter in any other circumstances.

Consider the following scenario:

```
Workstation X1                     Workstation N1

Start as Active                    Start as Backup
Process transactions               Track transactions
Machine failure                    Take over as new Active
Repair.....                        Process transactions
.....machine                       Normal shutdown
Start as  ???
```

In the last step, it is not safe for the XLR server on X1 to initialize. With /O specified, X1 would erroneously assume that it is still the active and use out-of-date data. The only thing to do is to get workstation N1 running as soon as possible. Meanwhile:

X1 waits until workstation N1 starts.
N1 initializes as the active, and X1 as the backup.
X1 catches up on the transactions that took place during its outage.
When catchup is complete, X1 is in a state to take over should N1 fail.

In the above scenario, you ***must*** wait until workstation N1 is initialized. The following is a scenario in which it is safe to use the /O parameter:

```
Workstation X1            Workstation N1

Start as Active           Start as Backup
Process transactions      Track transactions
Machine failure           Take over as new Active
Repair.....               Process transactions
      .....               Normal shutdown
      .....               Start as Active with /O parameter
```

In this case, N1 has up-to-date databases and can safely be started as the active, so the use of the /O parameter is valid. When X1 or another XLR server again becomes available, it automatically becomes the backup, bringing its log and databases up-to-date by communication with N1.

Note: After messages EHC0597 and EHC0588 are issued and before user intervention with the /O parameter, another SAM may start up. If this causes a majority vote, the chosen active XLR server continues initialization.

SNA server

```
LOADER SNA##.EXE [/S:nnn] [/W:x] [/C:1kkkkkkk] [/R:r]
```

where:

nnn Is the number of entries, divided by 50 (*nnn* * 50 = number of entries), in the communications provider LUA correlation table.

The parameter is optional. The parameter value ranges from 1 to 255. The default is 1.

x Specifies whether the wrap option for the communications provider correlation table will be used.

The parameter is optional. The parameter value can be:

Y To set the wrap option.
N Not to set the wrap option.

The default is N.

kkkkkkk

Is the master key to be used, provided cryptography is managed by the SNA server. The default is TMK*ssww*, where *ss* stands for the session ID and *ww* stands for the workstation ID.

To have cryptography managed by the SNA server, and use the default master key, specify /C.

Omit the parameter if you do not wish to have cryptography managed by the SNA server.

r Is the number of retries if a Connect to communications provider for a session returns the LINK_NOT_STARTED_RETRY message.

The value must be an integer in the range 0 (do not retry) to 8. It is advisable to specify a low value, for example 1; if the connection does not succeed after one retry, there is likely to be a problem in the communications network which needs to be resolved before the connection can succeed.

Store-for-forwarding server

```
LOADER SFORFORW.EXE [/K:y]
```

where:

y Is the size, in KBs, of the buffer used to insert the store-for-forwarding records. It ranges from 1 to 4. The default is 1.

The size of the buffer must be large enough to hold the maximum:

- Size of the longest store-for-forwarding record plus 32B.
- Split criteria plus 36B plus size of the selected keys. Note that you have to include the *hidden key*, which is 8-bytes long.

Supervisor

```
LOADER SPV.EXE /pc-id [/CL:n] [/PRIOR:pp]
```

where:

pc-id Is the identifier of the workstation that was assigned during customization. The parameter value is a string of up to 2 alphanumeric characters, and is case sensitive.

n Must be a value greater than or equal to 0, and less than or equal to 4. A value equal to 3 is not recommended. The default is 4.

This parameter is optional. It enables you to change the priority class of the SPV.EXE and all the LANDP servers. The value *n* has the same meaning and rules as the OS/2 system function DosSetPrty for the field Priority Class.

pp Must be a value greater than or equal to -31, and less than or equal to 31. The default is 15.

This parameter is optional. It enables you to change the priority level of the SPV.EXE and all the LANDP servers. The value *pp* has the same meaning and rules as the OS/2 system function DosSetPrty for the field Priority Delta.

System manager server

```
LOADER SMGR.EXE /D:x /O:yyyyyyyy  [/Z:nnnn]
```

where:

x Is the drive where the FBSS#GDT backup is located. If the parameter is omitted, the backup is not performed.

yyyyyyyy
Is the NetView operator ID. The default is OPER1.

nnnn Is the host code page identifier (DBCS countries only). The permitted values are:

933	Korea
935 or 1388	People’s Republic of China (the default value is 935)
937	Taiwan

System manager operator

```
START SMOP.EXE
```

TCP/IP wide area communications server

```
LOADER EHCTCP.EXE [/M:aaaaaaaa[,aaaaaaaa]] [/T]
```

Where:

aaaaaaaa Is the name of a LANDP communications server to be emulated. This can be SNA## to emulate the SNA server, PPC to emulate the PPC server, or both.

`/T` Is an optional parameter to turn on internal tracing for problem determination.

Trace tools

`LOADER EHCTRACW [/R:`*rrrr*`][/B:`*bbb*`][/T:`*xxx*`][/MT:`*mmm*`][/ML:`*mmm*`][/LT:`*lll*`][/LL:`*lll*`]`
`[/PT:[`*d*`:] [`*path*`]`*filename*`][/PL:[`*d*`:][`*path*`]`*filename*`]`

where:

rrrr Is the record length in shared memory.

The variable *rrrr* can take values between 64 and 1024 + 64. The default is 394.

bbb Is the maximum number of records in shared memory.

The variable *bbb* can take values greater than 1. The minimum is 2, and the default is 162.

Note that the maximum number of *bbb* is calculated using the formula:

[(64 × 1024 - 64) ÷ (record length *rrrr*)]

xxx Is the trace option. There are three options for /T (trace facility):

`NO`	No trace is provided.
`MEMORY`	Trace records provided in memory only.
`FILE`	Trace records stored in both memory and in the file specified by the /PT parameter.

The default is MEMORY. This parameter does not affect the log file, because a log error file is always provided by the server.

mmm Is the maximum number of records for Trace file (/MT) and Log file (/ML). The variable *mmm* can take values between 1 and 50000. If there is not enough space available, an error will be returned. The default in both cases is 512.

You must erase the existing Trace or Log files when you are creating new ones, otherwise the new parameters will not take effect.

Note that you must specify /T:FILE in the loading statement when you create a new Trace file.

lll Is the maximum record length for Trace file (/LT) and Log file (/LL). The variable *lll* must be less than or equal to *rrrr*. The minimum value is 128.

The default for Trace is 394. The default for Log is 150.

d Is the drive where the Trace file and the Log file will be created.

path Is the path where the Trace file (/PT) and the Log file (/PL) will be created. The path must be less than 128 bytes.

filename

Is the name of the Trace file (/PT) and the Log file (/PL). The default name is EHCTRC*xx*.DAT for Trace, and EHCLOG*xx*.DAT for Log. In both cases, *xx* is the workstation identifier.

If EHCTRACW is loaded while the supervisor is still in the loading process, the workstation ID value *xx* in the filenames EHCTRC*xx* and EHCLOG*xx* will sometimes be given the value [!!]. This may happen if you load with the DETACH or START command. To avoid this, you can either use the LOADER command, or rename the *xx* value for the files afterwards.

Financial printer server

```
LOADER PR47X2##.EXE [/K:n]
```

where:

n Is the maximum number of KB to be printed at a time.

The parameter is optional. The parameter value ranges from 1 to 4. The default is 1.

4748 and 9068-D01 printer server

```
LOADER PR4748##.EXE [/K:n]
```

where:

n Is the maximum number of KB to be printed at a time.

The parameter is optional. The parameter value ranges from 1 to 4. The default is 1.

To change the baud rate of the device, you can issue a MODE OS/2 command before loading the server. Specify 9600, 4800, 2400, 1200, 600, 300, or 150. The default is 1200 bits per second.

4770 printer server

```
LOADER PR4770##.EXE [/K:n]
```

where:

n Is the maximum number of KB to be printed at a time.

The parameter is optional. The parameter value ranges from 1 to 4. The default is 1.

Loading statements for emulators in an OS/2 MVDM

If you select MVDM relay during customization, and define the emulators to be loaded in an OS/2 MVDM, the customization program creates an AUTOFBSS.BAT file with the emulator loading statements. The LSI command is used to run the emulators in the OS/2 MVDM.

When you install more than one emulator, they can run in different OS/2 MVDMs. A separate AUTOFBSS.BAT file is required for each MVDM, containing the loading statements for the emulators that run in that particular MVDM. To create those files, copy only the required loading statements.

FBSS (DOS) and LANDP for DOS applications can also be run in an OS/2 MVDM.

Banking interactive workstation program

```
LSI VBIWP.EXE /C:atr /K:kbd /D:dis /T:at1   /P:pin /M:msi /N:mso /I:n /F:nnn
```

where:

atr Is the name of the selected display color attributes table.

kbd Is the name of the selected keyboard ASCII-to-EBCDIC translation table.

dis Is the name of the selected display EBCDIC-to-ASCII translation table.

at1 Is the model and attribute definitions.

pin Is the name of the selected PIN pad input table.

msi Is the name of the selected MSR/E input table.

mso Is the name of the selected MSR/E output table.

n Is the emulator identification number.

nnn Applies only to BIWP running in a DOS MVDM under OS/2, and enables MSR/E and PINPad sharing between the 4700 and local applications. The BIWP window has the focus while it is active and loses the focus when inactive.

nnn is the interval, in seconds, for which BIWP holds the device after its window has lost the focus. The value specified must be in the range 001 to 255. All 3 digits must be specified. A value of 000 means that BIWP will hold the device indefinitely (as is the case without this parameter). BIWP reacquires the device automatically when it regains the focus.

A user server, WINFOCUS.EXE, must be installed under OS/2. The server is shipped with LANDP for OS/2 but the following steps are necessary to install it:

1. Add the following line to LANCONF.SPC

   ```
   SERVER=WINFOCUS,
   ```

2. Define as a user server in COMMON.SPC. For example:

   ```
   DEFSERV NAME=WINFOCUS,
           TYPE=OS/2,
           SCOPE=BOTH,
           DESCRIPTION='Check Window Focus',
           OBJECT=WINFOCUS.EXE,
           SUBDIR=EHCO000,
           LOADER=LOADER,
           PRIORITY=3,
           LANUNIQ=N,
           ALLCLI=N
   ```

3. Load the user server with LOADER.EXE

The version of the BIWP server must be 10026 or later.

3270 emulator

```
LSI EMU3270.EXE /C:atr /K:kbd /D:dis /I:n  [/H:hh] [/W:www] [/B:y] [/S:xxxxxxxx]
 [/Z:nnnn] [/P:a] [/L:l] [/T:tt]
```

where:

atr Is the name of the selected display color attributes table.

kbd Is the name of the selected keyboard ASCII-to-EBCDIC translation table.

dis Is the name of the selected display EBCDIC-to-ASCII translation table.

n Is the emulator identification number.

hh Specifies the alternate screen height (number of rows) of the 3270 display to be emulated. (The height specified should not include the operator information area line at the bottom of the emulator screen.) *nn* must be in the range 24 through 49. For 132-column screens, the maximum height might be limited by the capabilities of the PC video display adapter installed in your system.

Use this parameter, in conjunction with /W, to make the emulated alternate screen look like one of the following 3270 models:

3270 model	Alternate screen height	Alternate screen width
2	24	80
3	32	80
4	43	80
5	27	132

If this parameter is omitted, the default is 24.

www Specifies the alternate screen width (number of columns) of the 3270 display to be emulated. *nnn* must be either 80 or 132. Some PC video display adapters do not support 132-column mode.

If this parameter is omitted, the default is 80.

y Indicates whether blinking is supported. Specify Y for yes or N for no.

If this parameter is omitted, the default is N.

xxxxxxxx
Specifies the long name of the 3270 emulator session (sometimes known as the “host session ID”), which is displayed in the operator information area on the screen. You can specify up to eight characters (with no imbedded blanks).

If this parameter is omitted, the default is a name of eight blanks.

nnnn Specifies the size (in bytes) of the buffer used to communicate with the host. *nnnn* can be any value in the range 2048 through 4096. Specifying a small buffer size minimizes memory requirements; using a large buffer can reduce the number of transmissions needed to send or receive a large data stream. The parameter value specified must match the RU size detailed in the bind session.

If this parameter is omitted, the default is 2048.

a Indicates whether the 3270 emulator should handle the Print Screen key. Specify N for no or Y for yes.

If this parameter is omitted, the default is Y.

l Indicates whether the SNA session is connected at emulator load time rather than at ‘hot key’ time. Specify N for NO or Y for YES. The default is N.

tt Specifies the minimum time in seconds between checks on ‘print screen’ key presses. The default is 3.

Note: When working with LANDP for OS/2 workstations, the 3270 emulator can be used only in an OS/2 MVDM. When working in DBCS mode, the 3270 emulator cannot be used in an OS/2 MVDM, and thus it cannot run on a LANDP for OS/2 workstation.

3287 printer emulator

```
LSI EMU3287.EXE /x /E:prt [/T:nn] [/P:HP] [/N:n]
```

where:

x Can be `S` or `M`. Select `S` for single and `M` for multiple LU_1 support.

Note: This parameter and its values are no longer supported, though they will be accepted for compatibility purposes with earlier versions of LANDP. If specified, they will be ignored.

prt Is the name of the selected EBCDIC-to-ASCII translation table.

nn Is the frequency of polling.

The parameter value ranges from 1 to 60. The default is 15.

`/P:HP` Indicates that the 3287 printer emulator will use either an IBM 4019 Printer, an IBM 4029 Printer, or an IBM 4039 Printer for output. The default is to use an IBM 4201 Proprinter® or equivalent device.

`/N:n` Specifies the range of logical printer numbers that can be used. The parameter value can be in the range 1 through 3. Specify:

- `/N:1` to use LPT1 only
- `/N:2` to use LPT1 and LPT2
- `/N:3` to use LPT1, LPT2, and LPT3

If the parameter is not specified, the emulator uses the number of parallel printer ports physically installed on the workstation.

Note: When working with LANDP for OS/2 workstations, the 3287 printer emulator can be used only in an OS/2 MVDM. When working in DBCS mode, the 3287 printer emulator cannot be used in an OS/2 MVDM, and thus it cannot run on a LANDP for OS/2 workstation.

Loading servers inline

LANDP for OS/2 provides a utility program to load the LANDP emulators supported by the multiple virtual DOS machine relay and the EXFS user server support (for the LANDP 3287 printer emulator) in virtual DOS machines of OS/2.

The load servers inline (LSI) utility program can also load the FBSI banking interactive workstation program (BIWP), if the FBSI is installed. The LSI program is called as follows:

```
LSI [/K:m] [/X] [/N:n] [/P:zz] servername serverparms
```

where:

m
: Is the entry hot key for the display emulators. If you are loading the 3270 emulator or the BIWP and you do not specify this parameter, the customized value will be used.

X
: Prevents intercepting hardware interrupts. Use this parameter only if you have coexistence problems in some environments.

n
: Is the alias server name.

zz
: Is the PCID for requests passed by LSI. The default is ' '.

servername
: Is the name of the emulator or user server to be loaded.

serverparms
: Are the parameters available for the emulator to be loaded; they depend on the emulator. For information on the LANDP emulator parameters that can be included in the loading statement, see earlier parts of this chapter.

Use the following command to unload LSI and the emulators from memory:

```
LSI /U
```

See the *LANDP Programming Reference* book for more information about the EXFS user-written server support for 3287 printer emulation.

Unloading LANDP for OS/2

The EHCFREE.EXE utility is provided to unload LANDP for OS/2. You can also unload LANDP for OS/2 by issuing a supervisor function call from an application program. For more information on supervisor function calls, see *LANDP Programming Reference*.

The LANDP for OS/2 utility, EHCFREE.EXE, is called as follows:

```
d:\path\EHCFREE SPV [/p]
```

where:

d:
: Is the drive where the utility is located.

path
: Is the path where the utility is located.

p
: Is an optional parameter that can take the following values:

- For weak unload, *p* = W.

  This is the default value, and requires that the supervisor function unload LANDP for OS/2 (**ES**) is issued to the LANDP components.

For information on supervisor function calls, refer to the *LANDP Programming Reference*.

- For forced unload, `p = F`.

  This requires that LANDP for OS/2 unloads the LANDP components through the system functions provided to cancel an OS/2 process.

The EHCFREE.EXE program errors are detected by the LAN server or the supervisor. For information on the corresponding return codes, refer to the *LANDP Problem Determination* manual.

Unloading LANDP for OS/2 servers

The EHCFREE.EXE utility program is used to dynamically unload a LANDP for OS/2 server at LANDP run time. The command must be entered in the workstation where the specific LANDP for OS/2 server to be unloaded is located.

Note that the EHCFREE.EXE program can also be used to unload the entire LANDP for OS/2 program.

The EHCFREE.EXE program is called as follows:

```
d:\path\EHCFREE servername [/p]
```

where:

d: Is the drive where the utility is located.

path Is the path where the utility is located.

servername
Is the name of the server to be unloaded, entered in upper or in lower case. Note that if you specify `SPV`, the entire LANDP for OS/2 is unloaded.

p Is an optional parameter, which can take the following values.

- For weak unload, `p = W`

  This is the default value, and requires that the supervisor function unload LANDP for OS/2 (**ES**) is issued to the server specified.

  For information on supervisor function calls, refer to the *LANDP Programming Reference*.

- For forced unload, `p = F`

  This requires that LANDP for OS/2 ends the servers specified, through the system functions provided to cancel an OS/2 process.

The EHCFREE.EXE program errors are detected by the LAN server or the supervisor. For information on the corresponding return codes, refer to the *LANDP Problem Determination* manual.

Installing run-time files

The LANDP family provides a utility program to check the path where the run-time files are located. See “Installing and validating system files” on page 236 for more information about this program.

For other utility programs also provided to be used at run-time, refer to Chapter 15, “Run-time utility programs” on page 229.

The customization program creates the EHC.MSG message file, which follows the OS/2 rules. The file is provided only for the workstations with:

- RCMS
- MVDM relay device drivers
- Electronic journal utility programs
- Shared file utility programs
- Query server utility programs
- EHCVAL program
- System manager
- System manager operator
- Service availability manager

The EHC.MSG file must be located in the current path, or in a path specified using a DPATH statement in the CONFIG.SYS file. Note that, for device drivers, the current path is the root directory.

Chapter 13. Preparing Windows NT workstations

The first part of the chapter, “Installing and configuring Windows NT workstations,” describes the requirements to install the LANDP for Windows NT run-time files on the Windows NT workstations. Some requirements are related to a specific server.

The second part of the chapter, “Modifying run-time files” on page 180, describes the process of modifying the run-time files created by the customization program, according to your needs.

The third part of the chapter, “Loading statements for LANDP for Windows NT servers” on page 182, describes how to check for a proper installation of the files.

Installing and configuring Windows NT workstations

The diskettes created for the Windows NT workstations do not contain any Windows NT system files. Therefore, before installing the LANDP for Windows NT operational diskettes into your Windows NT workstation, make sure that Windows NT V4.0 or Windows 2000 is installed according to the existing recommendations.

The NetBIOS or TCP/IP support that LANDP for Windows NT requires is provided as part of Microsoft Windows NT Version 4.0 or Windows 2000.

The SNA support that LANDP for Windows NT may require is provided as part of one of the following products:

- IBM SecureWay Communications Server for Windows NT, Version 6.0
  - Version 6.02 is a pre-requisite for DBCS support
  - Version 6.1 is a pre-requisite for Windows 2000 support
- IBM SecureWay Personal Communications Version 4.31 for DOS/Windows, Windows 95 and Windows NT
- Microsoft SNA Server, Version 4

When working in DBCS mode, the operating system can be one of the following depending on your national language:

Language	**Operating system**	**Supported codepage**
Traditional Chinese	Microsoft Windows NT V4 (Traditional Chinese)	950 (no defaults)
Simplified Chinese	Microsoft Windows NT V4 (Simplified Chinese)	1386 (no defaults)

Note: Windows NT (Korean) is not supported by LANDP for Windows NT.

Installation requirements for NetBIOS transport protocol

This section applies to LANDP for Windows NT workstations integrated in a LANDP workgroup that uses NetBIOS as the transport protocol.

Important

The NetBIOS interface must be installed and it is recommended that you configure it so that Lana (LAN adapter) Number 0 is assigned to the Network route starting with Nbf-> because LANDP uses adapter number 0 by default. (The NetBIOS interface can be configured by selecting the appropriate items within the Network function of the Windows NT Control Panel.)

For information on installation and configuration on Windows 2000, see the READ.ME file.

If you choose to use the ADAPTNUM keyword in LANCONF.SPC (see “LANCONF vector” on page 411) or the /N parameter on loading LAN.EXE (see “LAN server” on page 185) to specify other adapter numbers, the NetBIOS interface must be configured so that these Lana (LAN adapter) Numbers are assigned to the required network routes.

In a workgroup with OS/2 or DOS machines as well as Windows NT machines, you should change the default NETBIOS timing parameters, which are different to those provided by IBM’s NetBios support. The T1, T2 and TI values should be the same on all machines. On Windows NT machines, these values are in the registry under the following subkey:

```
HKEY_LOCAL_MACHINE\SYSTEM\Services\NBF\Parameters
```

Installation requirements for TCP/IP transport protocol

This section applies to LANDP for Windows NT workstations integrated in a LANDP workgroup that uses TCP/IP as the transport protocol.

TCP/IP must be configured in all the workstations so that you can use TCP/IP as the internal communications protocol for a LANDP workgroup.

For detailed information, refer to the TCP/IP information in the Microsoft Windows NT online documentation.

For more information about the TCP/IP transport protocol, see see Appendix E, “Using TCP/IP for internal communication” on page 525.

Installation requirements for workstations with SNA servers

The LANDP for Windows NT SNA server uses the LUA interface that is provided by one of the following products:

- IBM SecureWay Communications Server for Windows NT, Version 6.0
  - Version 6.02 is a pre-requisite for DBCS support
  - Version 6.1 is a pre-requisite for Windows 2000 support

- IBM SecureWay Personal Communications Version 4.31 for DOS/Windows, Windows 95 and Windows NT
- Microsoft SNA Server, Version 4

The workstation configured to run the SNA server must be the one that is connected to the host computer. All logical units types 0, 1, and 2 receiving services from the LANDP SNA server must be defined in the SNA provider of the workstation that is connected to the host computer.

You define the workstation profile and the SNA profiles in the configuration files of the SNA provider, as follows:

- For IBM SecureWay Communications Server for Windows NT, and IBM SecureWay Personal Communications for DOS/Windows, Windows 95 and Windows NT

  Select and add appropriate information in:

  – SNA Node Operations
  – Node
  – Launch
  – SNA Node Configuration
  – New or Open
  – Scenarios
  – Advanced
  – Configure Node, with:
    - Devices
    - Connections
    - Local LU 0 to 3
  – Configure Devices

- For Microsoft SNA Server Version 4: use the SNA Admin function.

  Select and add appropriate information in:

  – Microsoft SNA Server (Common)
  – Manager
  – Domain (which will already have a domain name present from the LANDP installation process)
  – Servers
  – SNA Service (Active)
  – Link Services
  – Connections
  – 3270 (to define profiles)
  – Application LU (LUA)

If you do not use LU pooling support, you have to define as many LUA profiles as SNA sessions.

If you plan to use LU pooling support, you have to group LUA profiles in LU pools. In this case, the number of LUA profiles depends on your configuration requirements.

The profile name is a string of eight characters, which must comply with the following conventions:

```
EHCxxnnb
```

where:

xx Has two different meanings:

- If you do not use LU pooling support, *xx* stands for the workstation ID of the workstation that uses the session. The parameter value is a string of two alphanumeric characters.

  If the workstation ID has only one character followed by a blank, the profile name you have to define is `EHCx@nnƀ`.

- If you use LU pooling support, *xx* stands for the pool ID assigned to the session during customization.

nn Is the SNA session identifier.

- If you do not use LU pooling support, see below for the correct values.
- If you use LU pooling support, you should number the identifiers from 01 to 99, sequentially. If you skip one number, only the identifiers specified before will be considered. For example, if you specify 01, 02, 03, 05, ... , 50, only the SNA sessions 01, 02, and 03 will be considered.

b Is one character you leave blank.

The variable *nn* takes the following values.

For the 3270 emulator:

You can have five sessions running simultaneously in each workstation, with the identifiers numbered from 51 (for the first session) through 55 (for the fifth session). That is, the value for *nn* for the 3270 emulator ranges from 51 to 55.

For the 3287 printer emulator:

You can have five sessions running simultaneously in each workstation, with the identifiers numbered from 76 (for the first session) through 80 (for the fifth session). That is, the value for *nn* for the 3287 printer emulator ranges from 76 to 80.

For RCMS (for LANDP for DOS, OS/2, and Windows NT):

The value for the variable *nn* for RCMS is 33.

For Forwarding (for LANDP for DOS, OS/2, and Windows NT):

You can have three sessions running simultaneously, with the identifiers numbered from 37 (for the first session) through 39 (for the third session).

For applications:

For DOS applications, the variable *nn* can take the values of 1 through 15. For DOS user servers, it can take the values of 16 through 30.

For OS/2 or Windows NT applications or user servers, it can take the values 1 through 30. However, 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. The SNA session identifier part of the LUA profile name (LU0 through LU3) (defined under the appropriate SNA communications provider) may now be any two ASCII characters instead of just two decimal digits.

For 4731 and 4737 (for LANDP for DOS):

The variable *nn* can take the values 34 and 35 for the 4731 Personal Banking Machine (PBM), and the value 34 only for the 4737 PBM.

For BIWP (Running in OS/2 MVDM and PC/Integrator):

The variable *nn* takes the value 63. For the value to specify when BIWP runs in OS/2, refer to the PC Integrator/2 manuals.

For the LDA 7 Program (Running in OS/2 MVDM and PC/Integrator):

The variable *nn* takes the value 64. For the value to specify when the LDA 7 program runs in OS/2, refer to the PC Integrator/2 manuals.

Following is an example definition of the SNA LUA profile names for a given workgroup, for which LU pooling is not used:

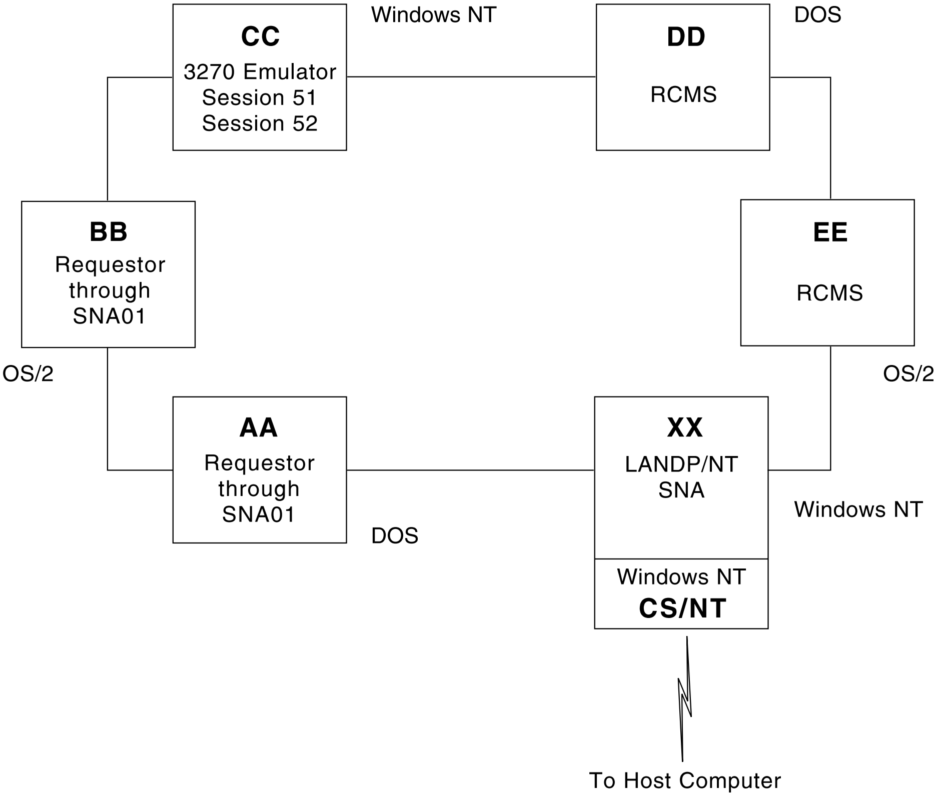


Figure 6. Example of SNA LUA profile names in a workgroup (Windows NT)

These are the expected profile names in the SNA provider for workstation XX (gateway):

EHCAA01	For the client workstation AA
EHCBB01	For the client workstation BB
EHCCC51	For the first emulator session in workstation CC
EHCCC52	For the second emulator session in workstation CC
EHCDD33	For the RCMS client workstation DD
EHCEE33	For the RCMS client workstation EE

You then specify the following profile parameters:

- Comment (optional)
- LU Number
- DLC type

Note: If you use *correlation tables*, ensure that the SNA provider's LUA level supports the corresponding definitions.

Installation requirements for TCP/IP wide area communications server

LANDP for Windows NT workstations, when configured to use the TCP/IP wide area communications server, must have Windows NT TCP/IP installed and configured.

There must also be an EHCTCP.INI file to provide the information required to map LANDP sessions with SNA and PPC servers to TCP/IP protocols, ports and internet addresses. The file can be created with any text editor, but it is best practice to use the file as generated by LANDP customisation. See SES&TCP on page 477 for a description of the file contents.

SNA over TCP/IP dependent LUs

When using the TCP/IP wide area communications server the dependent LU names must have a common alphanumeric prefix and a numeric suffix with a constant difference between the suffix and the LOCADDR. For example:

```
IYCKT300 LU LOCADDR=2
IYCKT301 LU LOCADDR=3
IYCKT327 LU LOCADDR=29
IYCKT481 LU LOCADDR=183
```

Installation requirements for workstations with PPC servers

To run the PPC server, you have to provide some definitions for the communications provider configuration.

The definitions that can be involved are shown in this section. However, you do not need to specify all the definitions in all the cases, due to the communications provider dynamic configuration.

Most definitions and the resulting configuration should be regarded as a pattern to define your own configuration.

For the workstation profile and automatic start options.

Translation table

The translation table specified in the workstation information option can be ACSGTAB.DAT.

Local node characteristics

- Network ID = *Name of the network*

  The parameter value can be, for example, FBSSNET.

- Local node name = *Name of the machine*

  The parameter value can be, for example, LAN1WS01.

- Node type =

  The default is `End Node - No Network Node Server`.

- Local node ID =

The parameter is important when communicating to a subarea host computer via switched connection.

- Local node alias name = *Alias of the machine*

  It is recommended that you specify the same value as that specified as the name of the machine.

Additional SNA features

- Transaction program definitions
  - Conversation security required =

    Specify the parameter value that meets your requirements.
  - Transaction program (TP) name =

    The default assumed by the PPC server is FBSSTPNA.
  - Windows NT program path and file name = *path*\PPCRATP.EXE

    Specify the *path* according to your configuration.
  - Operation type = Non queued - Attach Manager started

    This value corresponds to the only operation type to be used with LANDP. It allows for multiple occurrences of the same program to run concurrently, and no PID control is made by APPC. This enables the PPC server to issue calls for conversations remotely allocated, because APPC does not control the PID of the started TP file specification (for example, PPCRATP.EXE) and regards the server as the program previously started.
- Transaction program defaults
  - Default program directory =

    The parameter value is the Windows NT path and directory where the Attach Manager should search for programs called in *attach requests* and not defined in the Transaction Program Definitions.
- Partner LUs
  - Fully qualified LU name = *LU partner name in the network*

    The parameter value can be, for example, FBSSNET.CICS2.
  - Alias =

    The default assumed by the PPC server is FBSSLUPA.
- Local LUs
  - LU name = *LU local name in the network*

    Specify the parameter value according to your configuration.
  - Alias =

    The default assumed by the PPC server is FBSSLULO.
  - NAU address =

The parameter value can be `00`, for independent LUs, or `HH` for host computer dependent LUs.

- Conversation security
  - User ID and password =

    Specify the parameter value according to your choice.
- Modes
  - Mode name =

    The default assumed by the PPC server is FBSSMODE.
  - Class of service = `#CONNECT`
  - Mode session limit =

    The parameter value can be 1, for dependent LUs, or the value required for independent LUs (for example, 3).
  - Minimum contention winners =

    The parameter value can be 0, for dependent LUs, or a higher value for independent LUs. It is recommended that the number of minimum contention winners is sufficiently high to enable all the expected Open Send requests to allocate a session.
  - Receive pacing window =

    It is recommended to specify 0, unless pacing is needed.
  - Maximum RU size =

    The parameter value can be, for example, 1920.

Connections, to peer node

- DLC type =

  Specify the parameter value according to your configuration.
- Link name =

  The parameter value is a name that identifies the connection.
- Partner network ID = *name of the partner network*

  The parameter value can be, for example, FBSSNET.
- Partner node name = *name of the partner machine*

  The parameter value can be, for example, FBSSHOST.

Installation requirements for workstations with MQSeries Link servers

A significant amount of planning and configuration is required for each MQSeries Queue Manager. For more details on MQSeries administration please refer to the MQSeries documentation. Book numbers and titles can be found in “Bibliography” on page 571.

You can use the CRTMQM utility to define queue managers for use with the MQSeries Link server. When defining the queue manager, you should ensure that the queue manager has the same name as specified for the MQSeries Link server during customization.

A default queue manager can be defined on the MQSeries server when CRTMQM is run, by using the `-q` flag.

It is suggested that the MQSeries Queue Manager is defined to have a maximum message length similar to the LANDP MQ Server. This can be achieved by use of the MQSeries attribute **MaxMessageLength**.

The MQSeries Link server must be installed on an MQSeries Server so that the MQSeries Queue Manager is on the same workstation as the MQSeries Link server.

In addition to MQSeries naming requirements, the MQSeries Link server can only accept MQSeries names that comply with the following.

- The special characters `%` and `/` should not be used.
- The MQSeries Queue Manager name has a maximum length of 24 characters. The characters must be upper case.
- Queue names have a maximum length of 24 characters.

For security control the MQSeries Link server relies on the services of the 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 has been enabled in the LANDP API with the GQ option EHCGQ_CONVERT. When requested conversion will be attempted by the MQSeries Queue Manager.

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 in `\EHC\EHCN500\SAMPLE`.

Installation requirements for workstations with shared-file servers

If you are migrating data definitions or Shared File data from a LANDP for DOS SHFILE## environment, refer to *LANDP Programming Reference*.

Installation requirements for workstations with ODBC query servers

The workstation must have the Windows NT ODBC Driver Manager and an ODBC Device Driver installed. The ODBC Device Driver must:

- Conform to ODBC version 3.0 and Level 1 API
- Be configured to use the database that is to be accessed
- Be registered as a system name data source

For instructions on installing the ODBC Driver Manager, refer to the Microsoft Windows NT documentation. For instructions on installing the ODBC Device Driver, refer to the vendor's instructions.

Installation requirements for Microsoft Visual Basic

On a machine that needs to use Microsoft Visual Basic, take the following steps to install the OCX.

1. Copy the `landp.OCX` file from the EHCN500 directory to the `\system32` directory, which should be under the Windows install directory, conventionally `c:\winnt\system32`.

   Note: The EHCN500 directory is on the customization machine and the target is probably a different machine. Use a diskette or the LAN to transfer the file. It would be helpful to copy the LANDP.HLP file.

2. Run the registration program `regsvr32` from the `system32` directory with the parameter `Landp.OCX`. For example:

   ```
   c:\winnt\system32 regsvr32 Landp.OCX
   ```

   Alternatively you can run regsvr32 from anywhere as long as you specify the fully qualified path name of where you placed the OCX. For example:

   ```
   x: regsvr32 c:\directory1\directory2\Landp.OCX
   ```

3. You should get a pop up window saying the registration succeeded.

   An error code of 7e usually indicates that the OCX cannot be found. Check that the path name been specified fully and correctly, and that the OCX is in the current directory.

4. Run Visual Basic. You should see the `Landp.OCX` referred to in the Custom Controls panel under the Tools directory.

5. Check the box next to the LANDP custom control statement and an icon should appear in the Controls toolbar.

6. Select the icon and place the control onto the form.

Installation requirements for Java support

The LANDP Java support on Windows NT can be found in the following files located in the EHCN500 directory:

- LDPJAVA.JAR, Java Archive File containing the LANDP Java classes
- LDPJMAN.EXE, LANDP Java Manager
- LDPJDISP.EXE, LANDP Java Dispatcher
- LDPJAVA.DLL, DLL that implements the native methods of the LANDP Java classes

To support the running of Java user servers, applications, applets or servlets on a LANDP workstation these files should be copied to the directory on the workstation containing the LANDP files.

Once copied, two system environment variables need to be updated. This can be done from the Control Panel System icon.

- If not already present, add the directory containing the LANDP files to the PATH statement.
- Add the following line to the CLASSPATH statement:

  ```
  <landp dir>\ldpjava.jar
  ```

  where `landp dir` is the directory in which the LANDP files reside.

Running LANDP DOS applications on Windows NT

The MVDM Relay DOS stub (EHCVDSPV.COM) must be loaded in any VDM that will run LANDP/DOS applications. This can be achieved in a number of ways, some of which are covered here. The most appropriate way depends on the way the workstation is used and the applications that will be used.

Firstly, an understanding of the Windows NT MS-DOS command window is required.

When an MS-DOS window is first opened, it runs the CMD.EXE command interpreter. This allows the invocation of Windows programs, OS/2 16 bit programs and DOS programs. When a DOS program is detected, a new process is created and COMMAND.COM is used to start the program. Before running the DOS program, CONFIG.NT and AUTOEXEC.NT from the SYSTEM32 subdirectory of the Windows directory (usually WINNT) are processed. When the DOS program ends, CMD.EXE regains control. Subsequent programs run under either CMD or COMMAND, depending on their type.

CONFIG.NT and AUTOEXEC.NT are the default files used for MS-DOS windows and WOW (Win16 On Win32) applications. However, different files may be specified using a PIF (Program Information File). Refer to Windows NT documentation for details.

Note: For naming PIF startup files, Windows NT and Windows 2000 use different conventions. Under Windows NT, the convention is AUTOEXEC.*nnn* and CONFIG.*nnn* where *nnn* is chosen by the user. Under Windows 2000, the convention is *nnnnnnnn*.NT where *nnnnnnnn* is chosen by the user.

This ability to have different start-up files can be used in a number of ways to implement an MVDM relay.

If the workstation will only be used for specific LANDP for DOS applications, it may be appropriate to invoke EHCVDSPV in-line with the application using a batch file as in the following example:

```
STARTAPP.BAT

         LOADHIGH EHCVDSPV.COM
         LANDPAPP
         EXIT
```

In this case the MVDM DOS stub is loaded into high memory and LANDP for DOS is started. The window is closed on termination of the application. No changes to CONFIG.NT or AUTOEXEC.NT are required.

If 16 bit Windows applications are to be run and/or all MS-DOS prompts have the capability to run LANDP for DOS applications, the following should be added to AUTOEXEC.NT:

```
D:
CD \LANDPV5
EHCVDSPV
```

(Assumes the LANDP runtime is in D:\LANDPV5)

In some circumstances it may be necessary to create a PIF for LANDP for DOS command prompts because of disruptions to the EHCVDSPV TSR caused by non-DOS programs. In the following example the PIF file for a shortcut to COMMAND.COM called "LANDPDOS" is modified to use CONFIG.LDP and AUTOEXEC.LDP. These files are copies of CONFIG.NT and AUTOEXEC.NT respectively, with the following additions:

```
Appended to CONFIG.LDP

   DOSONLY

Appended to AUTOEXEC.LDP

   D:
   CD \LANDPV5
   EHCVDSPV
```

Note that only DOS programs will be able to run in this window.

EHCVDSPV optional parameters

/Q Quiet mode; no copyright messages issued.

/I Initialize immediately. Requires that LANDP for Windows NT has been started already. By default, initialization is delayed until the first LANDP API call.

/U Unload a previously installed copy of EHCVDSPV if safe to do so.

Loadhigh may be used to load EHCVDSPV. Refer to the Windows NT help for details.

16 Bit Windows LANDP applications

16 bit Windows LANDP applications developed for Windows 3.x may be run on Windows NT. AUTOEXEC.NT must be modified as in the previous section and EHCWIN.DLL must be present in the LANDP runtime directory.

Only one LANDP/DOS application may be run in any one WOW (Win16 on Win32) VDM. Multiple applications may, however, be run if they are run in a separate VDM (See Windows documentation).

SVPCPRBW may be used to verify correct installation of Win16 support.

Running LANDP/NT servers as Windows NT Services

Under Windows NT, programs are normally run by, and associated with, one or more users. If the user logs off, the program is terminated. As this is not desirable in some environments, there is a facility to run programs as services. By default, LANDP Servers run as services. This imposes some restrictions:

- Servers run as services must reside on a local drive. That is, they cannot be on a network drive.
- Each service name must be unique (the executable name in the case of a LANDP server).
- The directory of the executable is recorded in the Registry and will only be executed from that directory.
- Only one instance of the service can be run at any one time on a workstation.
- The server cannot directly interact with the user.
- By default, the service runs under the Local System Account and uses its access authority. (The services applet in the control panel may be used to change the account.)
- All supplied servers can be run as normal application programs if any of these limitations is not acceptable. However, the user must remain logged on while LANDP is active.

Modifying run-time files

The main areas of modification are:

- Registry contents
- AUTOFBSS.BAT contents

Depending on the applications and servers you have developed, and the devices you have installed, additional files may need to be copied.

When the shared-file server starts up, it creates the necessary shared-file directories if they do not already exist. This removes the need (which existed in earlier LANDP releases) to call CREADB.BAT to create these directories.

Registry contents

LANDP for Windows NT adds entries to the Windows NT registry so that LANDP servers can be run as Windows NT services. See also “Loading statements for LANDP for Windows NT servers” on page 182.

AUTOFBSS.BAT contents

The AUTOFBSS.BAT file contains the loading statements of the supervisor and the servers that are to reside on the workstation. The first program loaded is the supervisor.

Ensure that any modification of this file does not cause a server to be loaded before the supervisor. Application loading statements should be placed at the end of the server loading statements.

Note: To get trace and log information about the loading process, you have to load the trace tool immediately after the supervisor.

The following is an example of a AUTOFBSS.BAT file for a workstation that contains the SNA server. The name of the workstation is AB.

```
LOADER %1 SPV.EXE /AB
IF ERRORLEVEL 1  GOTO END
LOADER %1 LAN.EXE
IF ERRORLEVEL 1  GOTO END
LOADER %1 SNA##.EXE
IF ERRORLEVEL 1  GOTO END
GOTO OK
:END
EHCFREE SPV /F
EXIT
:OK
```

AUTOFBSS.BAT may be invoked with one positional parameter that will be passed to the loader. Any valid loader parameters may be used. For example:

```
 AUTOFBSS /R:
```

This will remove all servers referenced in AUTOFBSS.BAT from the registry. See page 183 for a description of LOADER parameters.

EHCLIP.EXE should be specified instead of LAN.EXE if the workstation is integrated in a LANDP workgroup that uses TCP/IP as the transport protocol.

You can specify the EXIT command to close the Windows NT session, and leave LANDP for Windows NT running in background. After the :OK label, you can specify a Windows NT command to invoke a LANDP for Windows NT application.

You should invoke the EHCINFO utility program to get an information window about LANDP for Windows NT status. EHCINFO is a LANDP utility program; it is described in *LANDP Servers and System Management*.

Automated unattended startup of LANDP

Under normal circumstances, AUTOFBSS.BAT can only be invoked by a logged on user with the necessary authority. As this may not be convenient in some environments, EHCAUTO.EXE is provided to automate LANDP startup when Windows NT is booted.

To automate the LANDP startup, after the LANDP installation has been verified, issue EHCAUTO from the command prompt in the LANDP runtime directory. You must have administrator authority. This registers EHCAUTO as a Windows NT service that will be started automatically at Windows startup. Subsequent reboots of Windows NT will now run EHCAUTO. In turn, EHCAUTO will invoke AUTOFBSS.BAT. Note that only servers which can run as services are supported and any other programs run from AUTOFBSS must not attempt to interact with the user. By default, EHCAUTO runs under the Local System Account (LSA) and inherits the LSA access authority (refer to Windows NT documentation).

Syntax:

```
EHCAUTO [/D:|/R:|/S:]
```

If no parameters are given, automated start-up is enabled. Administrator authority is required.

```
/D:    - Displays the status of the EHCAUTO service.

/R:    - Disables automated start-up.  Administrator authority
         is required.

/S:    - Stops the EHCAUTO service if it is running.
```

Loading statements for LANDP for Windows NT servers

Most loading statements explained in this section are automatically created by the customization program, using the parameters you provided. The following commands appear in the AUTOFBSS.BAT file provided by customization. Manual modifications can be done if need be. The LOADER command is used to run the servers and the supervisor.

You can use any Windows NT command, such as the START command, to run these programs because they are regular Windows NT programs. Note that using the LOADER command displays the return codes corresponding to the initialization process.

The following syntax is used when servers are to be started as Windows NT services:

```
LOADER [/P:] [/T:nnn] server_name[.exe] [server_options]
```

The first time LOADER is run for any given server, the server will be registered as a Windows NT service. This registration can be done only by users with Administrators authority. When the service has been registered, any user can start the service (see the /P: parameter, below).

Note: You cannot run multiple instances of a server with the same name if the server is running as a Windows NT service.

The LOADER options are:

`/P:` Protected mode. Only users with Administrator or Power User authority can start or stop this server. This option must be specified when the service is first registered for protected mode to be enforced.

nnnn Server initialization timeout (in seconds). The range is 1 to 3600; the default is 30.

The following form of the LOADER statement can be used to manage LANDP servers that are run as services:

```
LOADER {/D:|/R:|/S:} server_name[.exe]
```

The LOADER options are:

`/D:` Displays the state of the service. All users can use this option.

`/R:` Removes the service from the Windows NT registry. The server will be registered again the next time it is started as a service. Only users with Administrator authority can use this option.

`/S:` Stops the service. This option should only be used if EHCFREE fails to stop the server. All users can use this option if the service was first registered without the /P: option. Otherwise, users must have Administrator or Power User authority.

A %1 parameter is generated within the LOADER statements in AUTOFBSS.BAT during customisation. This is to facilitate the addition of a common option to each LOADER request. For example, a `/D:` loader request would display the status of all LANDP servers contained within AUTOFBSS.BAT, by entering the command `AUTOFBSS /D:`.

When servers are to be run as normal applications, the format is:

```
LOADER [/T:nnn] /N: server[.exe] serverparms
```

The LOADER options are:

`/N:` Normal mode (that is, not as a Windows NT service).

nnnn Server initialization timeout (in seconds). The range is 1 to 3600; the default is 30.

Note: Servers stop if a user logs off.

See “Unloading LANDP for Windows NT servers” on page 200 for details of how to stop LANDP for Windows NT servers.

If the initialization works without errors, the server performs initialization, then goes into a wait state until it receives a request from an application. If an initialization error occurs, it is logged in the error log file.

Use IF ERRORLEVEL statements to check the type of errors. If a LANDP error occurs, the corresponding return code is provided.

For return codes during loading, refer to *LANDP Problem Determination*.

Errors detected by LOADER are logged to the Application section of the Windows NT event log. System and Security Event Log messages can also be generated by system functions invoked by LOADER. Server initialization messages may also be logged in EHCLOGxx.DAT.

The following commands are used when you load the LANDP for Windows NT functional areas.

Electronic journal server

```
LOADER ELECJO##.EXE [/K:y]
```

where:

y Is the size, in KBs, of the buffer used to insert the electronic journal records. It ranges from 1 to 4. The default is 1.

The size of the buffer must be large enough to hold the maximum:

- Size of the longest electronic journal record plus 32B
- Split criteria plus 36B plus size of the selected keys. Note that you have to include the *hidden key*, which is 8-bytes long.

Forwarding server

```
LOADER FORWARD.EXE /O:vvvvvvvv.vvv [/T:wwww]
 [/S:xxxxxxxx.xxx]
 [/K:y] [/H:z] [/Z:nnnn]
```

where:

vvvvvvvv.vvv
Is the name of the file corresponding to the ASCII-to-EBCDIC translation table. It must follow the operating system rules.

wwww Is the number of time ticks after which the supervisor will dispatch the forwarding function. One time tick is roughly 0.05 seconds.

It can range from 1 to 6000. The default is 20 (about 1 second).

xxxxxxxx.xxx
Is the name of the file corresponding to the sign-on feature. It must follow the operating system rules.

y Is the size, in KBs, of the buffer used to read the store-for-forwarding records. It ranges from 1 to 4. The default is 1.

The parameter value must be the value assigned in the loading statement of the store-for-forwarding server.

z Specifies whether headers are included when sending host computer messages. The parameter value can be Y, to include headers, or N, not to include headers. The default is Y.

nnnn Is the host code page identifier (DBCS countries only). The permitted values are:

933	Korea
935 or 1388	People's Republic of China (the default value is 935)
937	Taiwan

LAN server

```
LOADER LAN.EXE [/N:n[,n,...]] [/I:x] [/S:y] [/U:s[,s,...]]
```

where:

`n[,n,...]`
Represents the logical adapter numbers used by the LAN.EXE program. These must match the values defined for the NetBIOS protocol driver during NetBIOS configuration. See "Installation requirements for NetBIOS transport protocol" on page 168.

The parameter value can range from 0 to 3. The default is 0.

You can specify a series of up to four adapter numbers. This enables you to configure a server which provides services to workstations on two or more otherwise unconnected LANs. For example:

```
/N:0,1,2,3
```

tells LAN.EXE to use all four adapters.

x Is the time period, in seconds, between attempts to establish the required NetBIOS sessions.

The parameter value can be 0 or in the range 5 through 3000. The default is 20 seconds. If a value of 0 is specified, only one attempt is made to establish each session, at startup, or after a session has been lost.

y Is the NetBIOS send timeout, in seconds.

NetBIOS send, LAN server

The parameter value can be 0 or in the range 10 through 127. The default is 10 seconds. A value of 0 implies no timeout.

`s[,s,...]`
Represents the number of NetBIOS sessions to be made via each adapter. The *s* values correspond to the adapters specified on the /N parameter. For example, the first *s* value after /U corresponds to the first *n* value after /N, and so on.

The parameter values can range from 0 to 254, provided sufficient sessions have been defined during NetBIOS configuration. If the parameter is omitted, or specified as 0, then enough sessions for all the related workstations are used. The total number of sessions specified must be not less than the number of related workstations.

The LAN server is needed only when more than one workstation is present in a LAN. It is included during the process of creating diskettes for distribution.

LANDP Internet Protocol

```
LOADER EHCLIP.EXE [/N:n] [/Y] [/T] [/C[:ws-id]] [/O:o[filename]] [/J]
```

where:

n Specifies the TCP/IP port number used by LANDP Internet Protocol.

The parameter value ranges from 1024 to 65535. The default is 52699.

Y Specifies that no availability probe datagram will be sent, when a session has no normal traffic.

T Requests LANDP Internet Protocol internal routines trace.

C Requests LANDP Internet Protocol communications trace. All sessions are traced, except when the *ws-id* parameter is specified. In this case, only the session with the workstation specified in that parameter is traced.

o Specifies the destination of trace data. The parameter value can be:

1 Standard output. Note that the screen is the usual output, and requires LANDP Internet Protocol be loaded in foreground, with the START command.

2 ASCII file. If you do not specify a name of file, the trace data is stored in the LIPTRACE.TRC file. The target file is initialized each time. Thus, if an existing file is used, the file contents get lost.

3 EHCTRACW (the default).

J Specifies that no checking for related workstations with undefined LANDP Internet Protocol addresses will be carried out.

Magnetic stripe reader/encoder server

```
LOADER MSRE47##.EXE [/C:x] [/D:n] [/O:e]
```

where:

x Identifies the Windows NT communications port to which the MSRE device is attached.

The parameter is optional. The value that can be specified is in the range 1 to 8. The default is 1.

n Identifies whether a 4777 MSRE device may be combined with a 4778 on the same workstation.

The parameter is optional; if used, the value specified must be 1.

e Enables some 4777 device requests to be redirected to a 4778.

The parameter is optional. The value specified can be 0 or 1. The value 0 (the default) disables redirected device requests. The value 1 enables the requests.

MQSeries Link server

`LOADER EHCMQ## [/Q:MQM_name][/L:lllll] [/M:m] [/P:[d:[path]]] [/T:tt] [/S:ss]`

where:

MQM_name
: is the MQSeries Queue Manager Name.

This parameter specifies the name of the queue manager to which the server is to connect. The parameter can be a string that conforms to the rules defined for MQSeries object naming.

Alternatively, the parameter can define an environment variable, for example, '%MQMNAME%', which is resolved when the AUTOFBSS file is processed. The special characters '%' and '/' should not be used in the queue manager name.

If this parameter is omitted or entered as spaces, EHCMQ## connects to the default MQSeries queue manager.

lllll Specifies the maximum message length permitted by the MQSeries Link server. This is used to restrict the maximum message length to something less than the default of 57,000 bytes. The variable *lllll* can take a value between 1000 and 57,000.

m Specifies the message detail level to be written to the log file. The variable *m* can have the values E|W|I where E=errors, W=warnings and I=info. I includes levels W and E, W includes level E. If *m* is omitted, logging is switched off, which is the default.

d Is the drive where the log files are created. If this parameter is omitted, the drive defaults to the drive of the current working directory for the EHCMQ## process.

path Is the path where the log files are created, which defaults to 'EHCMQLOG'. If a path is specified, a drive should also be defined. If both drive and path are omitted, 'EHCMQLOG' is created as a sub- directory of the current working directory for the EHCMQ## process.

A log file is created for each workstation that initiates a session. The log file is named as XXhhmmss.ddd, where

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

The maximum length of the combined /P arguments, d and path, must not exceed 128 bytes .

tt Is the total number of permitted sessions.

This parameter specifies the maximum number of different sessions that the server can process at the same time. The parameter is optional and can be in the range from 1 to 64. The default is 32.

ss Is the number of MQ connections that should be made at startup. Every session requires a MQ connection, which can be made at startup or as session requests are received. The parameter is optional. The variable *ss* can be in the range 0 to *tt*. The default value is 0. When this option is non-zero, the LOADER server's timeout probably needs to be increased. Refer to “Loading statements for LANDP for Windows NT servers” on page 182.

Multiple virtual DOS machine relay

`LOADER EHCVDMGR.EXE [`*path*`\`*name*`]`

where:

path Is the directory where the configuration file created for the MVDM relay is located.

name Is the name of the configuration file.

The configuration file is optional. The default is EHCBOXS.CFG, if it exists.

PIN pad server

`LOADER PINP47##.EXE [/M] [/C:`*x*`]`

where:

/M Indicates that 4778 magnetic stripe reader capabilities are to be used.

The parameter is optional.

x Identifies the Windows NT communications port to which the PIN pad device is attached.

The parameter is optional. The value that can be specified is in the range 1 to 8. The default is 1.

ODBC Query Server

`LOADER EHCODB##.EXE [/C:`*datasource*`] [/T:`*tt*`] [/MT:`*mm*`] [/S:`*ss*`] [/W:`*zz*`] [/EL] [/A]`

where:

datasource
Specifies the name of the ODBC data source that is to be connected. The default is the ODBC data source “CONFIGUR”.

tt Specifies the number of worker threads to create at start up. Each thread allows simultaneous processing of requests, the more threads the greater the throughput. However, the more threads the higher the overhead on system resources. The default is 5. The maximum permitted value is the lowest of 128 and the value of *mm*.

mm Specifies the maximum number of worker threads that EHCODB## can start. This value should be set to enable the ODBC Query Server to increase throughput without using too much system resource. The default is 128. This value must be greater than or equal to *tt*.

ss Specifies the maximum number of sessions a single workstation can handle. The default is 10. This value must be in the range 1 to 64.

zz Specifies the timeout length in seconds for a database request. The default is 15. This value must be in the range 10 to 100000.

`/EL` Specifies that the ODBC Query Server should be enabled to write information messages to the Windows NT event log.

`/A` Specifies that EHCODB## server is aliased to the EHCSQL## server. The EHCODB## server receives requests directed to the EHCSQL## server, as well as direct equests to the EHCODB## server. This is used to aid migration of existing EHCSQL## (query server) applications to EHCODB## (ODBC query server).

Program-to-program communication server

```
LOADER PPC.EXE [/FA] [/FS] [/D:nn]
```

where:

`FA` Specifies MC_FLUSH after ALLOCATE, as a result of an Open Services (OP) function.

`FS` Specifies MC_FLUSH after a Send Data (SD) function.

When the application sends data to the partner application program, the data is stored in buffers that are automatically flushed, and is immediately sent to the partner application. Note that this option may affect the workstations and the network performance.

nn Is the buffer pool that the PPC server initializes to send data to or receive data from the communications provider The default is 16.

Remote change management services

```
LOADER RCMS.EXE /I:xxx /O:yyy [/T:ttt] [/L:n]
[/P:q]
[/C:x] [/R:r]  [/Z:nnnn]
```

where:

xxx Is the file extension of the EBCDIC-to-ASCII translation table (EARCMS.*xxx*). In DBCS mode the parameter does not apply.

yyy Is the file extension of the ASCII-to-EBCDIC translation table (AERCMS.*yyy*). In DBCS mode the parameter does not apply.

ttt Is the number of timer ticks after which RCMS receives control. One timer tick is roughly 0.05 seconds. The default is 10. The value must be a decimal number in the range 1 to 999, For example:

```
/T:1, /T:2, or /T:55
```

n Specifies the number of lines in the EHCRCMS.LOG file. The parameter value can range from 100 to 10000. The default is 1000.

Attention: **If an EHCRCMS.LOG file with** *n1* **lines exists already, and you**

choose a value for the L parameter that is different from *n1*, your old file will be destroyed and a new one created. If you want to keep the old file, rename it or copy it to somewhere else before running the LOADER program.

q Specifies the translation mode. The parameter applies only to DBCS mode. The parameter value can be:

`S` Standard ASCII-EBCDIC and EBCDIC-ASCII translation

`P` 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

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

x Specifies the reception mode for CLISTs. The parameter value can be:

`B` The CLIST is received as a binary file.
`E` The CLIST is received as an EBCDIC file.

The default is `E`.

r Is the interval, in minutes, before retrying a connect to SNA and communications provider after a communications problem. The value must be an integer in the range 0 (which is taken to mean 30 seconds) and 1440 (24 hours). The default value is 0.

nnnn Is the host code page identifier (DBCS countries only). The permitted values are:

933	Korea
935 or 1388	People’s Republic of China (the default value is 935)
937	Taiwan

Searcher

```
LOADER SFQUERY.EXE [/K:y]
```

where *y* is the size, in KBs, of the buffer used to read the electronic journal and store-for-forwarding records. It ranges from 1 to 4. The default is 1.

The parameter value must be the highest of the values assigned in the loading statements of the electronic journal and store-for-forwarding servers.

This loading statement does not correspond to a separate functional area. The SFQUERY.EXE program is required by the electronic journal and the store-for-forwarding servers.

Service Availability Manager

This loading statement does not correspond to a separate functional area. The EHCSAM.EXE program is required by shared-file servers when using the XLR (external logging replicator) facility. For details, see *LANDP Servers and System Management*.

```
LOADER EHCSAM.EXE
```

Shared-file server

(**Note** In this statement, the last parameter uses the letter O, not the digit 0.)

```
LOADER SHFILE##.EXE [/C:confname] [/B:nnn] [/E] [/R] [/S:xxx] [/L:y] [/F:zz]
[/X:ss] [/A:ttt] /O
```

where:

confname
Specifies the name of the profile that defines the shared file. If you omit this parameter, the server uses the name CONFIGUR.

nnn Specifies the number of additional 1 KB index buffers to be allocated; that is, buffers over 15. More index buffers increase system throughput, but also reduce the amount of free storage available for the server workstation. A rule of thumb is that the number of buffers should be 10 per workstation using the shared file server simultaneously. A practical limit is approximately 100, depending on available storage size. The maximum value is 968.

Another factor that must be considered is that the more buffers you have, the greater is the probability of losing index file data when the shared file server is abnormally ended. Thus, if many index buffers are allocated, and the server workstation is switched off with a transaction still in process, or if no RF function has been called, an automatic index rebuild is issued the next time the server is loaded.

`/R` Is an optional parameter to rebuild FREECHAIN. Use it after receiving a `X'A7'` loading error.

`/E` Is an optional parameter to create a file for the statistics gathered during the session.

xxx Specifies the total number of additional sessions in the whole workgroup that the server can manage. The maximum is 243.

This number plus the number of workstations that receive services must not be higher than 243.

y Specifies the log management type. The parameter value can be:

`0`	Dynamic and static log with a unique log file
`1`	Dynamic log with a unique log file
`2`	Dynamic and static log with two log files
`3`	Dynamic log with two log files

The default is 0.

zz Is the number of files open at a time.

The parameter value can range from 10 to 245.

ss Is the suffix of the XLR server pair. For example, for `SHFILE01/BKFILE01`, `/X:01` would be specified.

ttt Is the time delay in seconds before an automatic takeover by the backup is attempted when the active terminates. A value of zero means that no automatic takeover is attempted. This parameter is valid only for XLR servers.

Range: 0-999

/O Overrides normal XLR initialization if only one EHCSAM server is available. This parameter has no default. Use it when only one EHCSAM server is available and then ***only when you are certain that up-to-date databases are available to that server. Do not use this parameter until you have read the following section.*** (**Note** This parameter uses the letter O, not the digit 0.)

Use of /O parameter: During XLR server initialization, LANDP:

- Determines how many EHCSAM servers are available.
- Ensures that state information (which workstation should be the active and which should be the backup) is reliable by polling all the servers.

If a majority vote of at least two SAMs cannot be achieved, LANDP issues the following messages, where *xlr* can be SH or BK.

```
EHC0587: Waiting for XLR state confirmation (Last XLR state on WS N1:
xlrFILE01).
EHC0588: Start other XLR WS, or use /O parameter on SHFILE## to override.
```

If one of the XLR workstations is permanently unavailable and no other SAM is active, the server cannot complete initialization. If you are sure that the state given in the first message is correct, and that the issuing server has up-to-date copies of the databases, you can force confirmation by restarting SHFILE## with the /O parameter. Do ***not*** use this parameter in any other circumstances.

Consider the following scenario:

```
Workstation X1                   Workstation N1

Start as Active                  Start as Backup
Process transactions             Track transactions
Machine failure                  Take over as new Active
Repair.....                      Process transactions
.....machine                     Normal shutdown
Start as  ???
```

In the last step, it is not safe for the XLR server on X1 to initialize. With /O specified, X1 would erroneously assume that it is still the active and use out-of-date data. The only thing to do is to get workstation N1 running as soon as possible. Meanwhile:

X1 waits until workstation N1 starts.
N1 initializes as the active, and X1 as the backup.
X1 catches up on the transactions that took place during its outage.
When catchup is complete, X1 is in a state to take over should N1 fail.

In the above scenario, you ***must*** wait until workstation N1 is initialized. The following is a scenario in which it is safe to use the /O parameter:

```
Workstation X1            Workstation N1

Start as Active           Start as Backup
Process transactions      Track transactions
Machine failure           Take over as new Active
Repair.....               Process transactions
      .....               Normal shutdown
      .....               Start as Active with /O parameter
```

In this case, N1 has up-to-date databases and can safely be started as the active, so the use of the /O parameter is valid. When X1 or another EHCSAM server again becomes available, it automatically becomes the backup, bringing its log and databases up-to-date by communication with N1.

Note: After messages EHC0597 and EHC0588 are issued and before user intervention with the /O parameter, another SAM may start up. If this causes a majority vote, the chosen active XLR server continues initialization.

SNA server

`LOADER SNA##.EXE [/R:`*`r`*`]`

where:

r Is the number of retries if a Connect to an SNA provider for a session returns the LINK_NOT_STARTED_RETRY message.

The value must be an integer in the range 0 (do not retry) to 8. It is advisable to specify a low value, for example `1`; if the connection does not succeed after one retry, there is likely to be a problem in the communications network which needs to be resolved before the connection can succeed.

Store-for-forwarding server

`LOADER SFORFORW.EXE [/K:`*`y`*`]`

where:

y Is the size, in KBs, of the buffer used to insert the store-for-forwarding records. It ranges from 1 to 4. The default is 1.

The size of the buffer must be large enough to hold the maximum:

- Size of the longest store-for-forwarding record plus 32B.
- Split criteria plus 36B plus size of the selected keys. Note that you have to include the *hidden key*, which is 8-bytes long.

Supervisor

`LOADER SPV.EXE /`*`pc-id`*

where:

pc-id Is the identifier of the workstation that was assigned during customization. The parameter value is a string of up to 2 alphanumeric characters, and is case sensitive.

System manager server

```
LOADER SMGR.EXE /D:x /O:yyyyyyyy  [/Z:nnnn]
```

where:

x Is the drive where the FBSS#GDT backup is located. If the parameter is omitted, the backup is not performed.

yyyyyyyy
Is the NetView operator ID. The default is OPER1.

nnnn Is the host code page identifier (DBCS countries only). The permitted values are:

933	Korea
935 or 1388	People's Republic of China (the default value is 935)
937	Taiwan

TCP/IP wide area communications server

```
LOADER EHCTCP.EXE [/M:aaaaaaaa[,aaaaaaaa]] [/T]
```

Where:

aaaaaaaa Is the name of a LANDP communications server to be emulated. This can be SNA## to emulate the SNA server, PPC to emulate the PPC server, or both.

`/T` Is an optional parameter to turn on internal tracing for problem determination.

Trace tools

```
LOADER EHCTRACW  [/R:rrrr] [/B:bbb]
 [/T:xxx]
 [/MT:mmm] [/ML:mmm] [/LT:lll] [/LL:lll]
[/PT:[d:][path]filename] [/PL:[d:][path]filename]
```

where:

rrrr Is the record length in shared memory.

The variable *rrrr* can take values between 64 and 1024 + 64. The default is 394.

bbb Is the maximum number of records in shared memory.

The variable *bbb* can take values greater than 1. The minimum is 2, and the default is 162.

Note that the maximum number of *bbb* is calculated using the formula:

[(64 × (1024 - 64)) ÷ (record length *rrrr*)]

xxx Is the trace option. There are three options for /T (trace facility):

`NO`	No trace is provided
`MEMORY`	Trace records in memory only

`FILE` Trace records in memory and in the file specified by the /PT parameter.

The default is MEMORY. This parameter does not affect the log file, because a log error file is always provided by the server.

mmm Is the maximum number of records for Trace file (/MT) and Log file (/ML). The variable *mmm* can take values between 1 and 50000. If there is not enough space available, an error will be returned. The default in both cases is 512.

You must erase the existing Trace or Log files when you are creating new ones, otherwise the new parameters will not take effect.

Note that you must specify /T:FILE in the loading statement when you create a new Trace file.

lll Is the maximum record length for Trace file (/LT) and Log file (/LL). The variable *lll* must be less than or equal to *rrrr*. The minimum value is 128.

The default for Trace is 394. The default for Log is 150.

d Is the drive where the Trace file and the Log file will be created.

path Is the path where the Trace file (/PT) and the Log file (/PL) will be created. The path must be less than 128 bytes.

filename

Is the name of the Trace file (/PT) and the Log file (/PL). The default name is EHCTRC*xx*.DAT for Trace, and EHCLOG*xx*.DAT for Log. In both cases, *xx* is the workstation identifier.

If EHCTRACW is loaded while the supervisor is still in the loading process, the workstation ID value *xx* in the filenames EHCTRC*xx* and EHCLOG*xx* will sometimes be given the value [!!]. This may happen if you load with the DETACH command. To avoid this, you can either use the LOADER command, or rename the *xx* value for the files afterwards.

Financial printer server

```
LOADER PR47X2##.EXE [/K:n] [/T] [/N] [/R]
```

where:

n Is the maximum number of KB to be printed at a time.

The parameter is optional. The parameter value ranges from 1 to 4. The default is 1.

`/T` Causes the printer event log PR4742.TXT to be generated.

The parameter is optional.

Network attached laser printer support: In LANDP for Windows NT only, the WR (Write to Printer) request allows the DEFAULT Parallel Attached printers to be accessed.

Use two new load-time parameters:

`/N` This tells WR to use the default printer on Windows (local or lan-attached). Without this parameter, the printer server writes only to the local attached printer in PROPRINTER mode.

`/R` This tells the server to leave the data as RAW data and not parse it.. Any printer-control escape sequences in the datastream can pass through without being reformatted by the printer server.

Running with `/N` but without `/R`, the server allows normal PROPRINTER escape sequences to be passed through to the printer. With `/R`, Postscript or PCL data streams pass through without change.

Accessing the DEFAULT Parallel Attached printers: While running as a service LANDP does not have any knowledge of who is or is not logged on. To allow it to find a default printer, the printer server needs an identity. To do this, allow the server to log on as a particular user. Do this ***after*** the servers have been registered (that is, AUTOFBSS has run at least once).

In control panel,

1. Select Services
2. Scroll to LANDP PR47X2## Server, and left double-click.
3. Change the part of the panel that says Log on AS from *system account* to *this account*.

   Select the account name and password that the printer server is supposed to use.
4. To make the changes take effect, stop LANDP and rerun AUTOFBSS.

Selecting as above in the services panel enables the server to find the default printer. There is no default way to do this, even if a user is logged on. This also allows a different user to be logged on when LANDP is running in the background.

If you unregister LANDP: If you run `AUTOFBSS /R:`, you totally unregister LANDP. If you then want to access DEFAULT parallel attached printers, you must:

1. Run AUTOFBSS to re-register PR47X2##.EXE
2. Repeat the process in “Accessing the DEFAULT Parallel Attached printers.”

4748 printer server

```
LOADER PR4748##.EXE [/K:n]
```

where:

n Is the maximum number of KB to be printed at a time.

The parameter is optional. The parameter value ranges from 1 to 4. The default is 1.

Loading statements for emulators in a Windows NT MVDM

If you select MVDM (multiple virtual DOS machine) relay during customization, and define the emulators to be loaded in a Windows NT MVDM, the customization program creates an AUTO_VDM.BAT file with the emulator loading statements. The LSI command is used to run the emulators in the Windows NT MVDM.

When you install more than one emulator, they can run in different Windows NT MVDMs. A separate AUTO_VDM.BAT file is required for each MVDM, containing the loading statements for the emulators that run in that particular MVDM. To create those files, copy only the required loading statements.

FBSS (DOS) and LANDP for DOS applications can also be run in a Windows NT MVDM. If the MVDM relay is not working on your Windows system, check the following.

If you are loading EHCVDSPV from AUTOEXEC.NT, all 16-bit Windows (WOW or Win16-on-Win32) lock EHCVDVXD.DLL, so it cannot be replaced. To overcome this either:

- Stop all WOW sessions, or
- Rename the existing DLL, and copy in the new level of EHCVDVXD.DLL, which becomes active at the next reboot.

3270 emulator

```
LSI EMU3270.EXE /C:atr /K:kbd /D:dis /I:n [/H:hh] [/W:www]
[/B:y]
[/S:xxxxxxxx]
[/Z:nnnn] [/P:a] [/L:l] [/T:tt]
```

where:

atr Is the file extension of the selected display color attributes table.

kbd Is the file extension of the selected keyboard ASCII-to-EBCDIC translation table.

dis Is the file extension of the selected display EBCDIC-to-ASCII translation table.

n Is the emulator identification number.

hh Specifies the alternate screen height (number of rows) of the 3270 display to be emulated. (The height specified should not include the operator information area line at the bottom of the emulator screen.) *nn* must be in the range 24 through 49.

Use this parameter, in conjunction with /W, to make the emulated alternate screen look like one of the following 3270 models:

3270 model	**Alternate screen height**	**Alternate screen width**
2	24	80
3	32	80
4	43	80

If this parameter is omitted, the default is 24.

www Specifies the alternate screen width (number of columns) of the 3270 display to be emulated. *nnn* must be 80.

If this parameter is omitted, the default is 80.

y Indicates whether blinking is supported. Specify Y for yes or N for no.

If this parameter is omitted, the default is N.

xxxxxxxx

Specifies the long name of the 3270 emulator session (sometimes known as the “host session ID”), which is displayed in the operator information area on the screen. You can specify up to eight characters (with no imbedded blanks).

If this parameter is omitted, the default is a name of eight blanks.

nnnn Specifies the size (in bytes) of the buffer used to communicate with the host. *nnnn* can be any value in the range 2048 through 4096. Specifying a small buffer size minimizes memory requirements; using a large buffer can reduce the number of transmissions needed to send or receive a large data stream. The parameter value specified must match the RU size detailed in the bind session.

If this parameter is omitted, the default is 2048.

a Indicates whether the 3270 emulator should handle the Print Screen key. Specify N for no or Y for yes.

If this parameter is omitted, the default is Y.

l Indicates whether the SNA session is connected at emulator load time rather than at ‘hot key’ time. Specify N for NO or Y for YES. The default is N.

tt Specifies the minimum time in seconds between checks on ‘print screen’ key presses. The default is 3.

Note: When working with Windows NT workstations, the 3270 emulator can be used only in a Windows NT MVDM. When working in DBCS mode, the 3270 emulator cannot be used in a Windows NT MVDM, and thus it cannot run on a LANDP for Windows NT workstation.

3287 printer emulator

```
LSI EMU3287.EXE /x /E:prt [/T:nn] [/P:HP] [/N:n]
```

where:

x Can be `S` or `M`. Select `S` for single and `M` for multiple LU_1 support.

Note: This parameter and its values are no longer supported, though they will be accepted for compatibility purposes with earlier versions of LANDP. If specified, they will be ignored.

prt Is the file extension of the selected EBCDIC-to-ASCII translation table.

nn Is the frequency of polling.

The parameter value ranges from 1 to 60. The default is 15.

/P:HP Indicates that the 3287 printer emulator will use either an IBM 4019 Printer, an IBM 4029 Printer, or an IBM 4039 Printer for output. The default is to use an IBM 4201 Proprinter or equivalent device.

/N:n Specifies the range of logical printer numbers that can be used. The parameter value can be in the range 1 through 3. Specify:

- /N:1 to use LPT1 only
- /N:2 to use LPT1 and LPT2
- /N:3 to use LPT1, LPT2, and LPT3

If the parameter is not specified, the emulator uses the number of parallel printer ports physically installed on the workstation.

Note: When working with Windows NT workstations, the 3287 printer emulator can be used only in a Windows NT MVDM. When working in DBCS mode, the 3287 printer emulator cannot be used in a Windows NT MVDM, and thus it cannot run on a LANDP for Windows NT workstation.

Unloading LANDP for Windows NT

The EHCFREE.EXE utility is provided to unload LANDP for Windows NT. You can also unload LANDP for Windows NT by issuing a supervisor function call from an application program. For more information on supervisor function calls, see *LANDP Programming Reference*.

The LANDP utility, EHCFREE.EXE, is called as follows:

```
d:\path\EHCFREE SPV [/p]
```

where:

d: Is the drive where the utility is located.

path Is the path where the utility is located.

p Is an optional parameter that can take the following values:

- For weak unload, *p* = W.

  This is the default value, and will send **ES** to the loaded LANDP components.

  For information on supervisor function calls, refer to the *LANDP Programming Reference*.

- For forced unload, *p* = F.

  This requires that LANDP for Windows NT unloads the LANDP components through the system functions provided to cancel a Windows NT process.

The EHCFREE.EXE program errors are detected by the LAN server or the supervisor. For information on the corresponding return codes, refer to the *LANDP Problem Determination* manual.

Unloading LANDP for Windows NT servers

The EHCFREE.EXE utility program is used to dynamically unload a LANDP for Windows NT server at LANDP run time. The command must be entered in the workstation where the specific LANDP for Windows NT server to be unloaded is located.

Note that the EHCFREE.EXE program can also be used to unload the entire LANDP for Windows NT program.

The EHCFREE.EXE program is called as follows:

```
d:\path\EHCFREE servername [/p]
```

where:

d: Is the drive where the utility is located.

path Is the path where the utility is located.

servername
Is the name of the server to be unloaded, entered in upper or in lower case. Note that if you specify SPV, the entire LANDP for Windows NT program is unloaded.

p Is an optional parameter, which can take the following values.

- For weak unload, *p* = W

  This is the default value, and requires that the supervisor function unload LANDP for Windows NT (**ES**) is issued to the server specified.

  For information on supervisor function calls, refer to the *LANDP Programming Reference*.

- For forced unload, *p* = F

  This requires that LANDP for Windows NT ends the servers specified, through the system functions provided to cancel a Windows NT process.

The EHCFREE.EXE program errors are detected by the LAN server or the supervisor. For information on the corresponding return codes, refer to the *LANDP Problem Determination* manual.

Installing run-time files

The LANDP family provides a utility program to check the path where the run-time files are located. See “Installing and validating system files” on page 236 for more information about this program.

For other utility programs also provided to be used at run-time, refer to Chapter 15, “Run-time utility programs” on page 229.

The LANDP customization process creates the EHCMSG.DLL file, which must be located in the current path or in a path specified using a PATH statement in the registry table.

Chapter 14. Preparing RS/6000 workstations

Following are the prerequisites you need to take into account before starting with the RS/6000 integration tasks.

The tasks involve two RS/6000s that are conceptually different. A *development RS/6000*, which may be installed in the development site, is used for completing customization and a *production RS/6000* participates in a LANDP workgroup.

It is possible to use the RS/6000 that is integrated in a LANDP workgroup also for the preparations. The following information considers both possibilities.

Prerequisites

Before the preparations for the RS/6000 can be started, make sure that the required hardware and software components are available. In addition, you must have completed the LANDP customization on your customization workstation.

LANDP for AIX is installed on the development RS/6000 into the *install library*.

Requirements

There are certain hardware and software requirements that have to be fulfilled before preparation. These are described below.

Tape drive

LANDP for AIX is delivered on a range of media, the default being an 8 mm data cartridge. You must ensure that you have the correct medium to enable you to install it on your system. You should also make sure you can exchange media between the development and the production RS/6000, if you have one.

RS/6000 software

For the preparation and installation tasks the following software has to be installed:

- Operating system IBM AIX Version 3.2 (Program Number 5756-030), running on the development and production RS/6000.
- If NetBIOS is being used, rather than TCP/IP, NetBIOS for AIX (program 5765-382).
- IBM AIX SNA Services/6000 or IBM SNA Server/6000, if you use the SNA or PPC server.
- DB2/6000 Version 1.1, Informix relational database manager Version 5.0 or higher, or Oracle7, if you use the query server, the store-for-forwarding server, or the electronic journal server.
- IBM AIX DCE Version 1.03 for LANDP-DCE interoperability.

- IBM AIX Xstation Manager®/6000 (program 5601-457) is required if Xstations are to be used.
- The AIX financial devices device driver, as shipped with the 4777 or 4778 devices, is required if either of these devices are to be used.
- For a financial printer server running a 4009 device, microcode level 5.0 or higher is required and the `honor CTC` option on the Operator Panel must be used.
- Motif Version 1.2 is required if you are going to use the trace display facility (dczytrdsp), the X-Windows version of the System Verification Program (dczysvp), or application programs that use Supervisor Local Functions (SP, TP, or QE).

Disk storage

To install LANDP for AIX you need about 15 MB disk storage.

Preparation tasks

Figure 7 on page 205 illustrates the flow of the preparation tasks you have to perform to enable the RS/6000 to participate in a LANDP workgroup.

Four different tasks are required:

1. Installing LANDP for AIX on the development or on the production RS/6000.
2. Transferring customization data to the development RS/6000.
3. Completing LANDP for AIX customization on the development RS/6000.
4. Transferring LANDP for AIX configuration files to the production RS/6000.

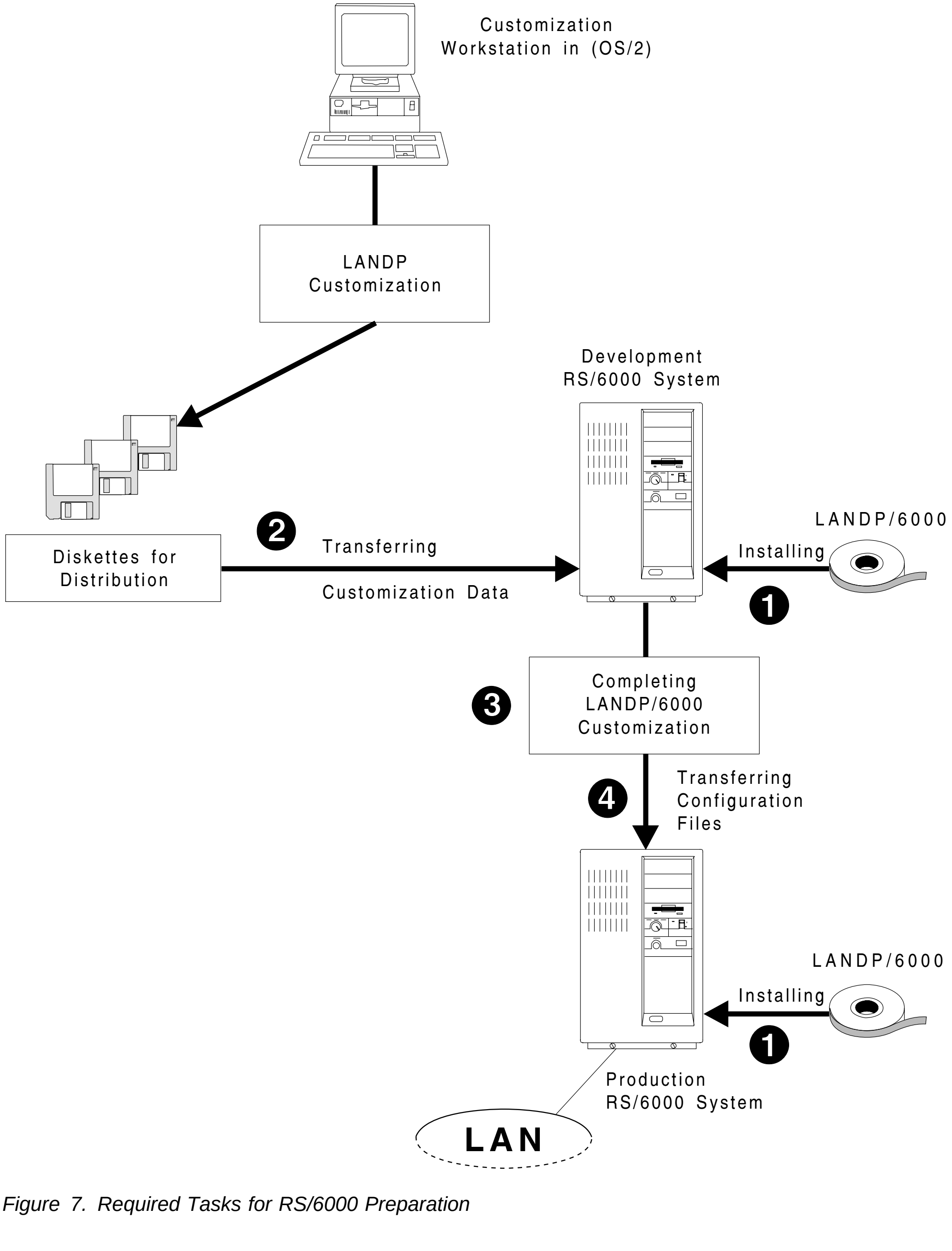

Customization Workstation in (OS/2)
LANDP Customization
Development RS/6000 System
Diskettes for Distribution
2
Transferring
Customization Data
LANDP/6000
Installing
1
3
Completing LANDP/6000 Customization
4
Transferring Configuration Files
LANDP/6000
Installing
1
Production RS/6000 System
LAN

Figure 7. Required Tasks for RS/6000 Preparation

Installing LANDP for AIX

See also **Step** ❶ in Figure 7 on page 205.

To install LANDP for AIX on the development RS/6000, or on the production RS/6000, log in as *root*.

You can now do one of the following:

- Use the AIX System Management Interface Tool (SMIT), which provides an easy-to-use interface to AIX. You can use SMIT to simplify many of the AIX operations required by LANDP for AIX. For detailed information see *IBM AIX System Management Guide: Operating System and Devices*, GC23-2486.

  The fastpath for invoking SMIT is to enter the following command:

  ```
  smit landp
  ```

- Enter the AIX

  ```
  installp
  ```

  command with the appropriate parameters. For information, refer to *IBM AIX Command Reference, Volume 2,* GC23-2366.

Communication profiles

LANDP for AIX provides SNA server and PPC server communication profiles. They should be regarded as samples.

Information on how to use the profiles and the sample applications is provided in the description file: `landpsamples.doc`, which is located in the `/usr/lpp/landp` directory. For more information about communications refer to the *IBM RS/6000 Communication Concepts and Procedures* manual.

Note that the interface type must be `extended`, and `parallel` must be specified to run parallel session concurrency.

Transferring customization data

See also **Step** ❷ in Figure 7 on page 205.

After LANDP customization has been successfully accomplished, the customization data created for LANDP for AIX must be transferred to the development RS/6000.

Copy the operable files to a *diskette for distribution*. For information, refer to Chapter 9, “Distributing software” on page 77.

To install the diskette on the RS/6000, select the **Install customized LANDP workstation diskette** option from the Configure LANDP for AIX menu. You can access the menu by selecting **Communications Applications and Services** on the

SMIT menu and **IBM AIX LAN Distributed Platform/6000 (LANDP for AIX)** on the subsequent one.

You can also copy the operable files to a directory on a hard disk in the customization workstation. For information, refer to Chapter 9, “Distributing software” on page 77.

To transfer those files to the RS/6000, you can use TCP/IP File Transfer Protocol (FTP). They have to be transferred to the directory

```
/var/landp/xx[.ssssssss]
```

where:

xx Is the workstation ID,

ssssssss Is the workgroup name.

Ensure that files in ASCII format are transferred in ASCII mode, and files in binary format are transferred in binary.

When the customization data is on the development RS/6000, you must run

```
dczyprlandp xx.ssssssss
```

Then, you can go on with completing LANDP for AIX customization.

Completing LANDP for AIX customization

See also **Step ❸** in Figure 7 on page 205.

After the LANDP customization data is transferred to the development RS/6000, you have to provide additional definitions that are closely related to the AIX environment.

Select the **Complete LANDP customization** option from the Configure LANDP for AIX menu. You can access the menu by selecting **Communications Applications and Services** on the SMIT menu and **IBM AIX LAN Distributed Platform/6000 (LANDP for AIX)** on the subsequent one.

On the Complete LANDP customization menu, you can select the action to be performed on the server or application parameters. After selection, a list is displayed that shows the server names as well as the Application Parameters item. When you have selected an item in the list, and entered the workstation ID in the subsequent panel, you can perform the action.

This process generates additional configuration files. The .cfg files generated during LANDP customization remain unchanged.

After completing LANDP for AIX customization, you can distribute the files of the configuration directory to the production RS/6000s.

The target directory must have the following AIX permissions and ownerships:

User ID	Any
Group	`system`
Access rights	`drwxrwxr-x`

Note: You can only use the **Add Client** option in the Complete LANDP/6000 customization menu if a client has been defined (during customization on LANDP for OS/2) and has not yet been configured using this option. If you use the option **Change Client Configuration**, and none of the parameters have changed, an error is reported.

Application parameters

You can define a number of user IDs for the LANDP for AIX workstation. Those user IDs can simultaneously run applications that use a unique module.

To make the application module use different instances of the same service, you have to assign a different service name to the module for each user ID that can start the application.

You can attach a number of terminals to the LANDP for AIX workstation. Each user ID can access the LANDP for AIX workstation from any terminal.

To make the application module use the same service from the same terminal, regardless of the user ID that runs the application program, you have to assign the service name to the module for that terminal address.

If an application module is assigned a service name for a user ID, and another service name for a terminal address, when the user ID starts the application from that terminal, the module uses the service assigned for the user ID.

Security in a mixed-system workgroup

If you plan to use LEVELF security in a workgroup that includes a LANDP for AIX workstation, you need to add extra SMGRUSER entries to COMMON.SPC, for the following reason.

If you have defined your workgroup to use LEVELF security, then a user must be signed on with LEVELF authority in order to use any application from a LANDP workstation.

Any request sent within a workgroup is checked by the supervisors running in the client and server workstations to confirm that it comes from an authorized user. When the workgroup includes a LANDP for AIX workstation, this check needs to ensure that the LANDP for AIX user is authorized. In order for the security system provided with LANDP to co-operate with the security system provided by AIX, you need to supply LANDP System Manager User (SMGRUSER) vectors for user IDs that may logon to the RS/6000 system.

You must provide in COMMON.SPC a SMGRUSER entry with LEVELF=Y for:

- A user ID of `root`
- Each user ID that you plan to use on a LANDP for AIX workstation

For details of syntax, see “SMGRUSER vector” on page 396.

Completing financial printer server customization

When you have selected a financial printer server session, you can change the session characteristics defined during LANDP customization:

- Port identification
- Port usage
- Device name

Given a session, the client workstation that uses it cannot be changed.

Notes:

1. In order to connect a 4712 or 4722 printer to a RS/6000 system, you must ensure that Switch 7 of the Printer RS232 Interface Card is **ON**.
2. When using a 4772 printer, Switch 7 of the Baud Rate Switch Group must be set to **ON**.

Completing MSR/E server customization

The MSR/E server requires that at least one Finance Device has previously been defined. For details of how to define a Finance Device, see *4777 Magnetic Stripe Unit and 4778 PIN-Pad Magnetic Stripe Reader AIX Programming Guide*, SA34-2358.

When you have selected a MSR/E server session, select the Finance Device to be used.

Completing PIN pad server customization

The PIN pad server requires that at least one Finance Device has previously been defined. For details of how to define a Finance Device, see *4777 Magnetic Stripe Unit and 4778 PIN-Pad Magnetic Stripe Reader AIX Programming Guide*, SA34-2358.

When you have selected a PIN pad server session, select the Finance Device to be used.

Completing SNA server customization

A LANDP for AIX workstation connected to a workgroup, and customized to provide SNA services to its local users and to remote workstations, requires additional, specific LANDP for AIX customization definitions.

The following information describes how to provide the additional customization definitions by using an example configuration as shown in Figure 8 on page 210.

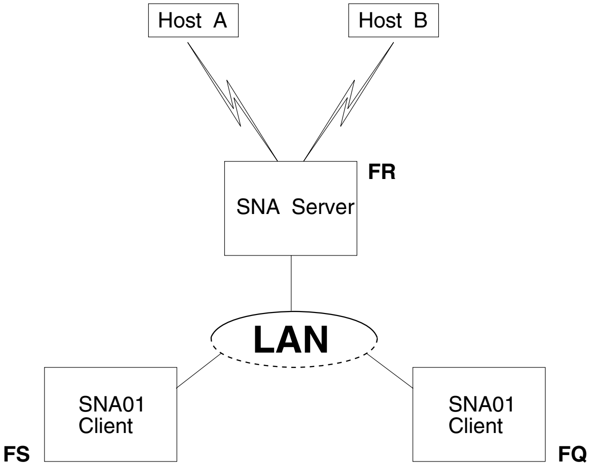


Figure 8. Sample Configuration with LANDP for AIX Communications Gateway

The LANDP for AIX workstation FR runs the LANDP SNA server and enables communication with the host computers, Host A and Host B. With the LANDP customization, local users of workstation FR and the workstations FS and FQ were defined as clients that use the SNA services.

With the additional, specific LANDP for AIX customization definitions, you define the SNA environment on the LANDP for AIX workstation required to meet the communication needs of the clients. The workstations FS and FQ use the SNA01 service from FR. You need to define which logical unit (LU) and which attachment profile name, that is which physical unit (PU), are to be assigned to these clients.

Customizing startup parameters

Some LANDP for AIX servers require specific definitions for startup.

Select the **Specify startup parameters** option from the Configure LANDP for AIX menu. You can access the menu by selecting **Communications Applications and Services** on the SMIT menu and **IBM AIX LAN Distributed Platform/6000 (LANDP for AIX)** on the subsequent one.

On the Specify Startup Parameters menu, you can select the action to be performed on the server. After selection, a list is displayed that shows the server names. When you have selected an item in the list, and entered the workstation ID in the subsequent panel, you can perform the action.

The parameters can also be defined through the *command line*. The `dczyloader` program is provided to run the supervisor and all the servers.

Before you finish debugging a user server, it is recommended not to load the server with `dczyloader`, because this program:

- Detaches from the current process and the controlling terminal
- Closes all file descriptors, and does not allow for screen traces
- Uses the root directory as current

Electronic journal server

To start the electronic journal server, enter:

```
  /usr/sbin/dczyloader /usr/sbin/dczyelecjo -slaves <sl> -sessions <ss>
-dbname <db>
```

where:

sl Specifies the number of slave processes to be started. The parameter value ranges from 1 to 100. The default is 5.

ss Specifies the number of pre-started sessions per slave process. The parameter value ranges from 0 to 255. The default is 1.

db Specifies the name of the database to be used for electronic journal data. The parameter value is a string of 10 ASCII characters.

The electronic journal server logs data set states to the `logical_table.dbname` and the `physical_table.dbname` files. These files are created in the path specified by `/etc/landp`.

For performance reasons, this path should not be a Network File System (NFS) path.

To reset the data set states, ensure that the server is not running, and erase those two files.

Financial printer server

To start the financial printer server, enter:

```
  /usr/sbin/dczyloader /usr/sbin/pr47x2## [-td] [-ofilename] [-cxxxy]
```

where:

t Specifies printer server traces activation.

You can also specify device driver traces activation, with the parameter *d*.

If you omit the parameter, the traces are not activated.

o Specifies that the traces will be stored in a file.

You can also specify the name of the file, through the *filename* parameter value. The default is `pr47x2##.trc`. The file is initialized each time.

If you omit the parameter, the traces are sent to the screen.

xxx Specifies the codepage to be used.

The default is assumed according to the environment settings.

y Specifies the font ID to be used.

The parameter value ranges from 0 to 8:

0	Default printer font
1	Default font, standard text print
2	Default font, letter quality print
3	DP Gothic font, standard text print
4	DP Gothic font, letter quality print
5	Prestige Elite 12 font, standard text print
6	Prestige Elite 12 font, letter quality print
7	Courier 10 font, standard text print
8	Courier 10 font, letter quality print

Values from 3 through 8 apply only to the 4772 printer. The default is assumed according to the environment settings.

Forwarding server

To start the forwarding server, enter:

```
  /usr/sbin/dczyloader /usr/sbin/dczyforward -o <translation table>
[-t <delay time>] -s <sign-on feature> -h <header transmission>
```

where:

translation table
Is the name of the file corresponding to the ASCII-to-EBCDIC translation table. It must follow the operating system rules.

delay time Is the amount of time (in multiples of 50 ms) during which the forwarding server periodically sleeps.

sign-on feature
Is the name of the file corresponding to the sign-on feature messages. It must follow the operating system rules.

header transmission
Specifies whether headers are transmitted when sending messages to the host computer. The parameter value can be `Y`, to transmit headers, or `N`, not to transmit headers. The default is `Y`.

If you plan to use the forwarding server with cryptography support, specify `dczyforwardc` instead of `dczyforward`. You can specify the same parameters, except for *sign-on feature*, which is not supported by `dczyforwardc`.

To use `dczyforwardc`, Transaction Security System/6000 must be installed.

Note that data delivered to the host transaction program is padded with one to seven EBCDIC blanks.

LAN server

To start the LAN server, enter:

```
/usr/sbin/dczyloader /usr/sbin/dczylan [-ax] ws_id
```

where:

x Specifies the identification of the local area network adapter to be used. The parameter is optional. The parameter value ranges from 0 to 3.

ws_id Is the identifier of the workstation that was assigned during customization. The parameter value is a string of up to 2 alphanumeric characters, and is case sensitive. The parameter is required.

LANDP-DCE interoperability

To enable the LANDP-DCE interoperability, enter:

```
/usr/sbin/dczyloader /usr/sbin/dczymap -cnnn -pstring1 [-pstring2]
```

where:

nnn Specifies the maximum number of concurrent DCE clients.

This means the maximum number of DCE clients that can simultaneously access LANDP servers. DCE clients can access LANDP servers after getting a *context*, and until they release it.

The parameter is required. The parameter value ranges from 0 to 999.

string1 Specifies a protocol sequence used by DCE clients to make requests to the LANDP servers.

The parameter is required. The parameter value can be either `ncacn_ip_udp` or `ncadg_ip_tcp`.

string2 Specifies a protocol sequence used by DCE clients to make requests to the LANDP servers.

The parameter is optional. The parameter value can be either `ncacn_ip_udp` or `ncadg_ip_tcp`. If it is omitted, only the protocol sequence specified as *string1* is used.

To enable LANDP workstations access to DCE servers, and DCE clients access to LANDP workstations, through a LANDP for AIX workstation, you must enable the LANDP-DCE interoperability in this workstation.

LANDP internet protocol

To start the LANDP Internet protocol, enter:

```
/usr/sbin/dczyloader /usr/sbin/dczylip [-Nn] [-Y] [-T][-Cws_id] [-Ofilename][-J]
```

where:

n Specifies the TCP/IP port number used by LANDP Internet protocol.

The parameter value ranges from 1024 to 65535. The default is 52669.

Y Specifies that no availability probe datagram will be send, when a session has no normal traffic.

T Requests LANDP Internet protocol internal routines trace.

C Requests LANDP Internet protocol communications trace. All sessions are traced, except when the *ws_id* parameter is specified. In this case, only the session with the workstation specified in that parameter is traced.

O Specifies the destination of trace data.

If the parameter is omitted, trace data is redirected to the standard output. If the parameter is specified, but no *filename* is provided, trace data is redirected to the DCZYLIP.TRC file. If both the parameter and the name of a file is specified, trace data is redirected to that file.

J Specifies that no checking for related workstations with undefined LANDP Internet Protocol addresses will be carried out.

Magnetic stripe reader/encoder server

To start the magnetic stripe reader/encoder server, enter:

```
/usr/sbin/dczyloader /usr/sbin/dczymsre [-t/ty] [-x/xz]
```

where:

t*y* Specifies that the traces will be sent to the screen.

The parameter value for *y* can be:

p Only parent traces
c Only child traces

x*z* Specifies that the traces will be stored in the msre47##.trc file.

The parameter value for *z* can be:

p Only parent traces
c Only child traces

PIN pad server

To start the PIN pad server, enter:

```
  /usr/sbin/dczyloader /usr/sbin/dczypinp [-t/ty] [-x/xz]
```

where:

t*y* Specifies that the traces will be sent to the screen.

The parameter value for *y* can be:

p Only parent traces
c Only child traces

x*z* Specifies that the traces will be stored in the pinp47##.trc file.

The parameter value for *z* can be:

p Only parent traces
c Only child traces

PPC server

To start the PPC server, enter:

```
  /usr/sbin/dczyloader /usr/sbin/dczyppc -a -s
```

where:

a Specifies MC_FLUSH after ALLOCATE, as a result of an open services (OP) function.

s Specifies MC_FLUSH after a send data (SD) function.

When the application sends data to the partner application program, the data is stored in buffers that are automatically flushed, and is immediately sent to the partner application.

Query server

To start the query server, enter:

```
  /usr/sbin/dczyloader /usr/sbin/servername -d <database> [-p <sessions>]
[-t <timeout>] [-u]
```

where:

servername Specifies the LANDP for AIX query server program, depending on the database to be used:

dczydbinf Informix Standard Engine or Informix Online
dczydbora Oracle7
dczydbdb2 DB2/6000

database Specifies the database (Informix or DB2/6000), or a connect string or SID (Oracle7).

sessions Specifies the number of pool sessions for the query server. The parameter value ranges from 1 to 100. The default is 1.

Pool sessions are working sessions opened during startup. When the query server receives an open query (OQ) or an open session (OS) function call, it checks for pool session availability.

If no pool session is available, the query server establishes a working session. If pool sessions are available, process is faster.

timeout Specifies the amount of time (in seconds) the query server waits for a resource response. The default is 0.

- For Informix Online and Informix Standard Engine databases, if the parameter value is 0, the query server does not wait for a resource to be unlocked but immediately returns a non-zero return code.
- For Oracle7 and DB2/6000 databases, if the parameter value is 0, the query server waits indefinitely for a resource to be unlocked.
- For Oracle7 (query mode only) databases, each statement selected for update has a WAIT/NOWAIT clause. The default is WAIT.

u Enables user separation.

If the parameter is specified, an application can change the user ID of a query server working session using the register user (RU) or the open session (OS) function.

Notes:

1. If you are using Informix Online, it is recommended that you run it in raw disk space, rather than as part of a file system. This should enhance performance.
2. If you are running on a DB2/6000 database, it is recommended that you remove the following statement from your /etc/landp/autolandp file:

   ```
   export MALLOCTYPE=3.1
   ```

   If you include this statement, you may receive error code RS or PE in certain circumstances when using DB2/6000.

Shared-file server

To start the shared-file server, enter:

```
  /usr/sbin/dczyloader /usr/sbin/dczyloader /usr/sbin/shfile [-C:confname] [-B:nnn]
[-E] [-S:xxx] [-L:y] [-F:zz]
```

where:

confname
Specifies the name of the profile that defines the shared file. If you omit this parameter, the server uses the name CONFIGUR.

nnn Specifies the number of additional 1 KB index buffers to be allocated; that is, buffers over 15. More index buffers increase system throughput, but also reduce the amount of free storage available for the server workstation. A rule of thumb is that the number of buffers should be 10 per workstation using the shared file server simultaneously. A practical limit is approximately 100, depending on available storage size. The maximum value is 968.

Another factor that must be considered is that the more buffers you have, the greater is the probability of losing index file data when the shared file server is abnormally ended. Thus, if many index buffers are allocated, and the server workstation is switched off with a transaction still in process, or if no RF function has been called, an automatic index rebuild is issued the next time the server is loaded.

`-E` Is an optional parameter to create a file for the statistics gathered during the session.

xxx Specifies the total number of additional sessions in the whole workgroup that the server can manage. The maximum is 245.

This number plus the number of workstations that receive services must not be higher than 245.

y Specifies the log management type. The parameter value can be:

0	Dynamic and static log with a unique log file
1	Dynamic log with a unique log file
2	Dynamic and static log with two log files
3	Dynamic log with two log files

The default is 0.

zz Is the number of files open at a time.

The parameter value ranges from 10 to 245.

When it is loaded, the shared file server produces messages which can be stored in the LANDP.LOG file. If the server fails to start, check this file for loading return codes. Information messages are also generated when the index and data files are created.

For more information on loading return codes, see *LANDP Problem Determination*. For more information on the LANDP.LOG file, see “Using the LANDP.LOG file” on page 226.

SNA server

To start the SNA server, enter:

```
/usr/sbin/dczyloader /usr/sbin/dczysna -x -s sscpid
```

where:

x Establishes X.25 connections (SVC and PVC) at SNA server startup. The process can take up to two minutes.

If the parameter is not specified, the X.25 connection is opened with the first connect (CN) function call.

s Makes the SNA server send alerts to the specified SSCP ID.

sscpid
Specifies the SSCP ID that receives the alerts.

Store-for-forwarding server

To start the store-for-forwarding server, enter:

```
  /usr/sbin/dczyloader /usr/sbin/dczysforforw -slaves <sl> -sessions <ss>
-dbname <db>
```

where:

sl Specifies the number of slave processes to be started. The parameter value ranges from 1 to 100. The default is 5.

ss Specifies the number of pre-started sessions per slave process. The parameter value ranges from 0 to 255. The default is 1.

db Specifies the name of the database to be used for store-for-forwarding data. The parameter value is a string of ten ASCII characters.

Transferring LANDP for AIX configuration files

See also **Step ❹** in Figure 7 on page 205.

If you use the same RS/6000 as development and production RS/6000, you do not need to transfer the configuration files.

To transfer the configuration files to the production RS/6000, you can use your own tools. For example, you can use TCP/IP File Transfer Protocol (FTP).

You can also copy the configuration files to a diskette, and install the diskette on the production RS/6000.

The configuration files have to be transferred to a directory with the same name as that where the files are located in the development RS/6000:

```
  /var/landp/xx[.ssssssss]
```

where:

xx Is the workstation ID,

ssssssss Is the workgroup name.

The files have to be transferred in binary, without character translation.

The target directory and files must have the same access rights as the source directory and files.

After completing LANDP for AIX customization, three additional .cfg files are created that must be transferred to the production RS/6000. The .cfg files created during LANDP customization remain unchanged.

Configuring NetBIOS for AIX

To configure and start NetBIOS for AIX, use the `mcsadm` command. This takes you into the NetBIOS for AIX Administration Utility.

1. Configure the protocols.
   a. From the main menu, select **Configure Protocols**.
   b. Select **LLC 802.2 Compatible Protocol**.
   c. Select **Configure General Parameters**.
   d. Select **Add Configuration**.
   e. Save using the default values, unless you have a particular need to change them.
   f. Return to the Configure Protocols panel, and select **Select NetBIOS**.
   g. Select **Configure NetBIOS LANAs**.
   h. Select **Add LANA Configuration**.
   i. Select **NetBEUI**.
   j. Save using the default values, unless you have a particular need to change them.
2. Start NetBIOS.

   Return to the main menu, and select **Start NetBIOS from AIX**.

For more information, refer to *NetBIOS for AIX System Administrator's Guide*, SC23-2761.

Checking for TCP/IP transport protocol requirements

To use TCP/IP as the internal communications protocol for a LANDP workgroup, TCP/IP must be installed and configured in all the workstations.

For detailed information, refer to the TCP/IP manuals. See "Bibliography" on page 571.

For more information about the TCP/IP transport protocol, see see Appendix E, "Using TCP/IP for internal communication" on page 525.

Defining LANDP for AIX databases

LANDP for AIX supports the following database products:

- Informix Standard Engine (Informix SE)
- Informix Online
- Oracle7
- DB2/6000

You should install the database using standard RDBMS procedures. This section contains examples to help you with your initial setup.

Informix SE and Informix Online

The following example shows some of the environment variables required for the Informix SE and Informix Online databases. These variables are dependent on the way the product was installed.

```
INFORMIXDIR=/usr/informix
PATH=$PATH:INFORMIXDIR/bin
DBEDIT=vi
DBDATE=DMY4
DBPATH=$HOME
SQLEXEC=$INFORMIXDIR/lib/sqlprog
```

where:

sqlprog Is `sqlexec` for Informix SE, and `sqlturbo` for Informix Online.

To start Informix Online, use the `tbinit` command.

To create a database:

1. Switch to the DBPATH directory.
2. Run the `dbaccess` command.
3. Select **Query Language** and enter one of the following commands:
   - For Informix SE:

     ```
     CREATE DATABASE landp WITH LOG IN "/logname" MODE ANSI
     ```

   - For Informix Online:

     ```
     CREATE DATABASE landp WITH LOG MODE ANSI
     ```

   where:

   landp Is the name of the database you wish to create.

   logname Is the log file name, for example `/home/landp/informix_se.log`.

To grant access to a user, enter the following command:

```
GRANT DBA TO "username"
```

where:

username
: Is the name of the user to whom you are granting access. You *must* grant access to the following users:

`root`
: This user must be granted access as the database administrator (DBA). The first time you load the query server, it is loaded from the root user. You must specify `root` in lower case.

  You should check the AIX system log for messages after loading the query server with a new database.

`landp`
: This user must be granted access to the database. The query server accesses the database through the landp user. You must specify `landp` in lower case.

To stop Informix Online, use the `tbmode -ky` command.

For Informix databases, only system users can access a database. No password is required.

Oracle7

The following example shows some of the environment variables required for the Oracle7 database. These variables are dependent on the way the product was installed.

```
ORACLE_BASE=/usr/oracle
ORACLE_HOME=/$ORACLE_BASE/product/7.0.1.6
ORACLE_SID=landp
ORACLE_TERM=hft
```

After installation, to start Oracle7 enter `dbstart`.

Run the following commands to create a database for the Oracle7 user:

```
sqldba lmode=yes
sqldba> connect internal
sqldba> startup nomount
sqldba> create database landp;
sqldba> create user OPS$LANDP IDENTIFIED EXTERNALLY;
sqldba> create user OPS$ROOT IDENTIFIED EXTERNALLY;
sqldba> grant dba to OPS$LANDP;
sqldba> grant dba to OPS$ROOT;
sqldba> exit
```

where:

landp
: Is the name of the database you wish to create.

You *must* create OPS$ROOT and OPS$LANDP.

OPS$ROOT is the database administrator, The first time you load the query server, it is loaded from OPS$ROOT. You should check the AIX system log for messages after loading the query server with a new database.

The query server then accesses the database through the OPS$LANDP user.

To stop Oracle7, enter the `dbshut` command.

For Oracle7 databases, only system users and database users can access a database.

Note: All user IDs used in Oracle7 must be prefixed by OPS$. This is the Oracle OS_AUTHENT_PREFIX, and is described in full in the *Oracle Administrator's Guide*.

DB2/6000

The following example shows some of the environment variables required for the DB2/6000 database:

```
DB2COMM=NONE
DB2INSTANCE=instname
```

where:

instname
Is the DB2/6000 instance.

After installing and creating a DB2/6000 instance, for example `landp`, to start DB2/6000 use the `db2start` command.

To create a database, enter the following commands:

```
db2 create database dbname AUTHENTICATION CLIENT
db2 grant dbadmin on database to root
db2 grant dbadmin on database to landp
```

where:

dbname Is the name of the database.

`root` Is the database administrator (DBA). The first time you load the query server, it is loaded from the root user. You must specify `root` in lower case.

You should check the AIX system log for messages after loading the query server with a new database.

`landp` Is the user through which the query server accesses the database. You must specify `landp` in lower case.

To stop DB2/6000, enter the `db2stop` command.

For DB2/6000 databases, only system users can access a database. No password is required.

Distributed computing environment considerations

This section applies if you plan to access the Distributed Computing Environment (DCE) servers, or to allow DCE clients access to the LANDP servers.

To enable LANDP-DCE interoperability, you have to provide some definitions related to:

- DCE cell directory server
- DCE security

This task can be performed either in the production RS/6000 or in any RS/6000 that belongs to the DCE cell.

For DB2/6000 databases, only system users can access a database. No password is required.

Cell directory server definitions

DCE objects store binding information in a database to make it available to other DCE objects. The database is maintained by the cell directory server (CDS).

To enable LANDP-DCE interoperability, you have to create the directory that will contain the corresponding LANDP entries. To create the directory, you can use SMIT.

Perform the following steps:

1. To login to DCE as cell_admin, enter:

   ```
   dce_login cell_admin cell_admin_password
   ```

2. To start SMIT for DCE tasks, enter:

   ```
   smit dce
   ```

3. Select the **CDS Administration** option.
4. Select the **Directories and Replicas** option.
5. Select the **Add a New Directory** option.
6. To enter the LANDP directory entry, specify `/.:/subsys/LANDP` as the `DIRECTORY name`, and the chosen `CLEARINGHOUSE`.
7. Exit SMIT.

Security definitions

If a DCE cell contains a security server, a DCE security entity must be created to enable LANDP-DCE interoperability.

This means you have to create a *principal* and an *account*. To create them, you can use SMIT.

Perform the following steps:

1. To login to DCE as cell_admin, enter:

```
dce_login cell_admin cell_admin_password
```

2. To start SMIT for DCE tasks, enter:

   ```
   smit dce
   ```

3. Select the **DCE Security & Users Administration** option.
4. Select the **Principals and Aliases** option.
5. Select the **Add a Principal** option.
6. To create the principal, specify `LANDPDCE` as the Principal Name. You can leave the other fields empty.
7. Exit to the DCE Security & Users Administration panel.
8. Select the **Accounts** option.
9. Select the **Add an Account** option.
10. To create the account, specify the following:
    - For type of ACCOUNT, enter `rpc server`.
    - For the PRINCIPAL to create account for, enter `LANDPDCE`.
    - For the ORGANIZATION to associate with this account, enter `none`.
    - The `PASSWORD` for the new account
    - When you are prompted to Enter your PASSWORD, type `cell_admin_password`.
11. Exit to the DCE Security & Users Administration panel.
12. Select the **GROUPS** option.
13. Select the **Add Members to a Group** option.
14. To define the LANDPDCE principal as a member of the subsys/dce/cds-admin group, specify the following:
    - `subsys/dce/cds-admin` as the GROUP name.
    - `LANDPDCE` as the PRINCIPALS to add.
15. Exit SMIT.

Starting and stopping LANDP for AIX

Starting and stopping LANDP for AIX is most easily done using the AIX System Management Interface Tool (SMIT).

Starting LANDP for AIX

1. Access the Control LANDP for AIX panel in SMIT.

   You can access this panel by selecting **Communications Applications and Services** on SMIT main menu and **IBM AIX LAN Distributed Platform/6000 (LANDP for AIX)** on the subsequent one.

2. Select **Start LANDP for AIX** from the Control LANDP for AIX panel.

System user group rights are required in order to start LANDP for AIX.

Starting LANDP for AIX servers

1. Access the Control LANDP for AIX panel in SMIT.

   You can access this panel by selecting **Communications Applications and Services** on SMIT main menu and **IBM AIX LAN Distributed Platform/6000 (LANDP for AIX)** on the subsequent one.

2. Select **Start specific LANDP for AIX server** from the Control LANDP for AIX panel.

You can also load LANDP for AIX servers (except for the shared-file server) by using the following command line statement:

```
dczyloader /path/servername [-serverparms]
```

where:

path Is the full path to locate the server.

servername
Is the name of the server to be loaded.

serverparms
Are parameters available for the server to be loaded; they depend on the server.

The `dczyloader` command is not used for the shared-file server. Instead, use the following command line statement:

```
shfile [-servparms]&
```

For information on the LANDP server parameters that can be included in the server loading statement, refer to the *LANDP Installation and Customization* manual.

System user group rights are required in order to start LANDP for AIX servers.

Stopping LANDP for AIX

1. Access the Control LANDP for AIX panel in SMIT.

   You can access this panel by selecting **Communications Applications and Services** on SMIT main menu and **IBM AIX LAN Distributed Platform/6000 (LANDP for AIX)** on the subsequent one.

2. Select **End specific LANDP for AIX server** from the Control LANDP for AIX panel.

System user group rights are required in order to end LANDP for AIX servers.

Using the LANDP.LOG file

The LANDP.LOG file can be used to store LANDP for AIX trace and error information. This can be very useful if you are attempting to debug a problem.

To use the LANDP.LOG file, you need to set the environment variable LANDPLOG, using the following statement:

```
export LANDPLOG=YES
```

The LANDP.LOG file is created on the /etc/landp directory.

Part 6. Utilities and Performance Tuning

This part describes various utilities you can use when installing and customizing LANDP. It also describes how to tune some of your LANDP servers, to obtain better performance.

It contains the following chapters:

Chapter 15. Run-time utility programs

This chapter explains some utility programs provided with LANDP to be used at run-time. Those programs help you:

- Update communication variable data
- Update workgroup variable parameters
- Verify configuration and system files

Communication variable data

Often organizations using the LANDP family have numerous production sites similarly equipped and configured. Because of this, there are usually several sets of almost identical customization files. The only differences are the definitions for the communication parameters that need to be unique, such as token-ring adapter address, physical unit address for multi-dropped SDLC lines, and so forth.

To change the communication definitions, LANDP for DOS supplies a facility so that you do not have to go through the entire customization process. This facility is provided by the VARDAT.EXE program, which updates the communication configuration records using the data stored on the fixed-format VARDAT.CFG file.

To create and distribute variable data, create the VARDAT.CFG file. You can create the file in the customization workstation or, to avoid distribution, in a workstation in a workgroup. Note that, during change of communication definitions, user intervention is required.

You can also create the VARDAT.CFG file in a host computer, and implement an application to create the communication variable data for each workstation in a workgroup.

If the VARDAT.EXE program does not find a file named VARDAT.CFG, it assumes that there is no variable data for the workgroup configuration.

Variable data record format

The characteristics of the file are as follows:

- Organization: Sequential
- Record length: Fixed, 60 bytes (plus the end of line indicator X'0D0A')

- Number of records: One for each communication server in the workstation for which the customized communication data is to be updated. An exception to this is for the X.25 server, which can have a record for changing the network user address and as many records as necessary to change the permanent and switched circuits. The fields that do not need to be updated can be left blank.

The records must have the formats shown in the table below. Create only the records that apply to the communication devices affected by the update. If the record contains optional fields, include only those that contain changed information.

Table 6 (Page 1 of 2). Variable data record formats for VARDAT utility

Field Description	Length (Bytes)	Form	Type	Value	Note
DCA Variable Data Record:					
Service Identification	8	C	R	"DCADLC "	
Number of Buffers	4	N	R	"nnnn"	1
1					
Filler	48	C	R	" "	
SDLC Variable Data Record:					
Service Identification	8	C	R	"SDLC "	
Number of Buffers	4	N	O	"nnnn"	1
Physical Unit Address	2	X	O	"xx"	2
Identification Number	5	X	O	"xxxxx"	2
Filler	41	C	R	" "	
Token-Ring Variable Data Record:					
Service Identification	8	C	R	"TRDLC "	
Number of Buffers	4	N	O	"nnnn"	1
PC Local Admin Address	8	X	O	"xxxxxxxx"	1, 7
Host Local Admin Address	8	X	O	"xxxxxxxx"	1
Identification Number	5	X	O	"xxxxx"	2
Filler	27	C	R	" "	
X.25 Variable Data Record: Network User Address					**3**
Service Identification	8	C	R	"X25DLCNA"	
Number of Buffers	4	N	O	"nnnn"	1
Network User Address	15	N	O	"n(15)"	1
Included in Packet Call	1	N	O	"n"	6
Filler	32	C	R	" "	

Table 6 (Page 2 of 2). Variable data record formats for VARDAT utility

Field Description	Length (Bytes)	Form	Type	Value	Note
X.25 Variable Data Record: Permanent Circuits					**4**
Service Identification	8	C	R	"X25DLCSA"	
Connection Type	1	N	R	"P"	
No. of The Perm Circuit	4	N	R	"nnnn"	1
Filler	30	C	R	" "	
Identification Number	5	X	R	"xxxxx"	2
Filler	12	C	R	" "	
X.25 Variable Data Record: Switched Circuits					**5**
Service Identification	8	C	R	"X25DLCSA"	
Connection Type	1	N	R	"I","O","B"	
Filler	4	C	R	" "	
Old Subscriber Address	15	N*	R	"n"(15)	1
New Subscriber Address	15	N*	O	"n"(15)	1
Identification Number	5	X	O	"xxxxx"	2
Filler	12	C	R	" "	

Definitions for Form column (field format):

- C - Character
- X - Hexadecimal from 0 to F
- N - Numeric
- N* - Numeric filled with blanks at right

Definitions for Type column:

- R - Required field
- O - Optional field

The Value of the fields must adhere to the following rules, depending on the DLC. The field format is validated according to these rules.

- DCADLC record:
  - The number of buffers must be in decimal format. The range is from 10 to 100.
- SDLC record:
  - The number of buffers must be in decimal format. The range is from 8 to 999.
  - The physical unit address must be in hexadecimal format.
  - The identification number must be in hexadecimal format.
- TRDLC record:
  - The number of buffers must be in decimal format. The range is from 16 to 216.
  - The PC Local Administrative Address must be in decimal format.

- The Host Local Administrative Address must be in decimal format.
- The Host and PC Local Administrative Address must have the first byte less than 8.
- The Host and PC Local Administrative Address cannot be identical.
- The Identification Number must be in hexadecimal format.

- X25DLCNA record:
  - The number of buffers must be in decimal format. The range is from 20 to 2000.
  - The Network User Address must be numeric. Its length can be from 5 to 15.
- X25DLCSA record:
  - The connection type must be B, O, I or P.
  - The number of permanent circuit must be decimal if the circuit is permanent.
  - The old and new subscriber address must be decimal and their length from 0 to 15.
  - The identification number must be in hexadecimal format.

Notes:

1. These fields are in numeric ASCII format. LANDP programs convert them to the proper internal format at loading time.

2. These fields are the ASCII image of a hexadecimal field. LANDP programs convert all bytes from 0 to F.

3. You can have only one network user address. This is identified by the header, X25DLCNA.

4. You can have as many permanent circuits as necessary. These are identified by the header X25DLCSA and connection type P. The number of permanent circuits in the record must exist in the model configuration file. The only part of this record that is considered variable data is the identification number.

5. You can have as many switched circuits as necessary. These are identified by the header X25DLCSA and connection types I (incoming), O (outgoing), or B (both ways).

   The old subscriber address in the record must exist in the model configuration file and be unique, as only one physical unit is allowed per subscriber address for incoming circuits and for outgoing circuits. After updating, all the subscriber addresses of each type must be different. The only part of this record that is considered variable data is the new subscriber address and the identification number.

6. This field can be updated only by manually editing the VARDAT.CFG file. Valid input for the field is:

   0 Telephone number not included in packet call
   1 Only telephone number included in packet call
   2 Country code, subcode, and telephone number included in packet call

7. You must change the following statement in your CONFIG.SYS file:

   ```
   DEVICE=DXMC0MOD.SYS address
   ```

   where *address* is the PC local administration address.

Updating communication configuration records

The program VARDAT.EXE is copied by the customization utility program to the operational diskettes for the gateway workstations in the branches and loaded before any server. VARDAT.EXE searches for a file called VARDAT.CFG. This is the file that contains the communication variable data. VARDAT.EXE performs the following steps:

1. Verifies if a VARDAT.CFG file exists in the directory. If the file does not exist, the program ends normally.
2. Validates file and record consistency.

   If errors are found, the program generates a return code. See *LANDP Problem Determination*.

   If no errors are found, the program updates the communication configuration files and deletes the variable communication data file VARDAT.CFG.

Workgroup variable parameters

OS/2 DOS 6000 NT

The workgroup variable run-time parameters program (VARPARM.EXE) enables you to change, at run-time:

- The workgroup ID used by the system manager server.

  During customization, for each workgroup you specify the workgroup name in the NAME keyword of the corresponding LANCONF vector. The workgroup name becomes the default workgroup ID.
- The workgroup suffix used when establishing LANDP sessions.

  If you select suffix usage through the SUFFIX keyword in the LANCONF vector, the workgroup name becomes the default workgroup suffix.
- The 4721 printer ID used in the workgroup.

  During customization, for each 4721 printer you specify the 4721 printer ID in the corresponding PAR&SP21 keyword (parameter 2) of the LWSCONF vector.

In the run-time environment, when AUTOFBSS is running, the program VARPARM.EXE is loaded. The program searches for the VARPARM.SPC file, validates it, and changes the workgroup ID, the workgroup suffix, or the 4721 printer ID, according to the specifications in that file.

You can create the VARPARM.SPC file using any *text editor*. The file has the following format:

```
LANCONFG NAME = newwgid,
         SUFFIX = newsuffix,
         ID4721 = new4721id
```

where:

newwgid Is the new workgroup ID to be used by the system manager server.
newsuffix Is the new workgroup suffix to be used when establishing LANDP sessions.

`new4721id` Is the new 4721 printer ID to be used in the workgroup.

Note: Position 1 *must* be a blank, and the statement must start in position 2 or greater.

The VARPARM.SPC file is renamed to VARPARM.OLD at the end of the process, so that it is not processed each time you run AUTOFBSS.

If the VARPARM.SPC file is not available, the VARPARM.EXE program ends with no error and no change is made. If errors are found in the VARPARM.SPC file, the VARPARM.EXE program generates a return code. See *LANDP Problem Determination*.

Verification programs

OS/2 DOS 6000 NT

This section explains the utility programs provided to check record formats, profiles and parameters, and installation paths.

Installing updated configuration files

Changes in the record definitions or in the electronic journal, store-for-forwarding, forwarding, or shared file server profiles can cause unpredictable errors at run-time. Make sure that the values specified in the new profiles are compatible with the existing shared files.

To prevent this, the customization program creates all the configuration files related to these definitions with a file name extension that is different from the one used at run-time:

Server profiles	From Customization	At Run-Time
Shared file profile	.DBN .PCN	.DBD .PCB
Electronic journal profile	EJOU.PRN	EJOU.PRO
Store-for-forwarding profile	STOR.PRN	STOR.PRO
Forwarding profile	FORW.PRN	FORW.PRO

In the run-time environment, when AUTOFBSS is running, the program NEWCFG.EXE is loaded. If any of the configuration files created by a new customization are found, the program NEWCFG.EXE checks to see if the same configuration file also exists with the different run-time name.

- If only the files with the name given in customization exist, the program renames them to the name used at run-time and the LANDP program is loaded.

- If the same file exists with the name given at customization and the name used at run-time, this indicates that after the file was customized and the LANDP program was loaded, a new version was customized. As this can cause errors, NEWCFG.EXE sends a return code and ends the loading procedure. See *LANDP Problem Determination*.

  You can then choose to delete either:

  - The files created by the new customization.
  - The old files. However, you must first be sure that no change was made in customization that affected the internal structure of the databases.

Installing and validating system files

The following is a list of files that must be in the paths specified during the software building process.

- For the ASCII-EBCDIC translation server (only Taiwan):
  - TBLASC.088
  - TBLEBC.088
  - TBLTAI.088
- For the system manager server:
  - Workgroup common data file (FBSS#GDT)
  - User profile file (FBSS#USP)
- For the system manager operator:
  - Overlay files (.OVL): for LANDP for DOS only
  - Message files (SM0.SMG, SM1.SMG, SM2.SMG, SM3.SMG, SM4.SMG, SM5.SMG)
- For the record definition facility:
  - Record definition file (RDF.CFG)
- For RCMS:
  - Logical name file (RCMS.LNF)
  - Message file (RCME.MSG)
  - EBCDIC-to-ASCII conversion table
  - ASCII-to-EBCDIC conversion table

The installation and validation utility program EHCVAL.EXE checks the customized paths in the FBSSPATH.DAT file against the servers being loaded. EHCVAL validates whether the path exists, and if it does not, enables the creation of the path. The program also validates the existence of all system files needed, and copies or updates all files required by the servers to be in the path. The copied files are erased from the original directory. In the run-time environment, when AUTOFBSS is running, the program EHCVAL.EXE is loaded.

Error messages, information messages, and prompts that require a response may be displayed. The messages are contained in the EHC.MSG file, or in the EHCMSG.DLL file for LANDP for Windows NT; see the *LANDP Problem Determination* book.

Chapter 16. Migrating and generating customization data

This chapter contains instructions for:

- Migrating customization data generated by the FBSS customization program
- Generating customization data in vector format

You can migrate customization data provided for:

- IBM Financial Branch System Services V. 2.1.1, or later
- IBM Financial Branch System Services/2 V. 1.0, or later

Customization data provided for PC/Integrator, and PC Integrator/2 can also be migrated.

Selecting workgroup configurations

This section applies when you want to process more than one workgroup configuration at the same time.

The LANLIMIT.SPC file is located in the EHCCUS directory. You can edit the LANLIMIT.SPC file using any *text editor*, and specify the workgroup configurations to be processed.

- To *select* workgroup configurations, use INCLUDE statements. Those workgroup configurations not specified, explicitly or implicitly, will not be processed. The format of the statement follows:

  ```
  INCLUDE = langroup\lanname
  ```

- To *exclude* workgroup configurations, use OMIT statements. Those workgroup configurations not specified, explicitly or implicitly, will be processed. The format of the statement follows:

  ```
  OMIT = langroup\lanname
  ```

Note that if *langroup* is omitted, EHCCUS is assumed.

You cannot use both INCLUDE and OMIT statements at the same time. Comments must start with /* and end with */.

The LANLIMIT.SPC file provided with the customization program has the following contents, and specifies that all the workgroup configurations will be processed.

```
INCLUDE = *
```

You can modify the LANLIMIT.SPC file to meet your requirements. Three examples follow.

Example 1:
Only the workgroup configurations located in the GROUP2 directory, and the workgroup configuration named CONF47 and located in the GROUP4 directory, will be processed.

```
/* LANLIMIT.SPC Example 1*/
   INCLUDE = GROUP2\*
   INCLUDE = GROUP4\CONF47
```

Example 2:
All the workgroup configurations, except for those located in the GROUP6 directory, will be processed.

```
  /* LANLIMIT.SPC Example 2 */
   OMIT = GROUP6\*
```

Example 3:
Only the workgroup configurations located in the GROUP8 directory, and the workgroup configuration named CONF13 and located in the EHCCUS directory, will be processed.

```
/* LANLIMIT.SPC Example 3*/
   INCLUDE = GROUP8\*
   INCLUDE = \CONF13
```

Return codes

The return codes generated by both the migration and the generation procedures are classified into four types. The following list shows the types of return codes, starting with the least severe. An identifier for each type appears in parentheses.

1. *Informative* (I) return code: some input may be missing or incorrectly specified.
2. *Attention* or *Warning* (W) return code: some secondary functions may not work properly at the production sites.
3. *Error* (E) return code: some LANDP functional areas will not work properly.
4. *Severe* (S) return code: the execution will be ended.

The identifier of the type of return code is displayed as the last character of the return code. For example, the following is displayed on the screen:

```
00039W   This is an attention or warning return code. It ends with a 'W'.
```

When running the migration or the generation procedure, you can specify the highest severity allowed.

Migrating

This section explains how to perform the migration step. It shows the data used by the migration procedure, and how to start that procedure.

The migration procedure reads data from the directories containing FBSS customization data. Then, using that information, the procedure generates data in the internal repository.

Source data for migration

The migration procedure uses the customization information that the FBSS customization program generated in the directories in the following list. They are located either in the root directory or in a first level subdirectory. The list also shows the contents of the directories.

Directory	Contents
FBSS	FBSS programs
FBSSCUS	Customization results
DIFFERENT	Customization results.

Note that DIFFERENT stands for the name of any directory you created during customization to locate your workgroup configurations.

The directories containing the information for each workgroup configuration are located under the FBSSCUS and DIFFERENT directories. In turn, the directories containing information for each workstation are located under the workgroup configuration directories.

If you used the **automatic building** option of the FBSS customization program, the migration procedure uses the automatic building file created during the building process.

If some customization data cannot be obtained from existing FBSS customization data, or if the automatic building file does not exist, the migration procedure provides defaults, when possible, or generates ? symbols. When ? symbols are generated, the migration procedure ends with a non-zero return code.

When it is displayed on the screen, you should:

1. Generate the LANCONF.SPC files corresponding to the workgroup configurations processed by the migration procedure, using the generation procedure.
2. Edit those files, replacing the ? symbols by the appropriate information.

When you have finished updating the LANCONF.SPC files, run the validation procedure.

To convert an FBSS/2 workstation configuration to a LANDP for OS/2 workstation configuration, that is to migrate from a workstation running IBM OS/2 EE V1.3 to a workstation running IBM OS/2 Warp V4.0, or higher, you have to:

1. Migrate the FBSS customization data corresponding to that FBSS/2 workstation.
2. Generate the LANCONF.SPC file containing the migrated data.
3. Edit that file, updating the type of the workstation to IBM OS/2 Warp V4, or higher.

When you have finished updating the LANCONF.SPC file, run the validation procedure.

To convert *all* the FBSS/2 workstation configurations to LANDP for OS/2 workstation configurations, the migration procedure provides a parameter to specify it.

Starting the migration procedure—MIGRATE

You can display *online information* about the MIGRATE procedure. From the EHCCUS directory, enter:

```
MIGRATE ?
```

To start the migration procedure, from the EHCCUS directory, enter:

```
MIGRATE [parm1] [parm2] [parm3] [parm4] [parm5] [parm6]
```

where:

parm1 Is an optional parameter to specify the type of data to be processed. The parameter value can be:

COMMON	To process common data, such as defaults, records, profiles, and tables.
LAN	To process workgroup configuration data.

If you specify LAN, but there is no common data in the internal repository, COMMON is also assumed. If the parameter is omitted, and there is no common data in the internal repository, both common and workgroup data are processed.

parm2 Is an optional parameter to specify the workgroup configuration to be processed. It applies when you process only one workgroup configuration.

If the parameter is used, only the workgroup configuration specified will be processed, no matter the LANLIMIT.SPC file contents.

The parameter format is:

```
lanpath\lanname
```

If *lanpath* is omitted, FBSSCUS is assumed as the directory where the source workgroup configuration data is located. The migrated workgroup configuration data is located in the current path, that is, in the EHCCUS directory.

parm3 Is an optional parameter to specify the highest severity of return codes allowed. The parameter value can be:

1 To admit only informative return codes. Higher severity results in execution ending.

2 To admit informative and attention return codes. Higher severity results in execution ending.

3 To admit informative, attention, and error return codes. Higher severity results in execution ending.

The default is 1. For further information, refer to “Return codes” on page 240.

parm4 Is an optional parameter to specify the path of the directory where the FBSS customization data is located. The parameter format is:

`d:\dirname\`

where *d* is the drive identifier and *dirname* is the name of the directory. The default is the current first level subdirectory.

parm5 Is an optional parameter to specify that all workstations running FBSS/2 are migrated to workstations running LANDP for OS/2. If the parameter is omitted, after migration, they remain FBSS/2 workstations.

The parameter value must be `NEWOS`.

parm6 Is an optional parameter to specify that all workstations running FBSS (DOS) are migrated to workstations running LANDP for DOS. If the parameter is omitted, after migration, they remain FBSS (DOS) workstations.

The parameter value must be `NEWDOS`.

The parameters can be specified in any order. You will get the CUSPARM.LST file in the EHCCUS subdirectory, containing pointers to any problems that arise.

To avoid long processing, it is strongly recommended to run the migration procedure specifying first the COMMON parameter value. Then, run the procedure specifying the LAN parameter value.

In addition, when processing workgroup configuration data, you can select the workgroup configurations affected by the process. See “Selecting workgroup configurations” on page 239.

Generating data

This section explains how to perform the generation step. It shows how to start the generation procedure, and the data generated by that procedure.

The generation procedure reads data in the internal repository, created either by the migration procedure or by the validation procedure, and generates vectors in the .SPC files.

If an .SPC file with the same name and in the same path already exists, the procedure creates a backup of the existing .SPC file, with the same filename and extension BAK. If a .BAK file corresponding to the .SPC file already exists, the procedure erases the existing .BAK file.

If an .SPC file is lost, you can recover it running the generation procedure and using the *current data* in the internal repository.

Starting the generation procedure—GENSPEC

You can display *online information* about the GENSPEC procedure. From the EHCCUS directory, enter:

```
GENSPEC ?
```

To start the generation procedure, from the EHCCUS directory, enter:

```
GENSPEC [parm1] [parm2] [parm3] [parm4]
```

where:

parm1 Is an optional parameter to specify the type of data to be processed. The parameter value can be:

COMMON	To process common data, such as defaults, records, profiles, and tables.
LAN	To process workgroup configuration data.
MODELS	To process model configuration data.

The default is LAN.

parm2 Is an optional parameter to specify the workgroup configuration to be processed. It applies when you process only one workgroup configuration.

If the parameter is used, only the workgroup configuration specified will be processed, no matter the LANLIMIT.SPC file contents.

The parameter format is:

```
langroup\lanname
```

If *langroup* is omitted, EHCCUS is assumed.

parm3 Is an optional parameter to specify the highest severity of return codes allowed. The parameter value can be:

1 To admit only informative return codes. Higher severity results in ending execution.

2 To admit informative and attention return codes. Higher severity results in ending execution.

3 To admit informative, attention, and error return codes. Higher severity results in ending execution.

The default is 1. For further information, refer to "Return codes" on page 240.

parm4 Is an optional parameter with only one possible value: DELETE.

If the parameter is specified, the customization data processed is automatically removed from the internal repository. Thus, after running the generation program, the data is stored only on the .SPC files.

If the parameter is not specified, the customization data processed also remains stored in the internal repository.

The parameters can be specified in any order. Return codes generated by the generation procedure are displayed on the screen.

To avoid long processing, it is strongly recommended to run the validation procedure specifying first the COMMON parameter value. Then, if it applies, run the procedure specifying the MODELS parameter value. Finally, run the validation procedure again specifying the LAN parameter value.

In addition, when processing workgroup configuration data, you can select the workgroup configurations affected by the process. See "Selecting workgroup configurations" on page 239.

Note: The GENSPEC utility runs in DOS, OS/2, and Windows NT only, although the data that is being generated can apply to all platforms.

Data in vector format

After generation of .SPC files, the customization data is stored in vector format in those files. The common data is stored in the COMMON.SPC file, in the EHCCUS directory.

The workgroup configuration data is stored in the LANCONF.SPC files, in the *yyyyyyyy\xxxxxxxx* directories, where:

yyyyyyyy Is the name of a group of workgroup configurations. The customization program default is EHCCUS.

xxxxxxxx Is the name of a workgroup configuration.

The vectors in a LANCONF.SPC file are:

LANCONF
LWSCONF

The model configuration data is stored in the MODELS.SPC file, in the EHCCUS directory. The vectors in that file are:

LANMODEL
LWSCONF
SVRMODEL
WSMODEL

Because the model concept has been introduced with the LANDP customization, no data exists in the FBSS customization files to generate the model configuration vectors. These vectors are generated only if you have already created a MODELS.SPC file, and run the validation procedure specifying the MODELS parameter value.

Chapter 17. Listing customization data

This chapter explains the following utility programs, which enable you to list customization data in different formats:

- The LISTRTOC program lists customization data in vector format. Refer to “Listing internal repository—LISTRTOC.”
- The EHCLIST program lists customization data structured in workgroup configurations. Refer to “Listing workgroup configurations—EHCLIST.”

Note: The utilities run in DOS and OS/2 only, although the data that is listed can apply to all platforms.

Listing internal repository—LISTRTOC

The LISTRTOC program is provided to generate the internal repository table of contents. The table of contents is generated on a file named CUSPARM.LST.

You can display *online information* about the LISTRTOC procedure. From the EHCCUS directory, enter:

```
LISTRTOC ?
```

To start the procedure, from the EHCCUS directory in your customization workstation, enter:

```
LISTRTOC [parm1]
```

where:

parm1 Is the type of data to be processed. The parameter value can be:

`COMMON` To process common data. The program generates a list of the common vectors.

`LAN` To process workgroup configuration data. The program generates a list of the workgroup configuration vectors.

`MODELS` To process model configuration data. The program generates a list of the workgroup model configuration vectors.

The default is `LAN`.

Listing workgroup configurations—EHCLIST

The EHCLIST program is provided to list and view the structure of your workgroup configurations. You can use this program after you have defined at least one workgroup. Two types of listings can be obtained: one oriented towards servers and the other towards services.

You can display *online information* about the EHCLIST procedure. From the EHCCUS directory, enter:

```
EHCLIST ?
```

To start the procedure, from the EHCCUS directory in your customization workstation, enter:

```
EHCLIST parm1 parm2 parm3 parm4 parm5
```

where:

parm1 Is the group of the workgroup configuration you wish to list.

parm2 Is the name of the workgroup configuration you wish to list.

parm3 Is the listing format.

The parameter has two possible values:

Y To have the listing printed continuously
N For separate pages.

parm4 Is the output device.

You can print the report or store it on a file. To print the report, enter the device name of the printer port to be used: LPT1, LPT2, or LPT3.

To store the report on a file, enter the name of the file. The file will be placed under the EHCCUS subdirectory.

parm5 Is the listing type.

The parameter has two possible values:

1 To have the listing oriented towards servers
2 To have the listing oriented towards services.

The following is an example of a *servers listing* for a workgroup configuration that contains an OS/2 workstation with an SNA server, MQSeries Link server, and a user-defined server, and an NT workstation with an ODBC query server, shared-file server and a 47x2 financial printer server.

SERVERS	SERVER PCS									
	GW OS/2	S1 NT								
EHCMQ##	*									
EHCODB01		*								
PR47X2##		*								
SHFILEBA		*								
SNA##	*									
USERSVR1	*									

In this listing you can see all the servers installed in the workgroup and the workstations in which they are installed.

The *services listing* provides a complete picture of your workgroup configuration and the connections between the server workstations and the client workstations. For a complete listing of all the configurations, make sure you define, under communication configuration customization, all the communication parameters. If you have not defined the communication parameters, the listing will show question marks where services need to be defined.

Following is an example of a services listing for the same workgroup configuration model as above:

SERVED PCs AND SERVICES	SERVER PCS									
	GW OS/2	S1 NT								
S1 NT										
PR47X2PP		*								
W1 OS/2										
USERSVR1	*									
EHCODB01		*								
PR47X201		*								
SHFILEBA		*								
W2 DOS										
SNA										
EMU32701	*									
EMU32702	*									
EHCMQ01	*									
SHFILEBA		*								
W3 NT										
EHCMQ01	*									
EHCMQ02	*									
EHCODB01		*								
PR47X202		*								
SHFILEBA		*								

The server workstations are shown horizontally at the top of the listing. On the left side, shown vertically, are the client workstations and the services they receive. For example:

- Workstation S1 has a PR47X2## server installed, and also receives the services of that server.

- The three workstations, W1, W2, and W3 make use of various services provided by GW and S1, with workstation W2 using two sessions of the 3270 emulator.

Chapter 18. Performance tuning

This chapter contains information on how to tune the performance of some LANDP components. The following list shows what these specific components are, and where to find the related information:

- Shared-file server
- LANDP for OS/2 query server, see page 252
- LANDP for Windows NT ODBC server, see page 252
- LANDP for AIX query server, see page 254
- Shared DOS directory services, see page 254
- Windows 3.1 LANDP for DOS, see page 255

Tuning the shared-file server

OS/2 DOS NT

The shared-file server is designed to take advantage of the system cache. Therefore, increasing the system cache will, in most cases, result in an increased server throughput.

The ideal cache size is the one resulting from adding the data and index file sizes that the server manages. However, when memory limitations do not allow having such a large cache size, a cache size that is about 20% of the previous addition will also result in improved performance.

You can specify the cache size in the following ways:

- For DOS: adding a cache device driver such as IBMCACHE.SYS or SMARTDRV.SYS to your CONFIG.SYS file, with a cache size large enough for your environment.
- For OS/2 with FAT drives: setting the parameter *xx* of the `DISKCACHE=`*xx* statement in your CONFIG.SYS file to a higher value.

  For OS/2 with HPFS drives: setting the parameter *xx* in the statement `IFS=HPFS.IFS /C:`*xx* to a higher value.
- For Windows NT: the cache is controlled dynamically by Windows NT.

You may also tune the server loading time parameter `/B:`*nnn* (number of index buffers) in order to improve the performance, if there is enough memory available in the system. The value *nnn* specifies the number of additional (above 10) 1KB index buffers to be allocated. More index buffers increase system throughput, but also reduce the amount of free storage available for the server workstation. A rule of thumb is that the number of buffers should be 10 per workstation using the shared-file server simultaneously. A practical limit is approximately 100, depending on available storage size. The maximum value for *nnn* is 484.

Another factor that must be considered is that the more buffers you have, the more likely you are to lose index data when the shared-file server is abnormally closed. Thus, if many index buffers are allocated, and the server workstation is switched off with a transaction still in process, or if no RF function has been called, an automatic index rebuild is performed the next time the server is loaded.

Note that these considerations are especially important when you specify a small number of simultaneously open files with the /F loading parameter.

Tuning the LANDP for OS/2 query server

Since the query server is highly dependant on the database configuration parameters, in terms of performance, you may want to modify some of these parameters in order to increase server throughput.

It is not particularly helpful to increase the system cache for this purpose, because the IBM Database Server for OS/2 Warp, Version 4 handles its own disk cache. Instead, it is highly recommended to set the **buffer pool size** parameter in the Change Database Configuration Tool panel of the Database Manager to a suitable value for the query server, since it is a dynamic SQL server.

To set the buffer pool size, select **Configuration Tool** in the Database Server main menu. After that, select **Configuration** and then **Change Database**. Here you can increase the number of pages for a buffer pool. 200 pages can be a good startup value. You may also need to change the **Default application heap** size to a larger setting.

Other configuration parameters do not affect performance considerably, unless they generate contention problems in a concurrent environment. If this is your situation and your applications receive many RL return codes, you can then verify that these problems are not caused by lock escalation. If they are caused by lock escalation, increase the storage allocated for lock lists in the same panel.

Even if these parameters are set to suitable values, contention problems may still occur. Deadlock and resource racing are normal in concurrent database environments. To avoid contention problems as much as possible, it is recommended that you change the isolation level of the server from *repeatable read* to *cursor stability*. You can do this by rebinding the requester program against the database to be used. Enter at the command line:

```
SQLBIND EHCSQLRQ.BND database-name /I=CS
```

You should write applications that access the query server in a way such that they can work properly with this isolation level.

The server loading parameters /P, /T, and /F affect considerably the throughput of the server. The number of processes, /P, and the number of threads, /T are related to the number of workstations that are accessing the server at the same time. In order to improve performance, fast cursor operation, `/F`, is recommended for applications that make extensive use of the functions GN, GP, GU, and GW, or for applications that access large data tables.

Tuning the LANDP for Windows NT ODBC query server for DB2

The performance of the ODBC query server is highly dependent on the database configuration parameters and the ODBC configuration parameters. To increase server throughput, you may want to modify some of these parameters. See the vendors' manuals for ideas on performance tuning.

In the ODBC Query Server LOADER statement, the /MT and /T parameters can be adjusted to increase throughput. These parameters handle the number of threads that are used to send requests to the ODBC Device Driver in use. /MT indicates the maximum number of threads that the ODBC Query Server can start (up to a maximum of 128). /T indicates the number of threads that the ODBC Query Server starts at initialisation. For every request received that can not be passed to an existing thread, the ODBC Query Server starts a new thread until the /MT value is reached.

To improve system throughput, you can increase the buffer pool size (`buffpage` parameter for DB2). To change this parameter, use the Command Line Processor or the Control Centre database configuration panel. For guidance, refer to the DB2 manuals. It may also be helpful to change the default Application Heapsize (`applheapsz`).

If deadlock contention occurs (for example, SQL code SQL0911N returned through a LANDP return code QE), you can adjust additional parameters. Suggestions:

1. In the ODBC device driver setup, change the Isolation Level (`txnisolation`) from *repeatable read* to *cursor stability*.
2. In the database configuration panel, adjust these parameters:

   Interval for Checking Deadlock (`dlchktime`)
   Lock Timeout (`locktimeout`)

   For guidance, refer to the DB2 manuals.

Tuning the LANDP for AIX query server

The LANDP for AIX query server performance can be influenced by the number of pool sessions specified, and the time-out value for SQL responses.

You can improve your query server performance by fine tuning the values of the loading parameters as follows:

- The `-p` parameter is used to specify the number of pool sessions. These pool sessions are established at query server startup. At run time, the query server can use these pool sessions whenever an Open Session, Open Online, or Open Query function is requested. If no pool session is available for the query server at run time, then the query server needs some time in order to open a new session.

  When specifying the -p parameter consider the following:

  – Specifying enough pool sessions for the query server at run time in order to minimize the time required for opening new sessions

  – Establishing a pool session will extend the startup time of the query server

  – Pool sessions will increase the system load, thus affecting system performance.

- Specifying a time-out value with the `-t` parameter that balances:

  – The time required for the query server to receive an SQL response
  – The probability that an application needs to request a function again.

For more options that may affect the query server performance, refer to the information related to the RDBMS you are using.

Tuning the shared DOS directory services

This section provides general recommendations for tuning the performance in a workgroup where you want to use the shared DOS directory services.

If you install the shared DOS directory services in a workgroup, you have to take into account the buffer size customized for the shared DOS directory client workstations in order to calculate the buffer pool size for the server workstation.

If you do not define a buffer pool size that is large enough, time-out situations may frequently occur. The client workstation will then display the following message:

```
Not ready reading drive...
Abort, Retry, Ignore?
```

To achieve optimum performance, the buffer pool size of the server workstation should be large enough so as to allocate all incoming requests, which are received from all the client workstations. However, this solution could lead to a shortage of memory in the server workstation, therefore you should find a compromise between performance (that is, buffer size of the shared DOS directory client workstations) and memory requirements (buffer pool size of the shared DOS directory server workstation). This consideration is especially critical when you are going to make copies of large remote files.

Sizing the client workstation buffer and the server workstation buffer pool must meet the specific requirements of every LANDP workgroup.

Tuning Windows 3.1 under LANDP for DOS

You can modify some Windows 3.1 settings in order to improve LANDP for DOS performance. For information about which Windows 3.1 settings to modify, refer to the SYSINI.WRI Windows file.

You should investigate the following Windows 3.1 settings to achieve optimum system performance, depending on your specific environment:

- MinTimeSlice
- WinTimeSlice
- KeyIdleDelay
- KeyBoostTime

Also, the Detect_Idle_Time setting of both the LANDP kernel and the LANDP application program information files (PIFs) should be set OFF in most cases to get better performance.

You can also improve LANDP for DOS performance in a Windows environment by loading EHCWKDE after LANDP for DOS kernel on the same VM. EHCWKDE is supplied with LANDP for DOS in the EHCD500 subdirectory of the workstation where LANDP for DOS is installed. After unloading LANDP for DOS, press the **Q** key to exit EHCWKDE and continue working on this VM.

Chapter 19. System maintenance—VERSION

OS/2 DOS 6000 NT

This chapter contains information about the VERSION utility program. This utility program is helpful in the maintenance of LANDP products and in the problem determination process.

The VERSION utility program determines the current version level of LANDP product components and keeps track of maintenance fixes installed on the components. The program also generates information that can be helpful when resolving LANDP problems.

IBM support representatives will require this information if you request assistance.

Using the VERSION utility program

The VERSION utility program performs the following functions:

- Determines the current version of the component program files
- Determines the last authorized program analysis report (APAR) or program temporary fix (PTF) applied to:
  - LANDP product components
  - FBSS, PC/Integrator, and PC Integrator/2 programs
- Displays the program name, program version, and last-applied APAR or PTF level

When you run the VERSION utility program, it creates a file called VERSION.TXT.

The VERSION program writes to this file the component program name, version, APAR, and PTF information.

You can run the VERSION utility program on any LANDP for DOS, OS/2, Windows NT, and AIX workstation.

Starting on a LANDP for DOS, OS/2, and Windows NT workstation

Start the VERSION utility program from the EHCMAINT directory on the DOS or OS/2 customization workstation, or from the directories that contain the LANDP product components on the production site workstation.

To start the program, do the following:

1. Complete *one* of the following steps:
   - Customization workstations: Make EHCMAINT the current directory.
   - Production site workstations: Make the directory that contains the components the current directory.

2. Enter the following command:

   ```
   VERSION [path1] [/o:path2]
   ```

   where:

   path1
   : Is the path and drive the VERSION program searches. Specify the path and drive where the LANDP programs reside.

     If you do not specify this parameter, the VERSION utility program searches only the current directory and the LANDP reserved subdirectories.

   /o:*path2*
   : Is the target drive and path where the VERSION utility program places the output file, VERSION.TXT.

     If you do not specify this parameter, the VERSION program places VERSION.TXT in the current directory.

Starting on a LANDP for AIX workstation

When you run the VERSION utility program on a LANDP for AIX workstation, you can specify the name of the file where the information will be written.

Start the VERSION utility program on the production site workstation, from the directories that contain the LANDP for AIX component files. To start the program, do the following:

1. Make the directory that contains the component files the current directory.
2. Enter the following command:

   ```
   dczyversion [>filename.extension]
   ```

   where *filename.extension* is the name of the file where the information will be written. This parameter is optional.

Note: You can also start this procedure from the SMIT menu.

Reading the output

The VERSION utility program generates this output when it runs successfully and all information is available:

```
ID=ssssssss    VL=vrmxx    nnnnnnn    xxxxxxx
```

ID=*ssssssss*
: Is the name of the program.

VL=*vrmxx*
: Is the version level:

  vr is a variable identifying which LANDP Version 5 components use a program:

D5	LANDP for DOS
O5	LANDP for OS/2
N5	LANDP for Windows NT
B5	LANDP for DOS and LANDP for OS/2
S5	LANDP for OS/2 and LANDP for Windows NT
U5	LANDP for DOS and LANDP for Windows NT

A5	LANDP for DOS, LANDP for OS/2, and LANDP for Windows NT
AR	LANDP for AIX

For the FBSS products, PC/Integrator, and PC Integrator/2, *v* is the version and *r* is the release.

m is the modification level.

xx is a sequence number that corresponds to the number of modifications applied to the LANDP program.

nnnnnnn
Is the last APAR or PTF applied to the program. (This information does not appear for LANDP for AIX.)

xxxxxxx
Is either:

ffffffff.fff The name of the file (LANDP for DOS, OS/2, and Windows NT)

or

5765-482 The LANDP for AIX program number

For information and procedures for installing program fixes, see Chapter 20, “Applying program fixes—APPLYFIX” on page 261.

Chapter 20. Applying program fixes—APPLYFIX

OS/2 DOS 6000 NT

This chapter describes how to install program fixes for LANDP product components. Program fixes are provided by the IBM Support Center (ISC). A fix is sometimes called a *program temporary fix* (PTF).

About program fix files

OS/2 DOS 6000 NT

The IBM Support Center uses the following naming convention for program fix files:

```
filename.ext
```

where:

filename Is the fix identification number.

ext Is the file extension, which is the first three characters from the **VL** field in the VERSION.TXT file. See “Using the VERSION utility program” on page 257 for information about VERSION.TXT.

An example of a program fix file name might be:

```
HC52249.D50
```

You must install program fix files on the workstation that is used for customization. To install program fixes, refer to the information that relates to the LANDP product components you are updating.

If you are updating LANDP for DOS, OS/2, and Windows NT components, see “Using the APPLYFIX utility program” on page 262. If you are updating LANDP for AIX components, see “Using other programs to apply fixes” on page 264.

Using the APPLYFIX utility program

Use the APPLYFIX utility program if you are installing program fixes for LANDP for DOS, OS/2, or Windows NT product components.

Note: APPLYFIX has to be run from a DOS or OS/2 customization workstation to apply fixes to the LANDP for Windows NT product.

The APPLYFIX utility program performs the following functions:

- Unpacking the program fix files
- Checking the component ID
- Checking the release level
- Checking the version level
- Checking the directory where the fix is to be installed
- Installing the program fix files
- Saving a backup copy of changed program files
- Creating or updating a history file, which is a log of the program changes that have been installed

Starting the process

OS/2 DOS

To start the apply-fix process, do the following:

1. Go to a DOS or OS/2 command prompt on the customization workstation.
2. Make EHCMAINT the current directory.
3. From EHCMAINT, enter the following command:

```
APPLYFIX [d:][\path\]filename.ext [/all]
```

where:

d Is the source drive that contains the fix file, usually drive A.

path Is the directory where the program fix resides.

filename.ext Is the name of the program fix file.

/all Applies all the changes from the program fix file, without showing the contents of the file.

Notes:

1. You can use wildcard characters instead of specifying the program fix file name and extension.
2. You can start the APPLYFIX program with the parameter `/?`, if you want online instructions about using the program.
3. You can press the F1 key for more online help after the program is started.

Examples:

- To install the program fix HC52249 that resides on a diskette in drive A, you would type the following:

  ```
  APPLYFIX A:\HC52249.D50
  ```

- To apply all the fixes, you would type:

  ```
  APPLYFIX A:\*.* /all
  ```

- To use wildcard characters, you would type the following:

  ```
  APPLYFIX A:\*.D50
  ```

  or

  ```
  APPLYFIX A:\*.*
  ```

- To install program fixes and view online instructions about the APPLYFIX utility program, you would type:

  ```
  APPLYFIX A:\HC52249.D50 /?
  ```

Installing program fixes

If you do not specify the `/all` option, the APPLYFIX utility program displays a screen containing information about the program fixes. From this screen, you can do the following:

- Install the program fixes by pressing the F10 key.
- View more information about the program fixes by pressing the F6 key.
- Get online help about using the APPLYFIX program by pressing the F1 key.

The APPLYFIX utility program checks the component ID, the release, the version level, and the directory where the program fix is to be installed. It then installs the fix and replaces product component files, as required.

The APPLYFIX utility program updates two history files by recording the changes made when program fixes were installed. The history files are named LANDP.HST and TOTS.HST. Both files reside in the EHCMAINT directory.

LANDP.HST and TOTS.HST are text files. You can view or print the files using the TYPE or PRINT commands from a DOS or OS/2 command prompt.

After you install program fixes, the updated product components can be copied to diskette or other media for transfer to the workstations in your workgroups.

Backing up and retrieving program files

When the APPLYFIX utility program replaces a file, it saves a backup copy of the replaced file in a directory named FIXSTORE. The file is saved with its program name and its version level number as the extension.

Example LANDP Version 5 backup file names

- LANDP for DOS files

  ```
  SPV.Dxx
  ```

- LANDP for OS/2 files

  ```
  SPV.Oxx
  ```

- LANDP for Windows NT files

  ```
  SPV.Nxx
  ```

- Files common to LANDP for DOS, OS/2, and Windows NT

  ```
  SPV.Axx
  ```

- Files common to LANDP for DOS and OS/2

  ```
  SPV.Bxx
  ```

- Files common to LANDP for OS/2 and Windows NT

  ```
  SPV.Sxx
  ```

In the above list `xx` is a numeric value identifying the version of the backed-up program.

You can retrieve files in the FIXSTORE directory by running the APPLYFIX utility program. You would start the APPLYFIX program, specifying the backup directory and program file that you want to retrieve. For example:

```
APPLYFIX ..\FIXSTORE\SPV.AXX
```

Note: After many updates, the FIXSTORE directory may occupy a large amount of disk space. Make sure you maintain and administer the FIXSTORE files properly.

Using other programs to apply fixes

If you are applying program fixes to LANDP for AIX product components, use the programs provided by RS/6000 systems.“Bibliography” on page 571 contains a list of books related to these systems.

- **LANDP for AIX program fixes**

  Install LANDP for AIX program fixes by calling SMIT and using the **Install Updates Only** option.

Part 7. Appendixes, Glossary, and Bibliography

This part includes various appendixes, showing a hands-on example and providing detailed vector descriptions for user servers, common data, and configuration data.

Appendix A. A hands-on example

To help you get a hands-on feeling for various LANDP configurations, samples are provided on the LANDP CD-ROM. After installing LANDP, you will find them in path

```
...\EHCCUS\SAMPLEn\
```

This appendix describes `SAMPLE1`, which shows the steps to customize a basic workgroup of three workstations. It provides a hands-on guide through the customization process, with basic modifications to the configuration.

SAMPLE1 is located in path

```
...\EHCCUS\SAMPLE1\
```

If you wish to modify or test this example, or find out more information about how it was written and configured, please refer to the individual chapters of this manual.

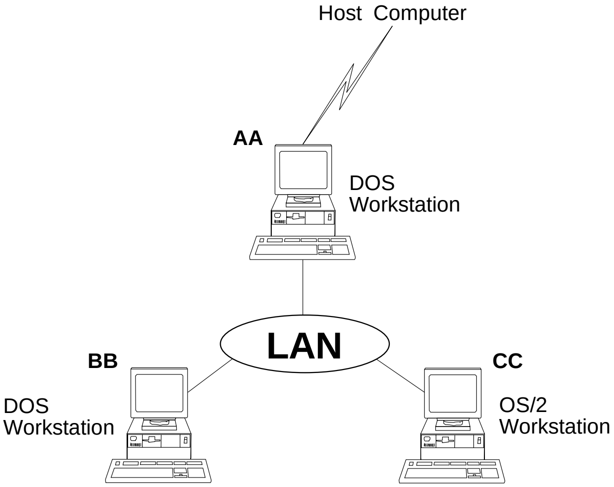
Background

SAMPLE1 shows all the steps and procedures needed to customize and obtain software for distribution to the work sites.

The example involves the following steps:

1. Installing
2. Creating common vectors
3. Editing common vectors
4. Validating common vectors
5. Editing workgroup vectors
6. Validating workgroup vectors
7. Obtaining software for distribution
8. Distributing the software

The workgroup in the example looks like this:



All workstations use SNA sessions to communicate with the host applications, and have the following requirements:

AA Workstation AA is DOS 7.1.

It contains the token-ring DLC server (TRDLC) and the SNA server (SNA##).

BB Workstation BB is DOS 7.1.

It uses two sessions of the SNA server in workstation AA, and receives services from the 3270 emulator.

CC Workstation CC is OS/2 Warp V4 or later.

It uses two sessions of the SNA server in workstation AA.

Step 1. Installing

Because the example is a mixed workgroup with both DOS and OS/2 workstations, you must have LANDP for DOS and LANDP for OS/2 installed in your customization workstation.

Example 1 assumes that you have installed LANDP in directory C:\LANDP The common and models customization files will be put in subdirectory EHCCUS, which is created by the installation program. All procedures will be called from the same subdirectory: C:\LANDP\EHCCUS.

Note: For NetBIOS support, you need to copy the following device drivers from the LAN Support Program installation diskette to the EHCD500 directory:

DXMA0MOD.SYS
DXMC0MOD.SYS
DXMT0MOD.SYS
DXMG0MOD.SYS

Step 2. Creating common vectors

Some servers in your workgroup may need common data specifications. For a reference list of which vectors you have to specify, refer to “Vectors - a quick reference” on page 54. If you do not need to specify common vectors, go directly to “Step 5. Editing workgroup vectors” on page 270.

In the example, defaults are required at common level. Therefore, you will create the common data default definitions.

To create the default definitions, start from the EHCCUS subdirectory (C:\LANDP\EHCCUS) and enter the following command:

```
   CREATE
```

The common data default definitions are created in the internal repository, and contain all the default values for your common data.

To generate the common vectors, enter:

```
GENSPEC COMMON
```

You may leave the default data and continue with defining workgroup vectors. If you want to see the data you have created, or make modifications to them, go to the next step.

Step 3. Editing common vectors

You created the common data default definitions in Step 2. This also created an editable COMMON.SPC file in the EHCCUS subdirectory. To edit the file, enter the following command from the same subdirectory:

```
   EDITSPC COMMON
```

Your COMMON.SPC file is now open and you can start editing.

In the example, you want to add an additional key definition for the 3270 emulator keyboard table. The key definition will be: `SCAN/ASCII 6D/00/END EMULATION`.

To add the key definition, take the default KBD3270 vector (with keyword EXTEN=KBD) and add one keyword:

```
   KEY=(6D,00,'END EMULATION')
```

Step 4. Validating common vectors

When you have specified the common vectors, you can check that they are correct. To call the validation program, enter from the same EHCCUS subdirectory:

```
   VALSPEC COMMON
```

The definitions of the COMMON.SPC file are validated, with a maximum allowed return code of 0.

If there are errors, attention items, or informative messages, the CUSPARM.LST file is created in the same subdirectory. This file contains the vectors and pointers to the messages or errors. Correct the data, and run the validation program again.

Once the common data definitions have been edited and validated, you can go on to define your workgroup specifications.

Step 5. Editing workgroup vectors

All the data for setting up workgroup definitions have to be defined for each workgroup customization.

The subdirectory where the workgroup definitions for this example will reside, is called SAMPLE1. Again, start from the EHCCUS subdirectory, and enter:

```
EDITSPC SAMPLE1 LAN
```

The LANCONF.SPC file is now open and ready to be edited. It is located in path C:\LANDP\EHCCUS\SAMPLE1.

If the path does not exist, or if you specify a different group and workgroup identification, the program asks if you want to create it. Enter Y, and the file LANCONF.SPC is created in the directory you specified.

All the vectors containing workgroup definitions for SAMPLE1 have to be written in this file. The LANCONF.SPC file is now open and you can start editing.

The workgroup vectors for the example are described below:

Workgroup configuration

This is an example of a LANCONF vector, which specifies the configuration for the workgroup:

```
LANCONF    GROUP=EHCCUS,
           NAME=SAMPLE1,
           WSNAMES=(AA,BB,CC)
```

Workstation configurations

After defining the LANCONF vector, the next step is to add the vectors for the individual workstation definitions.

This is the vector for workstation **AA:**

```
LWSCONF     NAME=AA,
            TYPE=DOS,
            SERVER=(TRDLC),
            PAR&TKR=(48,04,04,00000001,00000099,017,00000),
            SERVER=(SNA##)
            PAR&SNA=(ANY,SRV)
```

The parameter values for this vector are:

Keyword	Parameter Values	
NAME	**AA**	Is the workstation identifier.
TYPE	**DOS**	Is the operating system.
SERVER	**TRDLC**	Loads the token-ring DLC server.
PAR&TKR	**48** **04** **00000001** **00000099** **017** **00000**	Is the number of buffers. Both host and workstation use SAP 04. Is the workstation TR locally administered address (400000000001). Is the host TR locally administered address (400000000099). Is ID block 017. Is ID Number 00000.
SERVER	**SNA##**	Loads the SNA server.
PAR&SNA	**ANY** **SRV**	Session can be initiated by application or host. BID command managed by SNA server.

This is the vector for workstation **BB:**

```
LWSCONF     NAME=BB,
            TYPE=DOS,
            CLIENT=(SNA01,AA),
            SES&SNA=(AA,01,31),
            CLIENT=(SNA02,AA),
            SES&SNA=(AA,02,32),
            SERVER=(EMU32701),
            PAR&3270=(Y,N),
            SES&3270=(AA,1,33,TKR,ATR,DIS,KBD,SESSION1)
```

The parameter values for this vector are:

Keyword	Parameter Values	
NAME	**BB**	Is the workstation identifier.
TYPE	**DOS**	Is the operating system.
CLIENT	**SNA01,AA**	Uses session 01 of the SNA server loaded in AA.
SES&SNA	**AA,01,31**	SNA session 01 provided by server in AA will use SNA LU number 31.
CLIENT	**SNA02,AA**	Uses session 02 of the SNA server loaded in AA.
SES&SNA	**AA,02,32**	SNA session 02 provided by server in AA will use SNA LU number 32.
SERVER	**EMU32701**	Loads the 3270 emulator, session 1.
PAR&3270	**Y,N**	(Y) means, HLLI will be used. (N) means, cryptographic support will not be used.

Keyword	Parameter Values	
SES&3270	**AA,1,33,TKR**	3270 emulator session 1 uses SNA server through token-ring DLC in AA. The SNA LU number is 33.
	ATR	Is the table default for color display.
	DIS	Is the table default for EBCDIC to ASCII translation.
	KBD	Is the table default for keyboard definitions.
	SESSION1	Is the host application session identification.

This is the vector for workstation **CC:**

```
LWSCONF     NAME=CC,
            TYPE=OS/2,
            CLIENT=(SNA01,AA),
            SES&SNA=(AA,01,11),
            CLIENT=(SNA02,AA),
            SES&SNA=(AA,02,12)
```

The parameter values for this vector are:

Keyword	Parameter Values	
NAME	**CC**	Is the workstation identifier.
TYPE	**OS/2**	Is the operating system (OS/2 Warp V4.0 or later).
CLIENT	**SNA01,AA**	Uses session 01 of the SNA server loaded in AA.
SES&SNA	**AA,01,11**	SNA session 01 provided by server in AA will use SNA LU number 11.
CLIENT	**SNA02,AA**	Uses session 02 of the SNA server loaded in AA.
SES&SNA	**AA,02,12**	SNA session 02 provided by server in AA will use SNA LU number 12.

Step 6. Validating workgroup vectors

You may wish to check whether the workgroup vectors are correct, especially if you have made any modifications to the example file. To call the validation program, from the same subdirectory enter:

```
VALSPEC LAN \SAMPLE1
```

The definitions of the LANCONF.SPC file for SAMPLE1 are validated, with a maximum allowed return code of 0.

If there are errors, attention items, or information messages, the CUSPARM.LST file is created in the same subdirectory. This file contains the vectors and pointers to the messages or errors. Correct the data, and run the validation program again.

When the workgroup data definitions have been edited and validated, you can go on to obtain the diskettes for distribution to your work sites.

Step 7. Obtaining software for distribution

Obtaining software for distribution is a two-stage process:

1. Convert data for one or all workgroups into run-time configuration files for each workstation. Enter the following command:

   ```
   GENRUN LAN \SAMPLE1
   ```

   The run-time files and a file containing a list of software required for each workstation (GETTING.SPC) have now been created. In the example, these files are created in path: C:\LANDP\EHCCUS\SAMPLE1\ws-name.

2. Copy the software for each workstation to diskettes or to any desired path. Enter the following command:

   ```
   GETTING  \SAMPLE1 WS=ALL 1 A:
   ```

   This command copies all the configuration files for all workstations in SAMPLE1 to diskettes in drive A. Make sure you have a formatted disk inserted in drive A.

Step 8. Distributing the software

Each diskette produced in “Step 7. Obtaining software for distribution” contains the files for one workstation in the LANDP workgroup.

To distribute the software to a workstation in the workgroup, you need to take the appropriate diskette to the workstation, then copy the files to the workstation. For example, you might use the following commands:

```
MD C:\EHC
XCOPY A:*.* C:\EHC
```

When the LANDP software is installed, and the necessary changes have been made to the CONFIG.SYS file, you can run LANDP using the AUTOFBSS command.

Appendix B. User servers

User servers are part of the common data, and thus have to be located and specified in the COMMON.SPC file. Two examples of different user server specifications are provided at the end of all the vector descriptions.

User server vectors - descriptions

This section provides information about each user server vector.

DEFSERV vector

Defines a user server and its corresponding characteristics. Define one DEFSERV vector for each server you are going to develop and use.

A quick reference

Vector Position	None
List of keywords	NAME, TYPE, SCOPE, DESCRIP OBJECT, OBJPAR, SUBDIR, LOADER, LOADPAR, PRIORITY, EXPMEM DCESNAME, DCESSYNT, DCESUUID, DCESTRBI, INSTANCE, LANUNIQ, ALLCLI, FIRSTLVL, LASTLVL
Vector relates to	Second level vectors PREVLOAD, POSTLOAD, DEVICE, SVRREQS, and DISTRIB
Vector format	DEFSERV NAME=*xxxxxxxx*, [TYPE=*xxxxxx*,] [SCOPE=*xxxxxx*,] [DESCRIP=*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*,] [OBJECT=*xxxxxxxx.yyy*,] [OBJPAR=' ',] [SUBDIR=*xxxxxxxx*,,] [LOADER=*xxxxxx*,,] [LOADPAR=' ',] [PRIORITY=*x*,] [EXPMEM=*x*,] [DCESNAME=*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*,] [DCESSYNT=*x*,] [DCESUUID=*xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx*,] [DCESTRBI=*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx,*] [INSTANCE=1|N,.] [LANUNIQ=Y|N,] [ALLCLI=Y|N,] [FIRSTLVL=F31|F11|L10|L20|L30|L40|L50] [LASTLVL=F31|F11|L10|L20|L30|L40|L50]

NAME, TYPE, and SCOPE define the user server.

OBJECT and the other keywords up to and including EXPMEM specify parameters related to AUTOFBSS.BAT and AUTOFBSS.CMD.

If the user server will run in a RS/6000 type workstation, and will be used as a DCE server, you can specify the DCESNAME, DCESSYNT, DCESUUID, or DCESTRBI keywords.

Keyword **Description**

NAME Is a required keyword to specify the name of the user server.

The parameter value is a string of up to eight alphanumeric characters, starting with an alphabetical character. It must be unique among the server names.

TYPE Is an optional keyword to specify the operating system of the workstation where the user server will be installed.

The parameter value can be:

`DOS`	For the DOS operating system
`OS/2`	For the OS/2 operating system
`NT`	For the Microsoft Windows NT operating system
`RS/6000`	For the AIX operating system

The default is DOS.

SCOPE Is an optional keyword to define the scope of the user server.

The parameter value can be:

`LOCAL`	The server can have only local clients
`REMOTE`	The server can have only remote clients
`BOTH`	The server can have both local and remote clients

The default is BOTH.

DESCRIP Is an optional keyword to provide a description of the user server.

The parameter value is a string of up to 40 characters. It must be enclosed within quotes.

OBJECT Is an optional keyword to specify the name and extension of the user server executable module.

The parameter has the following format:

```
xxxxxxxx.yyy
```

The name must be the name of the server, which you specified in the NAME keyword, and the extension can be:

EXE
COM
BAT
CMD

The default is the server name specified in the NAME keyword, and the EXE extension.

OBJPAR Is an optional keyword to specify the user server parameters. These parameters will be added after the server name in the corresponding statement of the AUTOFBSS file.

The parameter value is a string of up to 40 characters, and the contents will appear in the loading statement in the same format as specified. It must be enclosed between quotes.

If &WSID appears in the string, it will be converted to workstation identification.

SUBDIR Is an optional keyword to specify the directory where the object module is located. The directory must be located at the same level as the EHCCUS directory.

The parameter value is a string of up to eight characters. The defaults are:

`EHCD000`	For DOS type servers
`EHCO000`	For OS/2 type servers (16 or 32 bits)
`EHCN000`	For Windows NT type servers (32 bits)

If you specify EHCD, EHCO, or EHCN, for DOS type servers, OS/2 type servers, or Windows NT type servers respectively, the customization program converts the values to those corresponding to the current release. For example, to the EHCD500 or EHCO500 values, if you are running LANDP for DOS or LANDP for OS/2.

It is not recommended that you specify these values or use these directories; instead, you should locate the user servers in a separate directory. Thus, the path of the user servers does not depend on the product release.

LOADER Is an optional keyword to specify the loading program that will be used to load the user server.

The parameter value can be:

NO (no loading statement is generated in the AUTOFBSS file)
LOADER
DETACH
START
LSI
CALL

The default is LOADER.

On LANDP for Windows NT:

- LOADER defaults to starting the server as a Windows NT service
- Only LOADER and NO are supported.

On LANDP for DOS, parameter value LOADER will be changed to LOADERE if overridden with other specifications.

LOADPAR Is an optional keyword to specify the parameters for the loading program. This parameter will be added to the loading statement in the AUTOFBSS file, after the name of the loading program.

The parameter value is a string of up to 40 characters. It must be enclosed between quotes.

PRIORITY Is an optional keyword to specify the order in which the loading statements will appear in the AUTOFBSS file.

The parameter value ranges from 1 to 9. The value 1 involves the highest priority; the value 9 involves the lowest priority. The default is 3.

All the loading statements with the same priority will be generated before the loading statements with lower priority. For example, all the statements with priority 1 will be processed before any statement with priority 2 is processed.

EXPMEM Is an optional keyword to specify whether the user server can be loaded into expanded memory. The parameter applies only to DOS type servers.

The parameter value can be Y or N. The default is N.

DCESNAME

LANDP for AIX: Is an optional keyword to specify the name with which the DCE server is known by the DCE Naming Services. The parameter value must match the entry name exported by the server to the Naming Services.

The keyword cannot be specified if you specify the DCESTRBI keyword. Either the DCESNAME keyword or the DCESTRBI keyword must be specified.

The parameter value is a string of up to 60 alphanumeric characters. The special characters ., -, _, /, =, and : are also valid.

To keep lowercase letters after processing, specify the string in quotes. If you omit the quotes, all the letters in the string are set to uppercase letters.

DCESSYNT

LANDP for AIX: Is an optional keyword to specify the syntax of the name with which the DCE server is known by the DCE Naming Services. The keyword is required, if you specify the DCESNAME keyword.

The parameter value can be:

`0` rcp_c_ns_syntax_default
`3` rcp_c_ns_syntax_dce

DCESUUID

LANDP for AIX: Is an optional keyword to specify the object UUID used by the DCE server. The keyword can be specified only if you specify the DCESNAME keyword.

The parameter has the following format:

```
  xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

where *x* represents a hexadecimal digit.

To keep lowercase letters after processing, specify the string in quotes. If you omit the quotes, all the letters in the string are set to uppercase letters.

DCESTRBI

LANDP for AIX: Is an optional keyword to specify the string binding provided by the DCE server.

The keyword cannot be specified if you specify the DCESNAME keyword. Either the DCESNAME keyword or the DCESTRBI keyword must be specified.

The parameter value is a string of up to 80 alphanumeric characters. The special characters ., -, _, @, :, [, and] are also valid.

To keep lowercase letters after processing, specify the string in quotes. If you omit the quotes, all the letters in the string are set to uppercase letters.

INSTANCE

Is an optional keyword to specify whether the user server should be loaded once or once for every session made with the server. The value can be either 1 or N. The value must be 1 for a server loaded as a Windows NT service. The default is **1**.

LANUNIQ Is an optional keyword to specify that only one workstation in the workgroup can have this server. The value can be either Y or N. The default is **N**.

ALLCLI Is an optional keyword to specify that every workstation in the workgroup is automatically a client of this server. The value can be either Y or N. The default is **N**. If you specify Y, you must also specify LANUNIQ=Y.

FIRSTLVL Is an optional keyword that specifies the first level of LANDP that can run the user server.

The parameter value can be:

F31	FBSS Version 3.1
F11	FBSS/2 Version 1.1
L10	LANDP Version 1.0
L20	LANDP Version 2.0
L30	LANDP Version 3.0
L40	LANDP Version 4.0
L50	LANDP Version 5.0

The default is **F11**. If you omit the FIRSTLVL keyword, all levels of LANDP can run the server (unless qualified by the value specified in LASTLVL).

LASTLVL Is an optional keyword that specifies the last level of LANDP that can run the user server.

The parameter value can be:

F31	FBSS Version 3.1
F11	FBSS/2 Version 1.1
L10	LANDP Version 1.0
L20	LANDP Version 2.0
L30	LANDP Version 3.0
L40	LANDP Version 4.0
L50	LANDP Version 5.0

The default is **L50**. If you omit the LASTLVL keyword, all levels of LANDP can run the server (unless qualified by the value specified in FIRSTLVL).

DEVICE vector

Defines a statement to be added to the CONFIG.SYS or CONFIG.ADD file, when the user server is selected for the corresponding workstation.

Define one DEVICE vector for each statement required by the server. The statements will be added in the order you define them.

A quick reference

Vector Position	Follows DEFSERV vector, or any of its second level vectors
List of keywords	UNIQUEID, TEXT
Vector relates to	None
Vector format	DEVICE [UNIQUEID=*xxxxxxxx*,] TEXT='*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*'

Keyword **Description**

UNIQUEID Is an optional keyword to define an identifier that will be used to avoid duplications in the CONFIG.SYS or CONFIG.ADD file, when more than one server requires the same statement. A unique identifier must be assigned to equal statements in different DEVICE vectors, no matter for which server you are defining the statement.

The parameter value is a string of up to eight alphanumeric characters.

TEXT Is a required keyword to define a complete CONFIG.SYS or CONFIG.ADD statement, which will be generated for the workstations with the server.

The parameter value is a string of up to 64 alphanumeric characters, and must be enclosed between quotes.

DISTRIB vector

Specifies a file to be copied to the workstation where the user server is loaded.

Define one DISTRIB vector for each file to be copied. Server executable files and device drivers do not require DISTRIB vectors.

A quick reference

Vector Position	Follows DEFSERV vector, or any of its second level vectors
List of keywords	FILENAME, SUBDIR, UNIQUEID, NEWNAME
Vector relates to	LANCONF vector (LANGUAGE keyword)
Vector format	DISTRIB [FILENAME=*xxxxxxxx.yyy*,] [SUBDIR=*xxxxxxxx*,] [UNIQUEID=*xxxxxxxx*,] [NEWNAME=*xxxxxxxx.yyy*]

Keyword **Description**

FILENAME

Is an optional keyword to define the filename and the extension of the file to be copied.

The parameter has the following format:

xxxxxxxx.yyy

If the alphanumeric field contains the &NLS string, this string will be replaced by the parameter value assigned in the LANGUAGE keyword of the LANCONF vector. You can specify &NLS as the extension, because once it is replaced by the language identifier, it will become a three-character string.

SUBDIR Is an optional keyword to specify the directory where the file is located in the customization workstation. The directory must be located at the same level as the EHCCUS directory.

The parameter value is a string of up to eight characters. The defaults are:

EHCD000	For DOS type servers
EHCO000	For OS/2 type servers (16 or 32 bits)
EHCN000	For Windows NT type servers (32 bits)

UNIQUEID Is an optional keyword to define an identifier that will be used to avoid duplications in the GETTING.SPC file, when more than one server requires the same file. A unique identifier must be assigned to equal files in different DISTRIB vectors, no matter for which server you are specifying the file.

The parameter value is a string of up to eight alphanumeric characters.

NEWNAME

Is an optional keyword to define the filename and the extension of the file, once copied to the target path.

The parameter has the following format:

```
  xxxxxxxx.yyy
```

PARPORT vector

Specifies the parallel ports used by the user server. You can specify up to three ports.

A quick reference

Vector Position	Follows DEFSERV vector, or any of its second level vectors
List of keywords	PORT1, PORT2, PORT3
Vector relates to	None
Vector format	PARPORT PORT1=*xxxx*, [PORT2=*xxxx*,] [PORT3=*xxxx*]

Keyword **Description**

PORT1 Is a required keyword to specify the first LPT*x* used by the server.

The parameter value can be:

LPT1
LPT2
LPT3

PORT2 Is an optional keyword to specify the second LPT*x* used by the server. It cannot be used, if the PORT1 keyword is not specified.

The parameter value can be:

LPT1
LPT2
LPT3

It cannot be the same as the value specified in the PORT1 keyword.

PORT3 Is an optional keyword to specify the third LPT*x* used by the server. It cannot be used, if the PORT1 and PORT2 keywords are not specified.

The parameter value can be:

LPT1
LPT2
LPT3

It cannot be the same as the value specified in the PORT1 or PORT2 keywords.

POSTLOAD vector

Defines statements that will be generated in the AUTOFBSS file, following the user server loading statement.

A quick reference

Vector Position	Follows DEFSERV vector, or any of its second level vectors
List of keywords	UNIQUEID, CONDID, TEXT
Vector relates to	None
Vector format	POSTLOAD [UNIQUEID=*xxxxxxxx*,] [CONDID=*xxxxxxxx*,] TEXT=‘*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*‘

Keyword **Description**

UNIQUEID Is an optional keyword to define an identifier that will be used to avoid duplications in the AUTOFBSS file, when more than one server requires the same statement. A unique identifier must be assigned to equal statements in different POSTLOAD vectors, no matter for which server you are defining the statement.

The parameter value is a string of up to eight alphanumeric characters.

CONDID Is an optional keyword to specify the identifier (UNIQUEID) of a condition to process this vector. If the vector where the UNIQUEID has been specified is included in the AUTOFBSS.xxx file, this vector will also be included.

The parameter value is a string of up to eight alphanumeric characters, and must match a UNIQUEID keyword in another specified vector.

TEXT Is a required keyword to define statements that are generated into the AUTOFBSS file following the server loading statements.

The parameter value is a string of up to 64 alphanumeric characters, and must be enclosed between quotes.

PREVLOAD vector

Defines statements that will be generated in the AUTOFBSS file, before the user server loading statement.

A quick reference

Vector Position	Follows DEFSERV vector, or any of its second level vectors
List of keywords	UNIQUEID, CONDID, TEXT
Vector relates to	None
Vector format	PREVLOAD [UNIQUEID=*xxxxxxxx*,] [CONDID=*xxxxxxxx*,] TEXT=‘*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*‘

Keyword **Description**

UNIQUEID Is an optional keyword to define an identifier that will be used to avoid duplications in the AUTOFBSS file, when more than one server requires the same statement. A unique identifier must be assigned to equal statements in different PREVLOAD vectors, no matter for which server you are defining the statement.

The parameter value is a string of up to eight alphanumeric characters.

CONDID Is an optional keyword to specify the identifier (UNIQUEID) of a condition to process this vector. If the vector where the UNIQUEID has been specified is included in the AUTOFBSS.xxx file, this vector will also be included.

The parameter value is a string of up to eight alphanumeric characters, and must match a UNIQUEID keyword in another specified vector.

TEXT Is a required keyword to define statements that are generated into the AUTOFBSS file following the server loading statements.

The parameter value is a string of up to 64 alphanumeric characters, and must be enclosed between quotes.

SVRREQS vector

Specifies LANDP prerequisite server for the user server.

Define one SVRREQS vector for each prerequisite to be specified.

A quick reference

Vector Position	Follows DEFSERV vector, or any of its second level vectors
List of keywords	COND, TYPE, SVRNAME
Vector relates to	None
Vector format	SVRREQS [COND=x,] [TYPE=x,] SVRNAME=xxxxxxxx

Keyword **Description**

COND Is an optional keyword to define the condition of the requirement.

The parameter value can be:

S The user server is specified in a SERVER keyword
C The user server is specified in a CLIENT keyword

The default is S.

TYPE Is an optional keyword to define the type of the requirement.

The parameter value can be:

S The workstation provides services of the required server
C The required server is specified as a client in the same workstation

The default is C.

SVRNAME Is a required keyword to specify the name of the server.

The parameter value is a string of up to eight alphanumeric characters.

User server examples

```
/* Vector DEFSERV  USERSVR1 server definition */
       DEFSERV  NAME=USERSVR1,
                OBJECT=USERSVR1.EXE,
                OBJPAR='/TBL /&WSID',
                LOADPAR='/w:e /&:e',
                EXPMEM=Y

       DEVICE   UNIQUEID=ID1,
                TEXT='DEVICE=DRIVER1.SYS /a /b /c'

       DEVICE   UNIQUEID=ID2,
                TEXT='DEVICE=DRIVER2.SYS /a /b /c'

       DISTRIB  FILENAME=PGM.EXE,
                UNIQUEID=ID3,
                NEWNAME=USERPGM.EXE

       DISTRIB  FILENAME=FILE002.&NLS,
                UNIQUEID=ID4,
                NEWNAME=FILE.MSG

       PARPORT PORT1=LPT1

       POSTLOAD UNIQUEID=ID5,
                TEXT='echo on Server loaded successfully'

       PREVLOAD UNIQUEID=ID6,
                TEXT='USERPGM.EXE /1'

        SVRREQS  SVRNAME=SNA##

/* Vector DEFSERV. USERSVR2 server definition */
       DEFSERV  NAME=USERSVR2,
                PRIORITY=4,
                OBJECT=USERSVR2.EXE,
                OBJPAR=('/123'),
                LOADPAR='/&:e'

       DEVICE   UNIQUEID=ID1,
                TEXT='DEVICE=DRIVER1.SYS /a /b /c'

       POSTLOAD TEXT='echo on Server loaded successfully'
```

```
PREVLOAD UNIQUEID=ID7,
         CONDID=ID1,
         TEXT='Pause Please, check the printer'

PREVLOAD UNIQUEID=ID8,
         CONDID=ID9,
         TEXT='Pause Please, Power off the printer'
```

For the example server definitions, the CONFIG.SYS file contains the following statements:

```
DEVICE=DRIVER1.SYS /a /b /c
DEVICE=DRIVER2.SYS /a /b /c
```

For the example server definitions, the AUTOFBSS.BAT file contains the following statements (assuming ws = AA)

```
Pause Please check the printer
LOADER /&:e USERSVR2.EXE /123
echo on Server loaded successfully
USERPGM.EXE /1
LOADERE /w:e /&:e USERSVR1.EXE /TBL /AA
echo on Server loaded successfully
```

In addition, the file FILE.MSG will be in the distribution diskette.

Appendix C. Editing common data

OS/2 DOS 6000 NT

This chapter provides information about data that is common for all or for several workstations or workgroups. This is called common data, and is specified and stored in vector format in the COMMON.SPC file, located in the EHCCUS directory.

Common vectors contain defaults, tables, record definitions, and profiles. They also contain definitions for user servers.

Data for user servers is stored and processed together with other common data in the COMMON.SPC file. For information on how to define user servers, refer to Appendix B, "User servers" on page 275.

For information on how to store customization data in specification files, refer to "Customization data structure" on page 36, especially if you plan to use *embedded files* (see page 39) or *partial files* (see page 39).

Creating and editing common vectors

Before you start creating and editing common vectors, consider the following:

1. The customization program provides a procedure to edit specification files.

   You can apply this procedure to the COMMON.SPC file, and display and modify the common vectors.

   If you want to display *online information* about this procedure, from the EHCCUS directory, enter:

   ```
   EDITSPC ?
   ```

   To start the procedure, from the EHCCUS directory, enter:

   ```
   EDITSPC COMMON
   ```

   If you use the OS/2 Enhanced Editor with the LANDP customization editing tool, the **LANDP** choice is listed in the action bar of the editor window. For information about that tool, select the **View doc** option in the pull-down that appears when you choose **LANDP**.

   Note: For all the vectors that have default values, you can specify only the required values in the COMMON.SPC file, and after the VALSPEC and GENSPEC procedures have been performed, the default values will be created by the program.

2. The customization program provides a procedure to create default common data, if it does not exist.

   The default common data is created in the internal repository. If the COMMON.SPC file already exists, the procedure also processes that file.

To start the procedure, from the EHCCUS directory, enter:

```
CREATE
```

3. The customization program provides a procedure to generate common vectors from the common data in the internal repository.

   The common vectors are generated in the COMMON.SPC file, located in the EHCCUS directory. The order in which the vectors appear in the file is not relevant.

   To start the procedure, from the EHCCUS directory, enter:

   ```
   GENSPEC COMMON
   ```

   For further information on that procedure, refer to “Generating data” on page 243.

Common vectors - an overview

Common vectors contain the following information.

User servers are part of common data. They must be created and processed in the COMMON.SPC file. If you want to define and add your own user servers, refer to Appendix B, “User servers” on page 275.

Vector Name	Defines:	Is valid for:	Go to page:
BPPPARM	Optional identifications for printer device parameters	Banking printer program server	290
COLSQTBL	Alternate Collating Sequence Table	Shared-file server	292
DEFAULTS	Hotkeys and session identifications	3270 Emulator, 3287 Printer Emulator, Banking Printer Program	294
DISPLATT	Display attribute tables	3270 Emulator, Banking Interactive Workstation Program	300
EJOUPRF	Electronic journal profile	Electronic journal server	304
EJOUREC	Record names to be used as electronic journal records	Electronic journal server	308
FORM4710	4710 printer device parameters (document and journal formats)	Banking printer program server	309
FORM4720	4720 printer device parameters (document, journal, passbook, and REMS formats)	Banking printer program server	312
FORM47X2	Financial printer device parameters (document, journal, passbook, and REMS formats)	Financial printer server	318
FORM4748	4748 printer device parameters (document and passbook formats)	4748 printer server	329
FORM4770	4770 printer device parameters (document and journal formats)	4770 printer server	337

Vector Name	Defines:	Is valid for:	Go to page:
FORWDS	Forwarding data set inside a forwarding profile	Forwarding server	341
FORWPRF	Forwarding profile	Forwarding server	344
KBD3270	Keyboard table definitions	3270 Emulator	348
KBD3270X	Extended keyboard table definitions	3270 Emulator	354
KBDBIWP	Keyboard table definitions	Banking Interactive Workstation Program	355
KSCCBIWP	Scan code translation table	Banking Interactive Workstation Program	360
KSTRBIWP	Key stroke translation table	Banking Interactive Workstation Program	361
LUPOOL	LUs and LU logical groups	SNA server and functional areas that use SNA services	363
MSRINTBL	MSR/E input conversion table	Banking Interactive Workstation Program for MSR/E server	364
MSROUTBL	MSR/E output conversion table	Banking Interactive Workstation Program for MSR/E server	366
P3287ATT	Time-out and common parameters	3287 printer emulator	368
PINPTBL	PIN pad conversion table	Banking Interactive Workstation Program for PIN pad server	370
RCMSLNF	RCMS logical names file	RCMS server	373
RECDEF	Record definitions	Electronic journal server, Store-for-forwarding server, Shared-file server, System manager server	374
RECFIELD	Field definitions inside a record definition	Electronic journal server, Store-for-forwarding server, Shared-file server, System manager server	377
SFORWPRF	Store-for-forwarding profile	Store-for-forwarding server	379
SFORWREC	Record names to be used as store-for-forwarding records	Store-for-forwarding server	382
SHFLDBD	Shared-file database description (DBD) inside a shared-file profile	Shared-file server	383
SHFLPCB	Program control block (PCB) to access the previously defined DBDs	Shared-file server	388
SHFLSGM	Segmented keys	Shared-file server	389
SMGRPRF	System manager profile	System manager server	392
SMGRUSER	System manager user profile	System manager server	396
SOFTPACK	Non-LANDP files to be distributed with the GETTING utility	Not applicable	399

Vector Name	Defines:	Is valid for:	Go to page:
XLAT2TBL	ASCII/EBCDIC and EBCDIC to ASCII translation tables	Applications using XLATE	402
XLATETBL	Translation tables	3270 emulator, 3287 printer emulator, Banking Interactive Workstation Program, Remote Change Management Services, Forwarding server	400
X25DIR	X.25 directory tables	SNA server	404

Common vectors - descriptions

This section provides information about each common vector, including examples.

The products for which the vectors apply are shown in boxes at the beginning of each vector description. These include PC/Integrator and PC Integrator/2 where FBSS data is involved.

BPPPARM vector

- PC/Integrator
- PC Integrator/2

Defines DEVPARM defaults used by the banking printer program.

This vector is optional. If it is not specified, a default vector will be created. It can only be specified once.

For examples of default (DUMMY*) FORM4710, FORM4720, and FORM47X2 vectors, see the respective vector sections.

A quick reference

Vector position	None
List of keywords	PAR4710, PAR4720, PAR4712, PAR4722, CHSTRID
Keywords relate to	**Keyword** — **Relates to keywords** **PAR4710** NAME (FORM4710 vector) **PAR4720** NAME (FORM4720 vector) **PAR4712** NAME (FORM47X2 vector) **PAR4722** NAME (FORM47X2 vector) **CHSTRID** EXTEN (XLATETBL vector)

Vector format	BPPPARM [PAR4710=*xxxxxxxx*,] [PAR4720=*xxxxxxxx*,] [PAR4712=*xxxxxxxx*,] [PAR4722=*xxxxxxxx*,] [CHSTRID=( ),] [CHSTRID=( )]

Keyword **Description**

PAR4710 Is an optional keyword to define the identification for the default DEVPARM when emulating the IBM 4710 Receipt/Validation Printer.

The parameter value is string of up to eight alphanumeric characters. The default is **DUMMYDOC**.

PAR4720 Is an optional keyword to define the identification for the default DEVPARM when emulating the IBM 4720 Forms/Passbook Printer.

The parameter value is a string of up to eight alphanumeric characters. The default is **DUMMYDOC**.

PAR4712 Is an optional keyword to define the identification for the default DEVPARM when using the financial printer.

The parameter value is a string of up to eight alphanumeric characters. The default is **DUMMYJOU**.

PAR4722 Is an optional keyword to define the identification for the default DEVPARM when using the financial printer.

The parameter value is a string of up to eight alphanumeric characters. The default is **DUMMYDOC**.

CHSTRID Is an optional keyword to associate the EBCDIC to ASCII translation tables used by the BPP with the 1-byte value that is used in the 4700 application to change translation tables.

You can specify one keyword for each character string table you want to define.

The keyword has two parameters:

1. The first parameter is the identification of the EBCDIC to ASCII table, and must match the keyword EXTEN of the XLATETBL vector (TYPE=EA47X2).
2. The second parameter specifies the 1-byte value that is used to change translation tables via the 4700 DEVPARM instructions with flag byte 20. It must be unique for this vector.

BPPPARM vector example

```
/* Vector BPPPARM  Default Example */

        BPPPARM  PAR4710=DUMMYDOC,
                 PAR4720=DUMMYDOC,
                 PAR4712=DUMMYJOU,
                 PAR4722=DUMMYDOC,
                 CHSTRID=(017,11)
```

COLSQTBL vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- LANDP for AIX

Defines an alternate collating sequence table for the shared-file server.

Define one COLSQTBL vector for each alternate collating sequence table you want to use in your workgroups. If this vector is not specified, a default vector will be created.

A quick reference

Vector position	None
List of keywords	NAME, SUBEXT, DATA0X to DATAFX
Keywords relate to	**Keyword** **Relates to keywords** **NAME** SHFLPRF (SHFLDBD vector)

Vector format	COLSQTBL NAME=*xxxxxxxx*,
	[SUBEXT=*x*,]
	DATA0X=( ),
	DATA1X=( ),
	:
	DATAFX=( )

Keyword **Description**

NAME Is a required keyword to define the name of the alternate collating sequence table. It must match the name of the shared-file profile (set of SHFLDBDs) that will use this table. This name is defined in the SHFLPRF keyword of the SHFLDBD vector.

The parameter value is a string of up to eight characters, which must adhere to DOS specifications for file names. It must be unique among all COLSQTBL vectors.

SUBEXT Is an optional keyword to specify the third character of the extension of the alternate collating sequence table. It must match the value specified in the third parameter of the key definitions (keywords KEYxx) of the shared-file profile (set of SHFLDBDs) that will use this table.

The parameter value can be any hexadecimal character, except for 0. The default is **Q**.

The extension becomes SE*x*, where *x* is the specified value. Thus, the default extension is SEQ.

When working under OS/2, only the alternate collating sequence table with extension SEQ can be substituted by the Default Country Collating Table.

DATA0X to DATAFX

Are required keywords to define the table contents. The parameter value is a string of 16 parameters, each one with 1-byte values from '00' to 'FF'.

DATA0X must contain the alternate collating sequence characters corresponding to ASCII characters '00' to '0F', DATA1X to ASCII '10' to '1F', and so on.

COLSQTBL vector example

```
/* Vector COLSQTBL  Default Example */

  COLSQTBL NAME=CONFIGUR,
           SUBEXT=1,
           DATA0X=(00,01,02,03,04,05,06,07,08,09,0A,0B,0C,0D,0E,0F),
           DATA1X=(10,11,12,13,14,15,16,17,18,19,1A,1B,1C,1D,1E,1F),
           DATA2X=(20,21,22,23,24,25,26,27,28,29,2A,2B,2C,2D,2E,2F),
           DATA3X=(30,31,32,33,34,35,36,37,38,39,3A,3B,3C,3D,3E,3F),
           DATA4X=(40,41,42,43,44,45,46,47,48,49,4A,4B,4C,4D,4E,4F),
           DATA5X=(50,51,52,53,54,55,56,57,58,59,5A,5B,5C,5D,5E,5F),
           DATA6X=(60,61,62,63,64,65,66,67,68,69,6A,6B,6C,6D,6E,6F),
           DATA7X=(70,71,72,73,74,75,76,77,78,79,7A,7B,7C,7D,7E,7F),
           DATA8X=(80,81,82,83,84,85,86,87,88,89,8A,8B,8C,8D,8E,8F),
           DATA9X=(90,91,92,93,94,95,96,97,98,99,9A,9B,9C,9D,9E,9F),
           DATAAX=(A0,A1,A2,A3,A4,A5,A6,A7,A8,A9,AA,AB,AC,AD,AE,AF),
           DATABX=(B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,BA,BB,BC,BD,BE,BF),
           DATACX=(C0,C1,C2,C3,C4,C5,C6,C7,C8,C9,CA,CB,CC,CD,CE,CF),
           DATADX=(D0,D1,D2,D3,D4,D5,D6,D7,D8,D9,DA,DB,DC,DD,DE,DF),
           DATAEX=(E0,E1,E2,E3,E4,E5,E6,E7,E8,E9,EA,EB,EC,ED,EE,EF),
           DATAFX=(F0,F1,F2,F3,F4,F5,F6,F7,F8,F9,FA,FB,FC,FD,FE,FF)
```

DEFAULTS vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- PC/Integrator
- PC Integrator/2

Defines required parameters for the supervisor and emulators.

Define as many DEFAULTS vectors as required in your installation. If this vector is omitted, a default vector will be created.

The DEFAULTS vectors will be referenced on workgroup level in the keyword DEFAULTS of the LANMODEL or LANCONF vectors.

The parameters that you define in the DEFAULTS vector are:

- Scan code and ASCII code to be returned to client when timer 0 (keyboard time-out) expires.

  There is a help facility for calling the scan and ASCII codes for each key on your keyboard. From the EHCCUS subdirectory, enter:

  ```
  EHCSC
  ```

  and you will get an interactive help panel where you can enter any key to find out the specified values.

- LAN timeout value.
- Host application identifications for the terminal emulators.
- Hotkeys to activate the LANDP for DOS trace tool, Operator Interface, System manager operator, and terminal emulators.

  For the hotkeys you define the scan code of the key that, when pressed with the Alt key, will transfer control to the corresponding LANDP for DOS server. The following scan codes *cannot* be chosen:

38	Alt
1D	Ctrl
2A	Shift left
36	Shift right
37	PrtSc Print screen
52	Ins
53	Del

  If you define a scan code of 00, the corresponding hotkey will not be defined.

A quick reference

Vector position	None

List of keywords	NAME, DEBUGHK, OPERHK, SMOPHK, TMR0SCC, TMR0ASC, TIMEOUT, E3270HK1 to E3270HK5, E3287SI1 to E3287SI5, BPPSI1 to BPPSI4, DBCSCTRY, DBCSPATH, PROCNAME
Keywords relate to	**Keyword** **Relates to keywords** **NAME** DEFAULTS (LANCONF and LANMODEL vectors) **E3270HKx** SES&3270 (parameter 8) **BPPSIx** SES&BPP (parameter 7) **E3287SIx** SES&3287 (parameter 9)
Vector format	DEFAULTS [NAME=*xxxxxxxx*,] [DEBUGHK=*xx*,] [OPERHK=*xx*,] [SMOPHK=*xx*,] [TMR0SCC=*xx*,] [TMR0ASC=*xx*,] [TIMEOUT=*xxxx*,] [E3270HK1=( ),] [E3270HK2=( ),] [E3270HK3=( ),] [E3270HK4=( ),] [E3270HK5=( ),] [E3287SI1=*xxxxxxxx*,] [E3287SI2=*xxxxxxxx*,] [E3287SI3=*xxxxxxxx*,] [E3287SI4=*xxxxxxxx*,] [E3287SI5=*xxxxxxxx*,] [BPPSI1=*xxxxxxxx*,] [BPPSI2=*xxxxxxxx*,] [BPPSI3=*xxxxxxxx*,] [BPPSI4=*xxxxxxxx*,] [DBCSCTRY=( ),] [DBCSPATH=*path*,] [PROCNAME=*xxxxxxxx*]

Keyword **Description**

NAME Is an optional keyword to define the name of the set of defaults.

The parameter value is a field with up to eight alphanumeric characters, and must be unique among the DEFAULTS vectors defined. The default is **GENERAL**.

DEBUGHK

LANDP for DOS: Is an optional keyword to define the scan code for the hotkey that, when pressed with the Alt key, activates the trace and debug tool.

The parameter value is a 1-byte field. The default is **45** (NumLock).

OPERHK LANDP for DOS: Is an optional keyword to define the scan code for the hotkey that, when pressed with the Alt key, activates the operator interface.

The parameter value is a 1-byte field. The default is **44** (F10).

SMOPHK LANDP for DOS: Is an optional keyword to define the scan code for the hotkey that, when pressed with the Alt key, activates the system manager operator.

The parameter value is a 1-byte field. The default is **3E** (F4).

TMR0SCC LANDP for DOS: Is an optional keyword to define the scan code sent to the application when timer 0 (keyboard time-out), elapses.

LANDP for OS/2: The keyword can be set to any extended ASCII character.

LANDP for Windows NT: The keyword can be set to the extended ASCII characters Alt F1-F10 and Ctrl F1-F10.

The parameter value is a 1-byte field. The default is **01** (Esc), which is returned when timer 0 elapses if TMR0SCC is not coded or is set to a value other than those specified above.

TMR0ASC Is an optional keyword to define the ASCII code sent to the application when timer 0 (keyboard time-out), elapses.

The parameter value is a 1-byte field. The default is **1B** (Esc).

TIMEOUT Is an optional keyword to define the time (in seconds) after which the supervisor returns a time-out return code when a LAN call has not been answered.

The parameter value is a field with up to four digits, and can range from 1 to 3600. The default is **30** seconds.

E3270HK1 to E3270HK5

DOS, OS/2 MVDM, Windows NT MVDM: Are optional keywords to define the scan codes for the keys that, when pressed with the Alt key, activate the corresponding 3270 emulator program. It also defines its host/application identification.

A maximum of five 3270 emulator programs can be installed in one workstation.

Each keyword has two parameters in the format (*xx*,*yyyyyyyy*), where:

xx Are two hexadecimal digits for the scan code used

yyyyyyyy Are up to eight alphanumeric digits for the host/application identification.

By default, the host/application identification is blank.

The defaults for the respective scan codes (examples, page 298) are:

E3270HK1	3F (F5)
E3270HK2	40 (F6)
E3270HK3	41 (F7)
E3270HK4	42 (F8)
E3270HK5	**43** (F9)

Multiple emulators

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

To chain four emulators so that the next emulator is activated when the preceding emulator is ended, ensure that the hotkey to enter an emulator is the same as at least one of the the hotkeys used to exit the emulator. Exit hotkeys are defined in the KBD3270 vector.

Table 7. Chained Emulators: Example 1

Emulator	**Entry Hotkey**	**Exit Hotkeys**
1	ALT+F2	Alt+F2, Alt+F10
2	ALT+F2	Alt+F2, Alt+F10
3	ALT+F2	Alt+F2, Alt+F10
4	ALT+F2	Alt+F2, Alt+F10

The first time ALT+F2 is pressed, Emulator 1 is activated:

- If ALT+F2 is pressed again, Emulator 2 is activated
- If ALT+F2 is pressed again, Emulator 3 is activated
- If ALT+F2 is pressed again, Emulator 4 is activated
- If ALT+F2 is pressed again, Emulator 1 is activated

Pressing ALT+F10 exits the emulation mode.

Notice that if ALT+F2 is pressed when the workstation is not in emulation mode, Emulator 1 is always activated. Using this configuration, it is not possible to directly activate any emulator other than Emulator 1 when the workstation is not in emulation mode.

To be able to activate a specific emulator, define a unique entry hotkey for each emulator.

Table 8. Chained Emulators: Example 2

Emulator	**Entry Hotkey**	**Exit Hotkeys**
1	ALT+F2	Alt+F3, Alt+F4, Alt+F5, Alt+F10
2	ALT+F3	Alt+F2, Alt+F4, Alt+F5, Alt+F10
3	ALT+F4	Alt+F2, Alt+F3, Alt+F5, Alt+F10
4	ALT+F5	Alt+F2, Alt+F3, Alt+F4, Alt+F10

Whether the workstation is in emulation mode or not:

- If ALT+F2 is pressed, Emulator 1 is activated
- If ALT+F3 is pressed, Emulator 2 is activated
- If ALT+F4 is pressed, Emulator 3 is activated
- If ALT+F5 is pressed, Emulator 4 is activated

Pressing ALT+F10 exits the emulation mode.

The scan codes for the E3270HK1-E3270HK5 keywords must be those returned by INT15H. The values for the keys can be found in *Technical Reference* for your workstation. Examples of correct values for the keywords are:

3B (F1)	44 (F10)
3C (F2)	57 (F11)
3D (F3)	58 (F12)
3E (F4)	01 (Esc)

E3287SI1 to E3287SI5

LANDP for DOS, OS/2, and Windows NT: Are optional keywords to define the host/application identification for 3287 printer emulator sessions.

A maximum of five 3287 printer emulator sessions can be installed in one workstation. Define the sessions you need in any order.

The parameter value is a field of up to eight alphanumeric characters. By default, all five host/application identifications are blank.

BPPSI1 to BPPSI4

PC/Integrator and PC Integrator/2: Is an optional keyword to define the host/application identification for banking printer program ports 1 to 4.

A maximum of four banking printer programs can be installed in one workstation for LANDP for DOS, and three for LANDP for OS/2. Define the sessions you need in any order.

The parameter value is a field of up to eight alphanumeric characters. By default, all four programs are blank.

DBCSCTRY

Is an optional keyword to specify the identifier of the language to be used when working in DBCS mode. You can also specify that DBCS support is not used.

The parameter value can be:

082 Korea (not supported on LANDP for Windows NT)
086 People's Republic of China
088 Taiwan
NO DBCS support not used

The default is **NO**.

The keyword has two parameters in the format (*xxx*,*yyyy*), where:

xxx Is the identifier of the language to be used.

yyyy Is the number of the code page to be used for that country.

If Korea (082) is specified, a second sub-parameter must also be specified designating the code page used in the LANDP workgroup. The supported code page is 949.

If People's Republic of China (086) is specified, a second sub-parameter must also be specified designating the code page used in the LANDP workgroup. The supported code pages are 1381 and 1386.

If Taiwan (088) is specified, a second sub-parameter must also be specified. In this case, the supported code pages are 938 and 950. For example to specify Taiwan as the DBCSCTRY using code page 950, you would specify:

```
DBCSCTRY=(088,950)
```

Additionally, for People's Republic of China (086), a third sub-parameter may be specified designating the code page of the host computer. The supported code pages are 935 and 1388. For example, to specify PRC as the DBCSCTRY using LANDP workgroup code page 1386 and host code page 1388, you would specify

```
DBCSCTRY=(086,1386,1388)
```

DBCSPATH

Is an optional keyword to specify the path where the TBLASC.088, TBLEBC.088, and TBLTAI.088 files are located. It applies only when working with the Taiwan DBCS code pages 938 or 950, (DBCSCTRY=(088,938)) or (DBCSCTRY=(088,950)).

The parameter value is a string of up to 30 alphanumeric characters. The format must be:

```
d:\[directory1\[directory2\[directory3\]]]
```

A maximum of three levels is permitted for the path. The default is **C:\**.

PROCNAME

Is an optional keyword to specify the name of the procedure to start the LANDP program.

The parameter value is a field with up to eight alphanumeric characters. The default is **AUTOFBSS**.

DEFAULTS vector example

```
/* Vector DEFAULTS  Default Example */

         DEFAULTS NAME=GENERAL,
                  DEBUGHK=45,
                  OPERHK=44,
                  SMOPHK=3E,
                  TMR0SCC=01,
                  TMR0ASC=1B,
                  TIMEOUT=30,
                  E3270HK1=(3F),
                  E3270HK2=(40),
                  E3270HK3=(41),
                  E3270HK4=(42),
                  E3270HK5=(43)
```

DISPLATT vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- PC/Integrator
- PC Integrator/2

Defines color display attribute tables for the 3270 emulator or the banking interactive workstation program. These are the colors that you want to appear on your workstation display for each field attribute. You may want to assign a different color set for each 3270 or banking interactive workstation program in a workstation.

Specify one vector for each table you want to use in your workgroups.

Table 9 shows the display colors that can be used.

Table 9. Display Colors

Color code	**Color**
0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	Light Grey
8	Dark Grey
9	Light Blue
A	Light Green
B	Light Cyan
C	Light Red
D	Light Magenta
E	Yellow
F	White

A quick reference

Vector position	None
List of keywords	TYPE, EXTEN, DEFCOLRS, BACKGRD, HIGHPROT, NORMPROT, HIGHUNPR, NORMUNPR, STATUS, BLUE, RED, PINK, GREEN, TURQUOIS, YELLOW, WHITE, HIGHDFLT, NORMDFLT

Keywords relate to	**Keyword** **Relates to keywords** **EXTEN** SES&3270 (parameter 5), PAR&BIWP (parameter 13)
Vector format	DISPLATT TYPE=*xxxx*, EXTEN=*xxx*, [DEFCOLRS=*xxxxx*,] [BACKGRD=*x*,] [HIGHPROT=*x*,] [NORMPROT=*x*,] [HIGHUNPR=*x*,] [NORMUNPR=*x*,] [BLUE=*x*,] [RED=*x*,] [PINK=*x*,] [GREEN=*x*,] [TURQUOIS=*x*,] [YELLOW=*x*,] [WHITE=*x*,] [HIGHDFLT=*x*,] [NORMDFLT=*x*,] [STATUS=*xx*]

Keyword **Description**

TYPE Is a required keyword to define the table type. The parameter values are:

3270 For a 3270 color display attribute table
BIWP For a BIWP color display attribute table

EXTEN Is a required keyword to give an identification for a 3270 or BIWP display attribute table.

The parameter value is a 3-character alphanumeric string, and may contain the special characters #, $, %, and @. It must be unique among all DISPLATT vectors.

DEFCOLRS
Is an optional keyword to define the default color scheme for all the keywords that follow. The keyword applies to TYPE=3270 only. The parameter values are:

MONO
Use monochrome colors as the default color scheme. This is the default.

DARK
Use dark colors as the default color scheme. (This is the suggested color scheme for blinking text.)

LIGHT
Use light colors as the default color scheme. (This is the suggested color scheme for non-blinking text.)

The attribute values for the three default color schemes are as follows:

Keyword	DEFCOLRS=MONO	DEFCOLRS=DARK	DEFCOLRS=LIGHT
BACKGRD	0 (Black)	0 (Black)	0 (Black)
HIGHPROT	F (White)	7 (Light grey)	F (White)
NORMPROT	7 (Light grey)	1 (Dark blue)	9 (Light blue)
HIGHUNPR	F (White)	4 (Dark red)	C (Light red)
NORMUNPR	7 (Light grey)	2 (Dark green)	A (Light green)
STATUS (background)	7 (Light grey)	2 (Dark green)	A (Light green)
STATUS (foreground)	0 (Black)	0 (Black)	0 (Black)
BLUE	7 (Light grey)	1 (Dark blue)	9 (Light blue)
RED	7 (Light grey)	4 (Dark red)	C (Light red)
PINK	F (White)	5 (Dark magenta)	D (Light magenta)
GREEN	7 (Light grey)	2 (Dark green)	A (Light green)
TURQUOIS	F (White)	3 (Dark cyan)	B (Light cyan)
YELLOW	F (White)	6 (Brown)	E (Yellow)
WHITE	F (White)	7 (Light grey)	F (White)
HIGHDFLT	F (White)	7 (Light grey)	F (White)
NORMDFLT	7 (Light grey)	2 (Dark green)	A (Light green)

Note: To maintain upward compatibility with older versions of LANDP, the default color scheme for DISPLATT is MONO, which assigns the colors white or light grey to the extended color DISPLATT keywords.

If you have changed your BACKGRD color to white or light grey, you will receive message EHC02204 to tell you that the background and foreground colors for the attribute you have chosen should not be the same (because that data will not be visible). You can ignore this message if you do not intend to use extended color on this display.

If you want to use extended color, you should specify DEFCOLRS=DARK or DEFCOLRS=LIGHT and then specify any variations from the default scheme that you want.

If you do not want to use extended color, you might want to do this anyway, to suppress the messages.

BACKGRD

Is an optional keyword to define the background color attribute. This parameter will be common for HIGHPROT, NORMPROT, HIGHUNPR, NORMUNPR, DEFCOLRS, BLUE, RED, PINK, GREEN, TURQUOIS, YELLOW, WHITE, HIGHDFLT, and NORMDFLT fields.

The parameter value is a 1-hexadecimal digit. The default is the value derived from the DEFCOLRS keyword.

HIGHPROT

Is an optional keyword to define the foreground color attribute for base color high intensity protected fields.

The parameter value is a 1-hexadecimal digit. The default is the value derived from the DEFCOLRS keyword.

NORMPROT
Is an optional keyword to define the attribute for base color normal intensity protected fields.

The parameter value is a 1-hexadecimal digit. The default is the value derived from the DEFCOLRS keyword.

HIGHUNPR
Is an optional keyword to define the attribute for base color high intensity unprotected fields.

The parameter value is a 1-hexadecimal digit. The default is the value derived from the DEFCOLRS keyword.

NORMUNPR
Is an optional keyword to define the attribute for base color normal intensity unprotected fields.

The parameter value is a 1-hexadecimal digit. The default is the value derived from the DEFCOLRS keyword.

BLUE Is an optional keyword to define the attribute for extended color blue characters and fields. The keyword applies to TYPE=3270 only.

The parameter value is a 1-hexadecimal digit. The default is the value derived from the DEFCOLRS keyword.

RED Is an optional keyword to define the attribute for extended color red characters and fields. The keyword applies to TYPE=3270 only.

The parameter value is a 1-hexadecimal digit. The default is the value derived from the DEFCOLRS keyword.

PINK Is an optional keyword to define the attribute for extended color pink characters and fields. The keyword applies to TYPE=3270 only.

The parameter value is a 1-hexadecimal digit. The default is the value derived from the DEFCOLRS keyword.

GREEN Is an optional keyword to define the attribute for extended color green characters and fields. The keyword applies to TYPE=3270 only.

The parameter value is a 1-hexadecimal digit. The default is the value derived from the DEFCOLRS keyword.

TURQUOIS
Is an optional keyword to define the attribute for extended color turquoise characters and fields. The keyword applies to TYPE=3270 only.

The parameter value is a 1-hexadecimal digit. The default is the value derived from the DEFCOLRS keyword.

YELLOW Is an optional keyword to define the attribute for extended color yellow characters and fields. The keyword applies to TYPE=3270 only.

The parameter value is a 1-hexadecimal digit. The default is the value derived from the DEFCOLRS keyword.

WHITE Is an optional keyword to define the attribute for extended color white characters and fields. The keyword applies to TYPE=3270 only.

The parameter value is a 1-hexadecimal digit. The default is the value derived from the DEFCOLRS keyword.

HIGHDFLT
Is an optional keyword to define the attribute for extended color default high intensity characters and fields. The keyword applies to TYPE=3270 only.

The parameter value is a 1-hexadecimal digit. The default is the value derived from the DEFCOLRS keyword.

NORMDFLT
Is an optional keyword to define the attribute for extended color default (normal intensity) characters and fields. The keyword applies to TYPE=3270 only.

The parameter value is a 1-hexadecimal digit. The default is the value derived from the DEFCOLRS keyword.

STATUS Is an optional keyword to define the attributes for the status symbol line.

The parameter is a value of two digits, with the first one specifying the background, and the second one the foreground color.

The parameter value is a 2-hexadecimal digit. The default is the values derived from the DEFCOLRS keyword.

The keyword does not apply to DBCS mode. When working in DBCS mode, the value used by the 3270 emulator is 07, which is black foreground on a light grey background.

DISPLATT vector example

```
/* Vector DISPLATT (TYPE=3270):  Example */

         DISPLATT TYPE=3270,
                  EXTEN=ATR,
                  DEFCOLRS=LIGHT,
                  BLUE=3            /* Change blue to dark cyan */
```

EJOUPRF vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT

- LANDP for AIX

Defines an electronic journal server profile. Define one EJOUPRF vector for each electronic journal profile you want to specify.

When you define a DOS, OS/2, or Windows NT workstation, this vector must be followed by as many EJOUREC vectors as different records will be used in this electronic journal profile.

When you define a RS/6000 workstation, the records are defined using a record definition utility program, shown in *LANDP Servers and System Management*.

You can choose among two electronic journal server environments: physical or logical. In the physical environment, each file is accessed through a physical journal name that is assigned by the application. In the logical environment, you can have up to 99 logical journals that the application can access by name within each physical journal.

A quick reference

Vector position	None. Is followed by at least one EJOUREC vector.
List of keywords	NAME, DATASETS, MAXNUML, SEPSESS, MAXACC, SHFLPRF, DBDPATH, SPLIT, KEY02 to KEY15
Keywords relate to	**Keyword** **Relates to keywords** **NAME** PAR&EJOU (parameter 1) **SHFLPRF** SHFLPRF (SHFLDBD vector) **KEYxx** NAME (RECFIELD vector)
Vector format	EJOUPRF NAME=xxxxxxxx, [DATASETS=*xx*,] [MAXNUML=*xx*,] [SEPSESS=*x*,] [MAXACC=*xxxxx*,] SHFLPRF=*xxxxxxxx*, DBDPATH=*path*, DBDPATH6=*path*, SPLIT=*length*, [KEY02=*xxxxxxxx*,] : [KEY15=*xxxxxxxx*]

Keyword **Description**

NAME Is a required keyword to define the name of the electronic journal profile.

The parameter value is a field of up to eight alphanumeric characters plus the special characters $, %, #, and @. It must be unique among all EJOUPRF vectors.

DATASETS

Is an optional keyword to define the number of data sets (files) to be used as physical electronic journals.

The parameter value is a field with up to two numeric characters, ranging from 1 to 70. The default is **3**.

MAXNUML

LANDP for DOS, OS/2, and Windows NT: Is an optional keyword to define the number of logical electronic journals to be used.

The parameter value is a field with up to two numeric characters, ranging from 0 to 99. The default is **0**. When you select 0, only the physical journal environment is used.

SEPSESS LANDP for DOS, OS/2, and Windows NT: Is an optional keyword to define a method for providing integrity for data access to the electronic journal server records.

Specify Y (Yes) if you want session integrity to be handled by the electronic journal server, independently of your own shared-file session. The default is **N** (No).

MAXACC LANDP for DOS, OS/2, and Windows NT: Is an optional keyword to define the maximum number of accesses in search operations. This is the maximum number of searches made before the search operation is ended and control is returned to the application.

The parameter value is a field with up to five numeric characters and it can range from 1 to 32767. The default is **200**.

SHFLPRF LANDP for DOS, OS/2, and Windows NT: Is a required keyword to define the name of the shared-file profile to be used.

You can use the name of a shared-file profile already defined in the keyword SHFLPRF of the SHFLDBD vector, or a new name. If you provide a new profile name, the corresponding shared-file profile will be created by the customization program.

In any case, the customization will add the DBD and PCB definitions required for this electronic journal profile to the shared-file profile.

The same shared-file profile can only be used by one electronic journal profile.

If you use the electronic journal server and the store-for-forwarding server in the same workstation, you must use the same shared-file profile for both of them.

The parameter value is a field with up to eight alphanumeric characters. It must be unique among the EJOUPRF vectors.

DBDPATH LANDP for DOS, OS/2, and Windows NT: Is a required keyword to define the path where the electronic journal data and index files will be located. This path must exist prior to running the LANDP family programs in a workstation with the shared-file server services.

The parameter value is the complete path (drive plus subdirectories). The number of subdirectories must not exceed four, and has to end with a back slash (\). It must at least contain three characters.

SPLIT LANDP for DOS, OS/2, and Windows NT: Is a required keyword to define the length of the user record that can be kept in one electronic journal record. If the record size exceeds the number specified, it will be split into two or more segments.

Parameter values range from 62 to 4096 minus the size of the header that is appended. The header is the length of all the defined keys (including the electronic journal hidden key EJHIDKEY which is 8 bytes long) plus 36.

KEY02 to KEY15
LANDP for DOS, OS/2, and Windows NT: Are optional parameters to define the names of the fields used as keys. These field names must be defined in at least one of the records specified in the EJOUREC vectors that follow this vector.

Fields in a record can be selected as electronic journal keys only if they are in character, unsigned ASCII numeric, or hexadecimal, and their length is fixed and less than 50 bytes.

If fields specified as keys are defined in more than one record, the characteristics of the field must match.

The keys must be specified sequentially; for example, KEY05 cannot be specified if KEY04 was omitted. The first key (KEY01) is not defined because it is the electronic journal hidden key.

EJOUPRF vector example

```
/* Vectors EJOUPRF and EJOUREC Examples */

         EJOUPRF  NAME=EJPRF01,
                  DATASETS=10,
                  MAXNUML=5,
                  SEPSESS=N,
                  MAXACC=200,
                  SHFLPRF=SFPROF01,
                  DBDPATH=C:\TELLER\EJOU\,
                  SPLIT=1024,
                  KEY02=REC1FL01,
                  KEY03=REC2FL03,
                  KEY04=REC1FL02,
                  KEY05=REC2FL01

         EJOUREC  RECNAME=RECORD01
```

EJOUREC vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT

Defines the names of the record structures defined with the RECDEF vector which will be used as electronic journal records.

At least one EJOUREC vector must be specified following an EJOUPRF vector.

A quick reference

Vector position	Follows EJOUPRF vector
List of keywords	RECNAME
Keywords relate to	**Keyword** **Relates to keywords** **RECNAME** NAME (RECDEF vector)
Vector format	EJOUREC RECNAME=*xxxxxxxx*

Keyword **Description**

RECNAME

Is a required keyword to specify the name of a record defined with the RECDEF vector which will be used as an electronic journal record.

The parameter value is a field with up to eight alphanumeric characters, and must match the name given in keyword NAME of a RECDEF vector. Records that have fields with names AND or OR cannot be used as electronic journal records.

EJOUREC vector example

```
/* Vectors EJOUPRF and EJOUREC Examples */

         EJOUPRF  NAME=EJPRF01,
                  DATASETS=10,
                  MAXNUML=5,
                  SEPSESS=N,
                  MAXACC=200,
                  SHFLPRF=SFPROF01,
                  DBDPATH=C:\TELLER\EJOU\,
                  SPLIT=1024,
                  KEY02=REC1FL01,
                  KEY03=REC2FL03,
                  KEY04=REC1FL02,
                  KEY05=REC2FL01

         EJOUREC  RECNAME=RECORD01
```

FORM4710 vector

- PC/Integrator
- PC Integrator/2

Defines default parameters (DEVPARM) for banking printer program when emulating a 4710 printer.

The following figure shows the layout of the document pages for a 4710 printer:

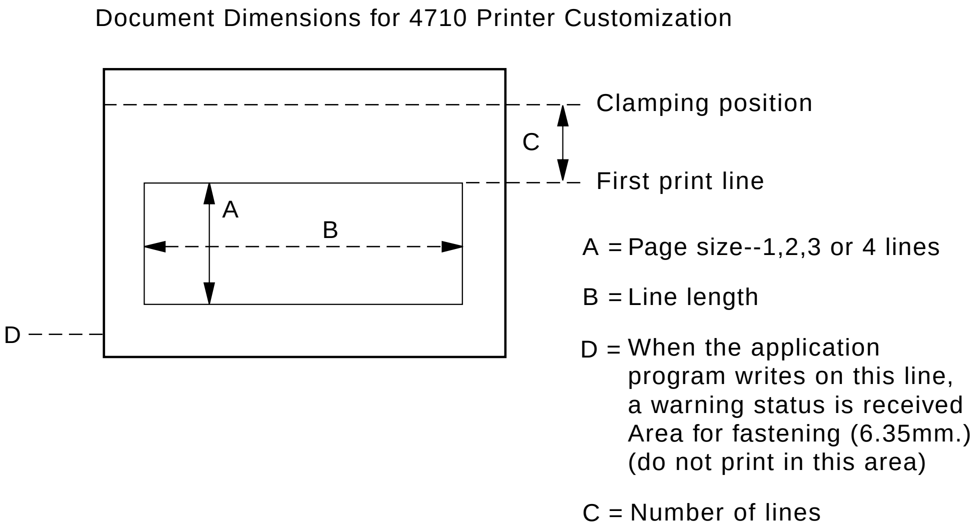

Document Dimensions for 4710 Printer Customization
Clamping position
C
First print line
A
B
D
A = Page size--1,2,3 or 4 lines
B = Line length
D = When the application program writes on this line, a warning status is received Area for fastening (6.35mm.) (do not print in this area)
C = Number of lines

Figure 9. Document dimensions for IBM 4710 printer customization

A quick reference

Vector position	None
List of keywords	See keyword TYPE
Keywords relate to	**Keyword** **Relates to keywords** **NAME** PAR4710 (BPPPARM vector)
Vector format	FORM4710 TYPE=*xxxx*, NAME=*xxxxxxxx*, [AUTONL=*x*,] [CPI=*xx*,] [LINELEN=*xx*,] [VRTOFFSL=*x*,] [PAGESIZE=*x*,] [STARTKEY=*x*,] [WRNLINE=*x*]

Keyword **Description**

TYPE Is a required keyword to define the type of object. The type can be DOCU for document, or JOUR for journal.

Depending on the type you selected, you define a different set of keywords:

DOCU NAME (required), CPI, LINELEN, PAGESIZE, WRNLINE, VRTOFFSL, AUTONL, STARTKEY

JOUR NAME (required), CPI, LINELEN, PAGESIZE, WRNLINE

NAME Is a required keyword to define the unique name of the form parameters you specify. The name can have a maximum of eight alphanumeric characters, A to Z and 0 to 9, plus the special characters $, %, #, and @.

AUTONL Is an optional keyword to define automatic skip to next line. The values are:

Y (Yes) The printer automatically goes to the next line when the text exceeds the last position on the current line.

N (No) An error message will be issued when the line length is exceeded.

The default is **Y**.

CPI Is an optional keyword to define printing density (number of characters per inch). Allowed values are 10 or 12. The default is **10**.

LINELEN Is an optional keyword to define the maximum number of characters being written in one line. The values depend on the value defined for CPI: (Dimension B)

- 40 characters for printing density 10
- 48 characters for printing density 12.

The default is the maximum value for the CPI specified.

VRTOFFSL

Is an optional keyword to define the number of lines between the first printable line and the line occupied by the tractor clamping mechanism. (See dimension C in Figure 9 on page 309.)

The value specified for VRTOFFSL plus the one for PAGESIZE must not be greater than 4. If WRNLINE is greater than 0, VRTOFFSL must not be greater than WRNLINE. The default is **0**.

PAGESIZE

Is an optional keyword to define the page size. (Dimension A)

- For type JOUR:

  Permits the journal to be split up in logical pages by defining the number of print lines per page. When the end of the page is reached, the application receives a message and the journal is placed in line 1, column 1 of the next page.

Allowed values are 0 to 63. Value 0 means continuous listing. The default is **0**.

- For type DOCU:

  Allowed values are 1 to 4. The default is **4**.

STARTKEY
Is an optional keyword to define a start key for each operation. The parameter values are:

Y (Yes) Each operation will be started when pressing the start key.

N (No) Each operation will be automatically started.

The default is **N**.

WRNLINE Is an optional keyword to define the number of the line at which a attention item is reported to the application (refer to G in Figure 9 on page 309). Allowed values are from 0 to the specified PAGESIZE. The default is **0**. (Dimension D)

Notes:

1. If the attention line is specified as 0, no attention line checking is done.
2. The value must be 0 if the PAGESIZE is 0.

FORM4710 vector example

```
/* Vector FORM4710 (Journal) Example */

         FORM4710 TYPE=JOUR,
                  NAME=JOURNAL1,
                  CPI=10,
                  LINELEN=40,
                  PAGESIZE=60,
                  WRNLINE=58

/* Vector FORM4710 (Document) Defaults Example */

         FORM4710 TYPE=DOCU,
                  NAME=DUMMYDOC,
                  PAGESIZE=1,
                  WRNLINE=0,
                  LINELEN=40,
                  CPI=10,
                  VRTOFFSL=4,
                  AUTONL=Y,
                  STARTKEY=Y
```

FORM4720 vector

- PC/Integrator
- PC Integrator/2

Defines default parameters (DEVPARM) for banking printer program when emulating a 4720 printer.

The following figure shows the layout of the passbook pages for a 4720 printer:

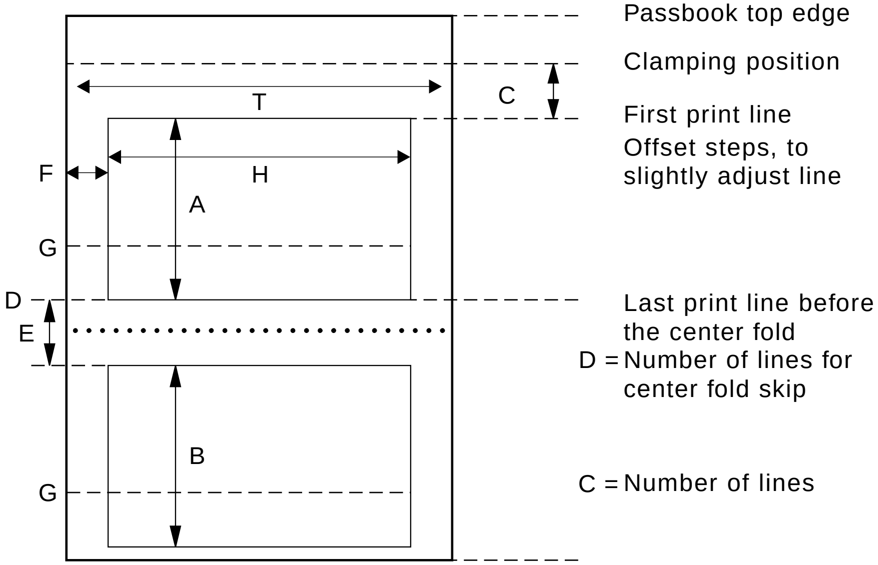

Passbook top edge
Clamping position
T
C
First print line
Offset steps, to slightly adjust line
F
H
A
G
D
E
Last print line before the center fold
D = Number of lines for center fold skip
B
G
C = Number of lines

Figure 10. Passbook Dimensions for IBM 4720 Printer Customization

A quick reference

Vector position	None
List of keywords	See keyword TYPE
Keywords relate to	**Keyword** **Relates to keywords** **NAME** PAR4720 (BPPPARM vector)

Vector format	FORM4720 TYPE=*xxxx*, NAME=*xxxxxxxx*, [AUTONL=*x*,] [BKWIDTH=*xxx*,] [CPI=*xx*,] [DISPLACE=*x*,] [DBLREC=*x*,] [FOLDBEG=*xx*,] [FOLDSKIP=*x*,] [FOLDTYPE=*x*,] [HRZOFFS=*x*,] [JOURADV=*x*,] [LINELEN=*xx*,] [LPI=*x*,] [MAXSKEW=*x*,] [REMSMODE=*x*,] [NEWDOC=*x*,] [VRTOFFSL=*x*,] [VRTOFFSS=*x*,] [PAGESIZE=*xx*,] [QUALITY=*x*,] [STARTKEY=*x*,] [WRNLINE=*x*]

Keyword **Description**

TYPE Is a required keyword to define the type of object. The type can be:

DOCU for document
JOUR for journal
PASS for passbook
REMS for the REMS facility.

Depending on the type you selected, you define a different set of keywords:

DOCU NAME, LPI, CPI, MAXSKEW, LINELEN, PAGESIZE, WRNLINE, QUALITY, VRTOFFSS, VRTOFFSL, AUTONL, STARTKEY, JOURADV

JOUR NAME, LPI, CPI, LINELEN, PAGESIZE, WRNLINE

PASS NAME, LPI, CPI, MAXSKEW, FOLDTYPE, LINELEN, PAGESIZE, FOLDBEG, FOLDSKIP, VRTOFFSS, VRTOFFSL, AUTONL, STARTKEY, HRZOFFS

REMS NAME, REMSMODE, DISPLACE, NEWDOC, DBLREC, BKWIDTH

Use the keywords only with their corresponding types. Otherwise, you will get an error.

NAME Is a required keyword to define the name of the object. The parameter value is a field of up to eight alphanumeric characters, and must be unique among the FORM4720 vectors.

AUTONL Is an optional keyword to define an automatic skip to the next line.. Its values are:

Y (Yes) The printer automatically goes to the next line when the text exceeds the last position on the current line.

N (No) An error message will be issued when the line length is exceeded.

The default is **Y**.

BKWIDTH Is an optional keyword to define the book width of the open document in millimeters (mm). The allowed values are from 127 to 212. The default is **210** (refer to ‘T’ in Figure 10 on page 312).

CPI Is an optional keyword to define printing density (number of characters per inch). The numbers are written in one inch (2.54 cm).

Allowed values are 10, 12, or 16. The default is **10**.

DISPLACE

Is an optional keyword to define whether the magnetic stripe is 20 mm above the bottom edge of the passbook or not.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

DBLREC Is an optional keyword to define whether you want the REMS device to encode in double record mode (the record is written twice to the magnetic stripe) or not.

The parameter value can be either Y (Yes) or N (No). The default is **Y**.

FOLDBEG Is an optional keyword to define the last line or character position before the center-fold skip for horizontal or vertical passbooks (refer to ‘D’ in Figure 10 on page 312).

For horizontal fold passbooks, the maximum value must not be greater than the specified PAGESIZE value.

For vertical fold passbooks, the maximum value must not be greater than the specified LINELEN value.

For both types of center fold, the default is **10**.

FOLDSKIP

Is an optional keyword to define the total number of lines or character positions skipped for horizontal fold or vertical fold passbooks. (See dimension ‘E’ in Figure 10 on page 312.)

The center fold skip can be any value provided that, when added to the page size (PAGESIZE for horizontal fold passbooks) or to the line length (LINELEN for vertical fold passbooks), the sum is less than or equal to the maximum logical page size or the maximum printer line length respectively.

For vertical passbooks, the default is **4**.

FOLDTYPE

Is an optional keyword to define the type of center fold. The values are H (horizontal) or V (vertical). The default is **H**.

HRZOFFS Is an optional keyword to define the horizontal character offset. That is, the number of characters that the first print character is to be offset beyond the minimum distance from the left edge. (See dimension ‘F’ in Figure 10 on page 312).
The value, when added to the line length (LINELEN) and center fold skip (FOLDTYPE) in a vertical fold passbook, must be less than or equal to the maximum line length. The default is **0**.

JOURADV Is an optional keyword to define automatic line advance each time a line is printed in a document. Its values are:

Y (Yes) The printer automatically goes to the next line when the text exceeds the last position on the current line.

N (No) An error message will be issued when the line length is exceeded.

LINELEN Is an optional keyword to define the maximum line length (number of characters being written in one line). (See dimension ‘H’ in Figure 10 on page 312).

The maximum values depend on the value defined for character density CPI. They are:

82 if CPI is 10
99 if CPI is 12
132 if CPI is 16

If 0 is specified, the maximum value is assumed. The default is **80**.

LPI Is an optional keyword to define the line density, that is, the number of lines written in one inch (2.54 cm). The parameter specified determines the maximum page size (PAGESIZE).

The parameter values are 5 or 6 lines per inch. The default is **5**.

MAXSKEW

Is an optional keyword to define the maximum slant of the document placed in the printer. If the allowed skew is exceeded, the document is rejected, and must be reinserted. The maximum skew depends on the initial vertical offset chosen.

The parameter values are:

- For DOCU type objects

  1 for 1.37 mm
  2 for 2.05 mm
  3 for 2.74 mm
  4 for 3.42 mm

- For PASS type objects

  0 for 0.68 mm
  1 for 1.37 mm
  2 for 2.05 mm

The default is **1**.

NEWDOC Is an optional parameter. Its values are:

Y (Yes) If you want the program to be notified of a change-of-document between reading the magnetic stripe and printing the passbook.

N (No) No notification is required.

The default is **N**.

PAGESIZE

Is an optional keyword to define the page size, that is, the number of print lines per page. (See dimension 'A' and 'B' in Figure 10 on page 312.)

The maximum number of lines depends on the line density (LPI) chosen and, in horizontal fold passbooks, the center fold skip.

The parameter values are a numeric field with up to two digits, ranging from 0 to the maximum value allowed.

Maximum values are:

- For DOCU type objects

  70 if LPI is 5
  84 if LPI is 6

  If 0 is specified, the maximum value is assumed. The default is **60**.

- For JOUR type objects

  70 if LPI is 5
  84 if LPI is 6

  If 0 is specified, no end of page checking is performed, and the journal is treated as a continuous list. The default is **0**.

- For PASS type objects

  38 if LPI is 5
  46 if LPI is 6

  If 0 is specified, the maximum value is assumed. The default is **0**.

QUALITY Is an optional keyword to define quality print.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

REMSMODE

Is an optional keyword to define the REMS recording mode. The parameter values are A (for ANSI mode) or D (for DIN mode).

The default is **D**.

STARTKEY

Is an optional keyword to define a start key for each operation. The parameter values are:

Y (Yes) Each operation will be started when pressing the start key.

N (No) Each operation will be automatically started.

The default is **N**.

VRTOFFSL

Is an optional keyword to define the number of lines between the first printable line and the line occupied by the tractor clamping mechanism. (See dimension ‘C’ in Figure 10 on page 312.)

The value specified for VRTOFFSL plus the values for PAGESIZE and VRTOFFSS (and FOLDSKIP for horizontal passbook) must not be greater than the maximum page size.

The default is **0**.

VRTOFFSS

Is an optional keyword to define the offset step, that is, a slight adjustment to print the first vertical position of a document that has preprinted boxes. It defines the number of 1/90-inch steps that the print line is to be offset beyond the minimum distance from the top or bottom edge of a cut form.

The parameter values are a numeric field with up to two digits, ranging from 0 to the maximum value allowed.

Maximum values are:

18 if LPI is 5
15 if LPI is 6

The default is **0**.

WRNLINE Is an optional keyword to define the number of the line at which an attention item is reported to the application. (See dimension ‘G’ in Figure 10 on page 312.) Allowed values are from 0 to the specified PAGESIZE. The default is **0**.

Notes:

1. If the attention line is specified as 0, no attention line checking is done.
2. The value must be 0 if the PAGESIZE is 0.

FORM4720 vector example

```
/* Vector FORM4720 (Passbook) Example */

        FORM4720 TYPE=PASS,
                 NAME=PASS001,
                 LPI=5,
                 CPI=10,
                 MAXSKEW=1,
                 FOLDTYPE=V,
                 LINELEN=70,
                 PAGESIZE=38,
                 FOLDBEG=30,
                 FOLDSKIP=1,
                 VRTOFFSS=0,
                 VRTOFFSL=0,
                 HRZOFFS=10,
                 AUTONL=Y,
                 STARTKEY=N
/* Vector FORM4720 (Document) Default Example */

        FORM4720 TYPE=DOCU,
                 NAME=DUMMYDOC,
                 PAGESIZE=1,
                 WRNLINE=0,
                 LINELEN=82,
                 STARTKEY=Y,
                 AUTONL=Y,
                 LPI=5,
                 CPI=10,
                 VRTOFFSL=0,
                 VRTOFFSS=0,
                 JOURADV=N,
                 QUALITY=N
```

FORM47X2 vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- PC/Integrator
- PC Integrator/2

Defines financial printer device parameters for documents, journals, passbooks, and REMS objects.

Define one FORM47X2 vector for each financial printer object you want to use in your workgroups (documents, journals, passbooks, or REMS).

The following figure shows the layout of the 4772 and 9068 passbook pages:

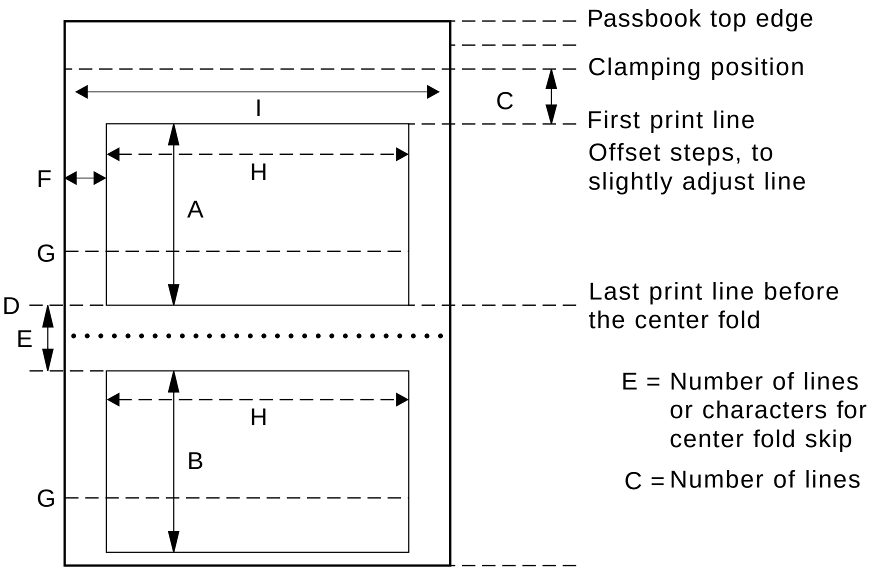

Passbook top edge
Clamping position
C
I
First print line
Offset steps, to slightly adjust line
F
H
A
G
D
E
Last print line before the center fold
E = Number of lines or characters for center fold skip
H
B
G
C = Number of lines

Figure 11. Passbook Dimensions for IBM 4772 and 9068 Printer Customization

The following figure shows the layout of the document pages:

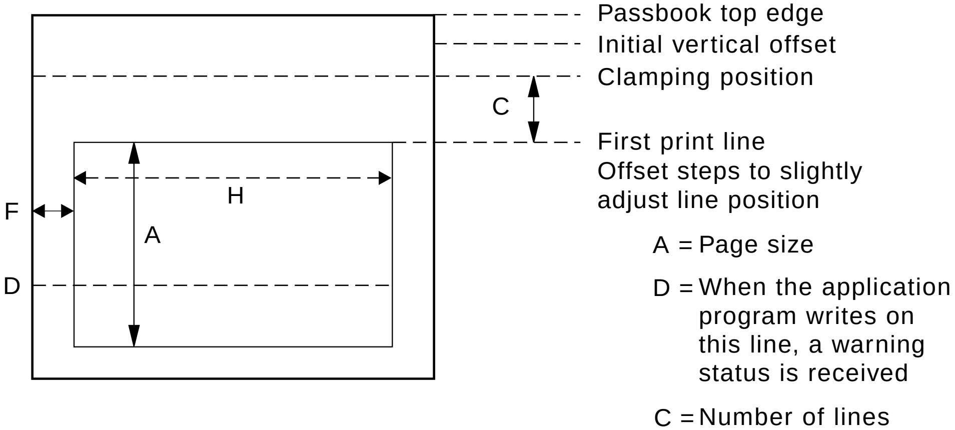

Passbook top edge
Initial vertical offset
Clamping position
C
First print line
Offset steps to slightly adjust line position
F
H
A
D
A = Page size
D = When the application program writes on this line, a warning status is received
C = Number of lines

Figure 12. Document Dimensions for IBM 4009, 4712, 4722, 4772, 9068 and 9069 Printer Customization

A quick reference

Vector position	None
List of keywords	See keyword TYPE
Keywords relate to	**Keyword** **Relates to keywords** **NAME** PAR4712 and PAR4722 (BPPPARM vector)
Vector format	FORM47X2 TYPE=*xxxx*, NAME=*xxxxxxxx*, [AUTONL=*x*,] [BKWIDTH=*xxx*,] [BOTTVAL=*xxx*,] [CPI=*xx*,] [DBLLINE=*x*,] [DBLREC=*x*,] [DBLSTRK=*x*,] [DISPLACE=*x*,] [EMPHASIZ=*x*,] [FOLDBEG=*xx*,] [FOLDSKIP=*x*,] [FOLDTYPE=*x*,] [HIGHCHR=*x*,] [HRZOFFS=*x*,] [LINELEN=*xx*,] [LPI=*x*,] [MAXSKEW=*x*,] [NEWDOC=*x*,] [PAGESIZE=*xx*,] [QUALITY=*x*,] [REMSMODE=*x*,] [SHARED=*x*,] [STARTKEY=*x*,] [VRTOFFSI=*x*,] [VRTOFFSL=*x*,] [VRTOFFSS=*x*,] [WIDECHR=*x*,] [WRNLINE=*x*]

Keyword **Description**

TYPE Is a required keyword to define the type of object. The parameter values are DOCU, JOUR, PASS, or REMS. Following are the keywords which have to be defined for each type:

DOCU NAME, LPI, CPI, VRTOFFSI, MAXSKEW, LINELEN, PAGESIZE, WRNLINE, QUALITY, DBLSTRK, EMPHASIZ, VRTOFFSS, VRTOFFSL, HRZOFFS, AUTONL, STARTKEY, SHARED, BOTTVAL, DBLLINE, HIGHCHR, WIDECHR

JOUR NAME, LPI, CPI, LINELEN, PAGESIZE, WRNLINE, QUALITY, DBLSTRK, EMPHASIZ, SHARED

PASS NAME, LPI, CPI, VRTOFFSI, MAXSKEW, FOLDTYPE, LINELEN, PAGESIZE, FOLDBEG, FOLDSKIP, VRTOFFSS, VRTOFFSL, HRZOFFS, AUTONL, STARTKEY, SHARED

REMS NAME, REMSMODE, DISPLACE, NEWDOC, DBLREC, BKWIDTH

Use the keywords only with their corresponding types. Otherwise, you will get an error.

NAME Is a required keyword to define the name of the object. The parameter value is a field of up to eight alphanumeric characters, and must be unique among the FORM47X2 and FORM4770 vectors.

AUTONL Is an optional keyword to define an automatic skip to next line. Its values are:

Y (Yes) The printer automatically goes to the next line when the text exceeds the last position on the current line.

N (No) An error message will be issued when the line length is exceeded.

The default is **Y**.

BKWIDTH Is an optional keyword to define the book width of the open document in millimeters (mm). The allowed values are from 127 to 212. The default is **210** (refer to ‘I’ in Figure 11 on page 319.).

BOTTVAL Is an optional keyword to define whether the bottom edge validation will be ON or OFF.

The maximum size for the printable area of a document to be checked is 2 inches.

The default is **OFF**.

CPI Is an optional keyword to define printing density (number of characters per inch). Allowed values are 10, 12, 15, or 17 (for 17.1). The default is **10**.

DISPLACE

Is an optional keyword to define whether the magnetic stripe is 10 mm above the bottom edge of the passbook or not.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

DBLLINE Is an optional keyword to define whether double line feed is skipped when printing, or not.

The parameter applies only to the 4772 printer. The parameter value can be either Y (Yes) or N (No). The default is **N**.

DBLREC Is an optional keyword to define whether you want the REMS device to encode in double record mode (the record is written twice to the magnetic stripe) or not.

The parameter value can be either Y (Yes) or N (No). The default is **Y**.

DBLSTRK Is an optional keyword to define double strike printing mode.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

EMPHASIZ

Is an optional keyword to define emphasized characters.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

FOLDBEG Is an optional keyword to define the last line or character position before the center-fold skip for horizontal or vertical passbooks (refer to ‘D’ in Figure 12 on page 319.).

For horizontal fold passbooks, the minimum value is 1, and the maximum must not be greater than the specified PAGESIZE value.

For vertical fold passbooks, the minimum value is 1, and the maximum must not be greater than the specified LINELEN value.

If 0 is specified, the fold skip is considered to be in the page center for horizontal fold passbooks, and in the line center for vertical fold passbooks. The default is **10**.

FOLDSKIP

Is an optional keyword to define the total number of lines or character positions skipped for horizontal fold or vertical fold passbooks. (Refer to ‘E’ in Figure 11 on page 319)

The center fold skip can be any value such that, when added to the page size (PAGESIZE for horizontal fold passbooks) or to the line length (LINELEN for vertical fold passbooks), the sum is less than or equal to the maximum logical page size or the maximum printer line length respectively.

The default is **4**.

FOLDTYPE

Is an optional keyword to define the type of center fold. The values are H (horizontal) or V (vertical). The default is **H**.

HIGHCHR Is an optional keyword to define whether characters are printed with double height, or not.

The parameter applies only to the 4772 printer, and when double line feed is selected through the DBLLINE keyword. The parameter value can be either Y (Yes) or N (No). The default is **N**.

HRZOFFS Is an optional keyword to define the horizontal character offset. That is, the number of characters that the first print character is to be offset beyond the minimum distance from the left edge (refer to ‘F’ in figures 11 and 12 on page 319).

The value, when added to the line length and center fold skip in a vertical fold passbook, must be less than or equal to the maximum line length. The default for PASS type objects is 0, and for DOCU type objects 1.

See parameter LINELEN (maximum line length) for reference.

LINELEN Is an optional keyword to define the maximum line length (number of characters being written in one line) (refer to ‘H’ in figures 11 and 12 on page 319).

The parameter values depend on the value defined for CPI, and on the printer supported by the financial printer server. The allowed values range from 0 to the following CPI values:

- For DOCU type objects (4712 and 9069 printers)

  40 if CPI is 10
  48 if CPI is 12
  68 if CPI is 17.1

  The default is **80**.

- For DOCU type objects (4722 printer)

  82 if CPI is 10
  99 if CPI is 12
  141 if CPI is 17.1

  The default is **80**.

- For DOCU type objects (4772 printer, WIDECHR=Y)

  41 if CPI is 10
  49 if CPI is 12
  61 if CPI is 15
  70 if CPI is 17.1

  The default is **41**.

- For DOCU type objects (4772 printer, WIDECHR=N)

  82 if CPI is 10
  99 if CPI is 12
  123 if CPI is 15
  141 if CPI is 17.1

  The default is **82**.

- For JOUR type objects (4712, 4722, 9068, or 9069 printer)

  40 if CPI is 10
  48 if CPI is 12
  68 if CPI is 17.1

  The default is **40**.

- For PASS type objects (4712, 4722, 9068 or 9069 printer)

  82 if CPI is 10
  99 if CPI is 12
  141 if CPI is 17.1

  The default is **80**.

The maximum value will be assumed if you specify 0 in any object type.

LPI Is an optional keyword to define the number of lines written in one inch (2.54 cm). The parameter specified determines the maximum PAGESIZE.

The parameter values depend on the object type:

- For DOCU type objects, values are 5, 6, or 8.

  The default is **5**.

- For JOUR type objects, values are 6 or 8.

  The default is **6**.

- For PASS type objects, values are 5 or 6.

  The default is **5**.

MAXSKEW

Is an optional keyword to define the maximum slant of the document placed in the printer.

If the allowed skew is exceeded, the document is rejected, and must be reinserted. The maximum skew depends on the initial vertical offset chosen.

The parameter values are:

- For DOCU type objects, VRTOFFSI=0

  1 for 1.37 mm
  2 for 2.05 mm

- For DOCU type objects, VRTOFFSI=1

  1 for 1.37 mm
  2 for 2.05 mm
  3 for 2.74 mm
  4 for 3.42 mm

- For PASS type objects, VRTOFFSI=0

  0 for 0.68 mm
  1 for 1.37 mm

- For PASS type objects, VRTOFFSI=1

  0 for 0.68 mm
  1 for 1.37 mm
  2 for 2.05 mm

The default is **1**.

NEWDOC Is an optional keyword to define whether the program must be notified of a change of document between two different Read/Encode magnetic stripe operations.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

PAGESIZE

Is an optional keyword to define the page size, that is, the number of print lines per page. (See dimension ‘A’ and ‘B’ in Figure 11 on page 319.)

The maximum number of lines depends on the line density (LPI) chosen and, in horizontal fold passbooks, the center fold skip. It also depends on the printer supported by the financial printer server.

The parameter values are a numeric field with up to two digits, ranging from 0 to the maximum value allowed.

Maximum values are:

- For DOCU type objects (4712 printer)

  10 if LPI is 5
  12 if LPI is 6
  16 if LPI is 8

  If 0 is specified, the default (**4**) will be taken.

- For DOCU type objects (4722 or 9069 printer)

  137 if LPI is 5
  165 if LPI is 6
  220 if LPI is 8

  If 0 is specified, the maximum value is assumed. The default is **60**.

- For DOCU type objects (4772 printer, DBLLINE=Y)

  68 if LPI is 5
  82 if LPI is 6
  110 if LPI is 8

  If 0 is specified, the maximum value is assumed. The default is **82**.

- For DOCU type objects (4772 printer, DBLLINE=N)

  137 if LPI is 5
  165 if LPI is 6
  220 if LPI is 8

  If 0 is specified, the maximum value is assumed. The default is **165**.

- For JOUR type objects (4712, 4722, or 9069 printer)

  Permits the journal to be split up in logical pages by defining the number of print lines per page. When the end of the page is reached, the application receives a message and the journal is placed in line 1, column 1 of the next page.

  66 if LPI is 6
  88 if LPI is 8

  If 0 is specified, no end of page checking will be performed, and the journal is treated as a continuous list. The default is **0**.

- For PASS type objects (4712, 4722, or 9069 printer)

  38 if LPI is 5
  46 if LPI is 6

  If 0 is specified, the maximum value is assumed. The default is **30**.

QUALITY Is an optional keyword to define quality print.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

REMSMODE
Is an optional keyword to define the REMS recording mode. The parameter values are:

D ISO/DIN mode
E ISO/DIN mode on track 1
I IBM 3604/4704 mode
J IBM 3604/4704 mode on track 2

The default is **D**.

SHARED Is an optional keyword to define if the printer will be used in shared A/B mode.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

If you select Y for this keyword, you also have to select Y for keyword STARTKEY.

STARTKEY
Is an optional keyword to define a start key for each operation. The parameter values are:

Y (Yes) Each operation will be started when pressing the start key.

N (No) Each operation will be automatically started.

The default is **N**.

If you have selected Y for keyword SHARED, you also have to select Y here.

VRTOFFSI Is an optional keyword to define the initial vertical offset, that is, the top margin lines. Parameter values are 0 (3.9 mm offset) or 1 (6.6 mm offset). The default is **0**.

VRTOFFSL
Is an optional keyword to define the number of lines between the first printable line and the line occupied by the tractor clamping mechanism (refer to ‘C in figures 11 and 12 on page 319).

The value specified for VRTOFFSL plus the values for PAGESIZE and VRTOFFSS (and FOLDSKIP for horizontal passbook) must not be greater than maximum page size. The default is **0**.

VRTOFFSS
Is an optional keyword to define the offset step, that is, a slight adjustment to print the first vertical position of a document that has preprinted boxes.

It defines the number of 1/90-inch steps that the print line is to be offset beyond the minimum distance from the top or bottom edge of a cut form.

The parameter values are a numeric field with up to two digits, ranging from 0 to the maximum value allowed.

Maximum values are:

- For DOCU type objects:

  18 if LPI is 5
  15 if LPI is 6
  11 if LPI is 8

- For PASS type objects:

  18 if LPI is 5
  15 if LPI is 6

The default is **0**.

WIDECHR Is an optional keyword to define whether characters are printed with double width, or not.

The parameter applies only to the 4772 printer. The parameter value can be either Y (Yes) or N (No). The default is **N**.

WRNLINE Is an optional keyword to define the number of the line at which an attention item is reported to the application. Allowed values are from 0 to the specified PAGESIZE. The default is **0**. (refer to ‘G in figures 11 and 12 on page 319).

Note: If the attention line is specified as 0, no attention line checking is done.

FORM47X2 vector example

```
 /* Vector FORM47X2 (Passbook) Example */

        FORM47X2 TYPE=PASS,
                 NAME=PASS001,
                 LPI=5,
                 CPI=10,
                 MAXSKEW=1,
                 FOLDTYPE=V,
                 LINELEN=70,
                 PAGESIZE=38,
                 FOLDBEG=30,
                 FOLDSKIP=1,
                 VRTOFFSS=0,
                 VRTOFFSL=0,
                 HRZOFFS=10,
                 AUTONL=Y,
                 STARTKEY=N,
                 SHARED=N


 /* Vector FORM47X2 (REMS) Example */

        FORM47X2 TYPE=REMS,
                 NAME=REMS01,
                 REMSMODE=D,
                 DISPLACE=Y,
                 NEWDOC=N,
                 DBLREC=N,
                 BKWIDTH=135


/* Vector FORM47X2 (Document) Default Example */

        FORM47X2 TYPE=DOCU,
                 NAME=DUMMYDOC,
                 PAGESIZE=0,
                 WRNLINE=0,
                 VRTOFFSS=0,
                 VRTOFFSL=0,
                 LINELEN=0,
                 SHARED=N,
                 STARTKEY=Y,
                 AUTONL=Y,
                 CPI=10,
                 LPI=5,
                 QUALITY=N,
                 DBLSTRK=N,
                 EMPHASIZ=N,
                 MAXSKEW=1,
                 BOTTVAL=OFF,
                 VRTOFFSI=1
```

```
/* Vector FORM47X2 (Journal) Default Example */

         FORM47X2 TYPE=JOUR,
                  NAME=DUMMYJOU,
                  PAGESIZE=0,
                  WRNLINE=0,
                  LINELEN=0,
                  QUALITY=N,
                  DBLSTRK=N,
                  EMPHASIZ=N,
                  LPI=6,
                  CPI=10,
                  SHARED=N
```

FORM4748 vector

- LANDP for DOS.
- LANDP for OS/2
- LANDP for Windows NT

Defines 4748 printer device parameters for documents and passbooks objects.

Define one FORM4748 vector for each 4748 printer object you want to use in your workgroups (documents or passbooks).

The following figure shows the layout of the document pages:

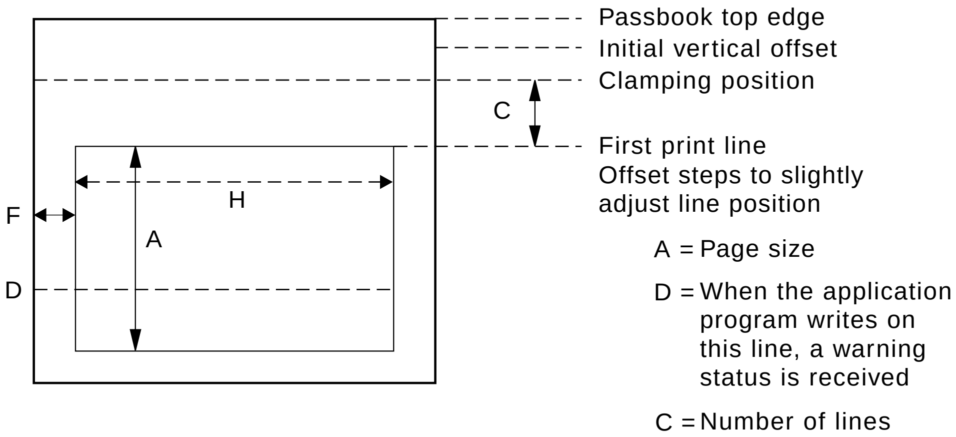

Passbook top edge
Initial vertical offset
Clamping position
C
First print line
Offset steps to slightly adjust line position
H
F
A
D
A = Page size
D = When the application program writes on this line, a warning status is received
C = Number of lines

Figure 13. Document Dimensions for IBM 4748 Printer Customization

The following figure shows the layout of the passbook pages:

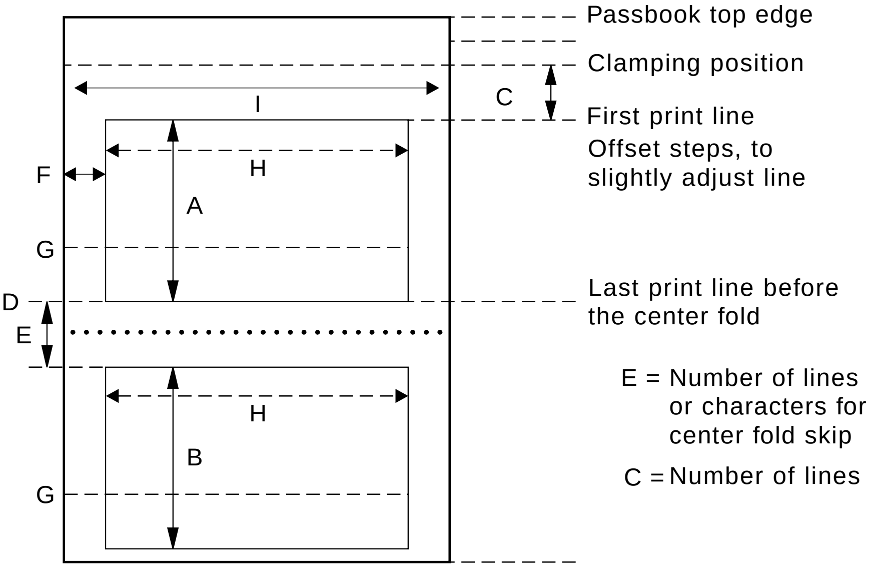

Passbook top edge
Clamping position
C
First print line
Offset steps, to slightly adjust line
I
H
F
A
G
D
E
Last print line before the center fold
H
B
G
E = Number of lines or characters for center fold skip
C = Number of lines

Figure 14. Passbook Dimensions for IBM 4748 Printer Customization

A quick reference

Vector position	None
List of keywords	See keyword TYPE
Keywords relate to	
Vector format	FORM4748 TYPE=*xxxx*, NAME=*xxxxxxxx*, [AUTONL=*x*,] [BKWIDTH=*xxx*,] [BOTTVAL=*xxx*,] [CPI=*xx*,] [DBLREC=*x*,] [DBLSTRK=*x*,] [DISPLACE=*x*,] [EMPHASIZ=*x*,] [FOLDADDS=*xx*,] [FOLDBEG=*xx*,] [FOLDSKIP=*x*,] [FOLDTYPE=*x*,] [FONT=*x*,] [HISPEED=*x*,] [HRZOFFS=*x*,] [INITPOS=*xxxxx*,] [LINELEN=*xx*,] [LPI=*x*,] [MAXSKEW=*x*,] [NEWDOC=*x*,] [PAGESIZE=*xx*,] [REMSMODE=*x*,] [SHARED=*x*,] [STARTKEY=*x*,] [VRTOFFSL=*x*,] [VRTOFFSS=*x*,] [WRNLINE=*x*]

Keyword **Description**

TYPE Is a required keyword to define the type of object. The parameter values are DOCU, PASS, or REMS. Following are the keywords which have to be defined for each type:

DOCU NAME, LPI, CPI, MAXSKEW, LINELEN, PAGESIZE, WRNLINE, DBLSTRK, EMPHASIZ, VRTOFFSS, VRTOFFSL, HRZOFFS, AUTONL, STARTKEY, SHARED, BOTTVAL, HISPEED, FONT

PASS NAME, LPI, CPI, MAXSKEW, FOLDTYPE, LINELEN, PAGESIZE, FOLDADDS, FOLDBEG, FOLDSKIP, DBLSTRK, VRTOFFSS, VRTOFFSL, HRZOFFS, AUTONL, STARTKEY, SHARED, FONT

REMS NAME, BKWIDTH, DBLREC, DISPLACE, INITPOS, NEWDOC, REMSMODE

Use the keywords only with their corresponding types. Otherwise, you will get an error.

NAME Is a required keyword to define the name of the object. The parameter value is a field of up to eight alphanumeric characters, and must be unique among the FORM4748 vectors.

AUTONL Is an optional keyword to define automatic skip to the next line. Its values are:

Y (Yes) The printer automatically goes to the next line when the text exceeds the last position on the current line.

N (No) An error message will be issued when the line length is exceeded.

The default is **Y**.

BKWIDTH Is an optional keyword to define the width of an open passbook in millimeters. The parameter value is a numeric field with three digits, in the range 120 through 213. The default is **210**. It is valid only when TYPE=REMS. (Refer to 'I' in Figure 14 on page 330.)

BOTTVAL Is an optional keyword to define whether the bottom edge validation will be ON or OFF.

The maximum size for the printable area of a document to be checked is 2 inches.

The default is **OFF**.

CPI Is an optional keyword to define printing density (number of characters per inch). Allowed values are 10, 12, 13, or 15.

Value 12 cannot be specified if you assign value 1, 3, or 5 to FONT keyword; value 13 or 15 cannot be specified if you assign value 1, 2, 3, or 5 to FONT keyword. The default is **10**.

DBLREC Is an optional keyword to define whether you want the REMS device to encode using double record mode (the record is written twice to the magnetic stripe). It is valid only when TYPE=REMS.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

DBLSTRK Is an optional keyword to define double strike printing mode.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

DBLREC Is an optional keyword to define whether the magnetic stripe is 10 mm above the bottom edge of the passbook. It is valid only when TYPE=REMS.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

EMPHASIZ

Is an optional keyword to define emphasized characters.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

FOLDADDS
Is an optional keyword to specify a slight adjustment in the fold (horizontal or vertical, depending on FOLDTYPE) of the passbook. FOLDADDS defines by how much the print line is to be offset from the fold skip start position, in units of 1/120-inch. It is valid only when TYPE=PASS.

The default is **0**.

The maximum number of xx for horizontal folds is:

24 if LPI is 5
20 if LPI is 6

The maximum number of xx for vertical folds is:

18 if LPI is 10
15 if LPI is 12
13 if LPI is 13
12 if LPI is 15

FOLDBEG Is an optional keyword to define the last line or character position before the center-fold skip for horizontal or vertical passbooks (refer to ‘D’ in Figure 14 on page 330).

For horizontal fold passbooks, the minimum value is 1, and the maximum must not be greater than the specified PAGESIZE value.

For vertical fold passbooks, the minimum value is 1, and the maximum must not be greater than the specified LINELEN value.

If **0** is specified, the fold skip is considered to be in the page center for horizontal fold passbooks, and in the line center for vertical fold passbooks. The default is **10**.

FOLDSKIP
Is an optional keyword to define the total number of lines or character positions skipped for horizontal fold or vertical fold passbooks. (See dimension ‘E’ in Figure 14 on page 330.)

The center fold skip can be any value such that, when added to the page size (PAGESIZE for horizontal fold passbooks) or to the line length (LINELEN for vertical fold passbooks), the sum is less than or equal to the maximum logical page size or the maximum printer line length respectively.

The default is **4**.

FOLDTYPE
Is an optional keyword to define the type of center fold. The values are H (horizontal) or V (vertical). The default is **H**.

FONT Is an optional keyword to define the type characters for the 4748 printer. The values are:

0 Mincho-12 (requires CPI=10, 12, 13, or 15)
1 DP Gothic-10 (requires CPI=10)
2 Prestige Elite-12 (requires CPI=10, or 12)
3 Courier-10 (requires CPI=10)
4 Mincho-12 (requires CPI=10, 12, 13, or 15)
5 Mincho-10 (requires CPI=10)

The default is **0**.

HISPEED Is an optional keyword to define high speed printing mode.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

HRZOFFS Is an optional keyword to define the horizontal character offset. That is, the number of characters that the first print character is to be offset beyond the minimum distance from the left edge (refer to ‘F’ in Figure 14 on page 330).

The value, when added to the line length and center fold skip in a vertical fold passbook, must be less than or equal to the maximum line length. The default for PASS type objects is 0, and for DOCU type objects 1.

See parameter LINELEN (maximum line length) for reference.

INITPOS Is an optional keyword to define the initial print line position when a passbook is inserted. It is valid only when TYPE=REMS.

The parameter value can be either FIRST or REMS. The default is **FIRST**.

LINELEN Is an optional keyword to define the maximum line length (number of characters being written in one line). (Refer to ‘H’ in Figure 14 on page 330.)

The parameter values depend on the value defined for CPI. The allowed values range from 0 to the following CPI values:

82 if CPI is 10
99 if CPI is 12
110 if CPI is 13
123 if CPI is 15

If you specify 0, the maximum value will be assumed. The default is **80**.

LPI Is an optional keyword to define the number of lines written in one inch (2.54 cm). The parameter specified determines the maximum PAGESIZE.

The parameter values depend on the object type. They are:

- For DOCU type objects, 5, 6, or 8.

  The default is **5**.

- For PASS type objects, 5 or 6.

  The default is **5**.

MAXSKEW

Is an optional keyword to define the maximum slant of the document or passbook object placed in the printer.

The parameter values are:

- For DOCU type objects:

  0 for 1.37 mm
  1 for 0.68 mm
  2 for 2.05 mm
  3 for 2.74 mm

  The default is **1**.

- For PASS type objects:

  0 for 1.37 mm
  1 for 0.68 mm
  2 for 2.05 mm

  The default is **1**.

NEWDOC Is an optional keyword to define whether the program must be notified of a change of document between two different read/encode magnetic stripe operations. It is valid only when TYPE=REMS.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

PAGESIZE

Is an optional keyword to define the page size, that is, the number of print lines per page. (See dimension ‘A’ and ‘B’ in Figure 14 on page 330.)

The maximum number of lines depends on the line density (LPI) chosen and, in horizontal fold passbooks, the center fold skip.

The parameter values are a numeric field with up to three digits, ranging from 0 to the maximum value allowed.

Maximum values are:

- For DOCU type objects:

  137 if LPI is 5
  165 if LPI is 6
  220 if LPI is 8

  If you specify 0, the maximum value will be assumed. The default is **60**.

- For PASS type objects:

  38 if LPI is 5
  46 if LPI is 6

  If you specify 0, the maximum value will be assumed. The default is **30**.

REMSMODE

Is an optional keyword to define the REMS recording mode. It is valid only when TYPE=REMS.

The parameter values are:

D	ISO/DIN mode
I	IBM 3604/4704 mode

The default is **D**.

SHARED Is an optional keyword to define if the printer will be used in shared A/B mode.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

If you select Y for this keyword, you also have to select Y for keyword STARTKEY.

STARTKEY

Is an optional keyword to define a start key for each operation. The parameter values are:

Y (Yes) Each operation will be started when pressing the start key.

N (No) Each operation will be automatically started.

The default is **N**.

If you have selected Y for keyword SHARED, you also have to select Y here.

VRTOFFSL

Is an optional keyword to define the number of lines between the first printable line and the line occupied by the tractor clamping mechanism (refer to 'C' in Figure 14 on page 330).

The value specified for VRTOFFSL plus the values for PAGESIZE and VRTOFFSS (and FOLDSKIP for horizontal passbook) must not be greater than maximum page size. The default is **0**.

VRTOFFSS

Is an optional keyword to define the offset step, that is, a slight adjustment to print the first vertical position of a document that has preprinted boxes. It defines the number of 1/120-inch steps that the print line is to be offset beyond the minimum distance from the top or bottom edge of a cut form.

The parameter values are a numeric field with up to two digits, ranging from 0 to the maximum value allowed.

Maximum values are:

- For DOCU type objects:

  24 if LPI is 5
  20 if LPI is 6
  15 if LPI is 8

- For PASS type objects:

  24 if LPI is 5
  20 if LPI is 6

The default is **0**.

WRNLINE Is an optional keyword to define the number of the line at which an attention item is reported to the application. Allowed values are from 0 to the specified PAGESIZE. The default is **0**. (Refer to ‘G’ in Figure 14 on page 330).

Note: If the attention line is specified as 0, no attention line checking is done.

FORM4748 vector example

```
/* Vector FORM4748 (Document) Example */

        FORM4748 TYPE=DOCU,
                 NAME=DOC01,
                 LPI=5,
                 CPI=10,
                 MAXSKEW=1,
                 LINELEN=80,
                 PAGESIZE=60,
                 WRNLINE=0,
                 DBLSTRK=N,
                 EMPHASIZ=N,
                 VRTOFFSS=0,
                 VRTOFFSL=0,
                 HRZOFFS=1,
                 AUTONL=Y,
                 STARTKEY=N,
                 SHARED=N,
                 BOTTVAL=OFF,
                 HISPEED=N,
                 FONT=4
```

FORM4770 vector

- LANDP for OS/2

Defines 4770 printer device parameters for documents and journals objects.

Define one FORM4770 vector for each 4770 printer object you want to use in your workgroups (documents or journals).

The following figure shows the layout of the document pages:

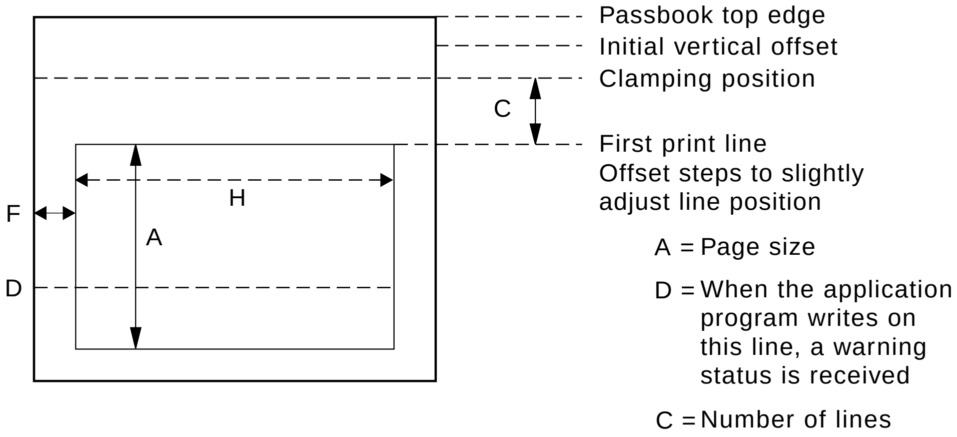

Passbook top edge
Initial vertical offset
Clamping position
C
First print line
Offset steps to slightly adjust line position
H
F
A
D
A = Page size
D = When the application program writes on this line, a warning status is received
C = Number of lines

Figure 15. Document Dimensions for IBM 4770 Printer Customization

A quick reference

Vector position	None
List of keywords	See keyword TYPE
Keywords relate to	
Vector format	FORM4770 TYPE=*xxxx*, NAME=*xxxxxxxx*, [AUTONL=*x*,] [FONT=*x*,] [HRZOFFS=*xx*,] [LINELEN=*xx*,] [PAGESIZE=*x*,] [SPACING=*x*,] [UPSDOWN=*x*,] [VRTOFFSL=*x*,] [VRTOFFSS=*xx*,] [WIDECHR=*x*,] [WRNLINE=*x*]

Keyword **Description**

TYPE Is a required keyword to define the type of object. The parameter values are DOCU or JOUR. Following are the keywords which have to be defined for each type:

DOCU NAME, AUTONL, FONT, HRZOFSS, LINELEN, PAGESIZE, SPACING, UPSDOWN, VRTOFFSL, VRTOFFSS, WIDECHR, WRNLINE

JOUR NAME, AUTONL, FONT, SPACING, UPSDOWN, WIDECHR

Use the keywords only with their corresponding types. Otherwise, you will get an error.

NAME Is a required keyword to define the name of the object. The parameter value is a field of up to eight alphanumeric characters, and must be unique among the FORM47X2 and FORM4770 vectors.

AUTONL Is an optional keyword to define an automatic skip to next line. Its values are:

Y (Yes) The printer automatically goes to the next line when the text exceeds the last position on the current line.

N (No) An error message will be issued when the line length is exceeded.

The default is **Y**.

FONT Is an optional keyword to define the type characters for the 4770 printer. The parameter value can be:

S Standard
L Large
B Bold

The default is **S**.

HRZOFFS Is an optional keyword to define the horizontal character offset. That is, the number of characters that the first print character is to be offset beyond the minimum distance from the left edge. (Refer to ‘F’ in Figure 15 on page 338).

The value, when added to the line length, must be less than or equal to the maximum line length. The default is **0**.

See parameter LINELEN (maximum line length) for reference.

LINELEN Is an optional keyword to define the maximum line length (number of characters being written in one line). (Refer to ‘H’ in Figure 15 on page 338).

The value, when added to the horizontal character offset, must be less than or equal to the maximum line length. The default is **0**.

The maximum line length is:

16 FONT=L, SPACING=T, and WIDECHR=Y; or FONT=B, SPACING=T, and WIDECHR=Y

17 FONT=L, SPACING=L, and WIDECHR=Y; or FONT=B, SPACING=L, and WIDECHR=Y

21 FONT=S, SPACING=T, and WIDECHR=Y

24 FONT=S, SPACING=L, and WIDECHR=Y

32 FONT=L, SPACING=T, and WIDECHR=N; or FONT=B, SPACING=T, and WIDECHR=N

35 FONT=L, SPACING=L, and WIDECHR=N; or FONT=B, SPACING=L, and WIDECHR=N

42 FONT=S, SPACING=T, and WIDECHR=N

48 FONT=S, SPACING=L, and WIDECHR=N

PAGESIZE

Is an optional keyword to define the page size, that is, the number of print lines per page. (See dimension ‘A’ in Figure 15 on page 338.)

The value, when added to the value specified in the VRTOFFSL keyword, must be less than or equal to the maximum page length.

The parameter values are a numeric field with up to two digits, ranging from 0 to the maximum value allowed. The default is **0**.

The maximum page length is:

6 SPACING=T
8 SPACING=L

SPACING Is an optional keyword to define the horizontal and vertical density of characters.

The parameter value can be either T (text) or L (logo). The default is **T**.

UPSDOWN

Is an optional keyword to define whether inverted printing (upside-down mode) will be used, or not.

The parameter value can be Y, to use inverted printing, or N, not to use it. The default is **N**.

VRTOFFSL

Is an optional keyword to define the number of lines between the first printable line and the real first printing line. (Refer to ‘C’ in Figure 15 on page 338).

The value, when added to the value specified in the PAGESIZE keyword, must be less than or equal to the maximum page length. The default is **0**.

VRTOFFSS

Is an optional keyword to define the offset step, that is, a slight adjustment to print the first vertical position of a document that has preprinted boxes. It defines the number of 1/90-inch steps that the print line is to be offset beyond the minimum distance from the top or bottom edge of a cut form.

The parameter values are a numeric field with up to two digits, ranging from 0 to the maximum value allowed. The default is **0**.

Maximum values are:

12 SPACING=L
15 SPACING=T

WIDECHR Is an optional keyword to define whether characters are printed with double width, or not.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

WRNLINE Is an optional keyword to define the number of the line at which an attention item is reported to the application. Allowed values are from 0 to the specified PAGESIZE. The default is **0**. (Refer to ‘D’ in Figure 15 on page 338).

Note: If the attention line is specified as 0, no attention line checking is done.

FORM4770 vector example

```
/* Vector FORM4770 (Document) Example */

         FORM4770 TYPE=DOCU,
                  NAME=DOCU01,
                  AUTONL=Y,
                  FONT=S,
                  HRZOFSS=0,
                  LINELEN=48,
                  PAGESIZE=6,
                  SPACING=T,
                  UPSDOWN=N,
                  VRTOFFSL=0,
                  VRTOFFSS=0,
                  WIDECHR=N,
                  WRNLINE=0

/* Vector FORM4770 (Journal) Example */

         FORM47X2 TYPE=JOUR,
                  NAME=JOUR01,
                  AUTONL=Y,
                  FONT=S,
                  SPACING=T,
                  UPSDOWN=N,
                  WIDECHR=N
```

FORWDS vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- LANDP for AIX

Defines parameters for forwarding a data set, that is, sending a data set to the host. Up to 64 FORWDS vectors can be defined in a forwarding server profile. The number of FORWDS vectors must match the number of data sets defined in the store-for-forwarding profile used.

A quick reference

Vector position	Follows FORWPRF vector. This vector can exist up to 64 times for each FORWPRF vector.
List of keywords	SESSION, TRANSACT, PRTY, EBCDICXL, AUTTRANS
Keywords relate to	**Keyword** **Relates to keywords** **SESSION** SES&FORW (parameter 2) **EBCDICXL** EXTEN (XLATETBL vector)
Vector format	FORWDS SESSION=*x*, TRANSACT=*xxxxxxxx*, [PRTY=*xx*,] [EBCDICXL=*x*,] [AUTTRANS=*x*]

Keyword **Description**

SESSION Is a required keyword to define the forwarding session used to send this data set.

The parameter value is a number from 1 to 3, depending on the number of sessions defined in the forwarding profile, and cannot be omitted.

TRANSACT

Is a required keyword to define the transaction code used when sending records to the host. This code is translated to EBCDIC before sending it to the host.

If the host session (SxTYPE in the FORWPRF vector) was **IMS**, this parameter is an alphanumeric field of up to eight characters. If the host session was **CICS**, the field has up to four characters. You can include blanks if the transaction code does not fill the field.

PRTY Is a keyword to define the priority level for this data set.

This keyword is required, if DSPRTY=Y was specified in the FORWPRF vector. If not, it must be omitted.

The parameter value can be any number from 0 to 99, where 0 means no priority and 99 corresponds to the highest priority. The default is **0**. Two data sets cannot have the same priority in the same session.

EBCDICXL

Is an optional keyword to define if ASCII to EBCDIC translation of fields defined as character or ASCII numeric will be performed before sending data to the host.

In DBCS mode those fields defined as pure DBCS or mixed, SBCS and DBCS, are also translated.

The parameter value can be either Y (Yes) or N (No). The default is **Y**.

AUTTRANS

Is an optional keyword to define automatic start for the data set transmission.

The parameter value can be either Y (Yes) or N (No). The default is **Y**.

If automatic transmission is selected, the forwarding server periodically checks if there is data to be transmitted and automatically starts up. New transactions can be stored while the forwarding is in process. If transmission is stopped by an application, the transaction must be explicitly restarted by the application.

If AUTTRANS=N, the application is responsible for starting and stopping transmissions.

FORWDS vector example

```
/* Vectors FORWPRF and FORWDS Examples */

         FORWPRF  NAME=FORWPRF1,
                  BLKFAC=1,
                  DSPRTY=Y,
                  ALTSESS=Y,
                  S1TYPE=CICS,
                  S1MAXCHR=256,
                  S1RESTO=30,
                  S1RETTO=300,
                  S1TYPDLC=SDLC,
                  S1APPNAM=C1C3C4F1,
                  S2TYPE=IMS,
                  S2MAXCHR=256,
                  S2RESTO=30,
                  S2RETTO=300,
                  S2TYPDLC=SDLC,
                  S2APPNAM=C1C3C2F2

         FORWDS   SESSION=1,
                  TRANSACT=TR01,
                  PRTY=03,
                  EBCDICXL=Y,
                  AUTTRANS=Y

         FORWDS   SESSION=2,
                  TRANSACT=TR02,
                  PRTY=02,
                  EBCDICXL=N,
                  AUTTRANS=Y
```

FORWPRF vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- LANDP for AIX

Defines a forwarding server profile. One vector has to be specified for each profile you are going to use in your workgroups. You can define up to three forwarding sessions in each profile. It may be necessary to define more than one profile, in order to:

- Match the forwarding server profiles with different store-for-forwarding server profiles
- Specify different host application names
- Allow different data link controls (DLCs)
- Establish priorities for different data sets.

This vector must be followed by as many FORWDS vectors as store-for-forwarding data sets have been defined in the SFORWPRF vector that will be used with this profile.

A quick reference

Vector position	None. Can be followed by up to 64 FORWDS vectors.
List of keywords	NAME, BLKFAC, DSPRTY, ALTSESS, S1TYPE, S1MAXCHR, S1RESTO, S1RETTO, S1TYPDLC, S1APPNAM, S1INITS1, S1INITS2, S2TYPE, S2MAXCHR, S2RESTO, S2RETTO, S2TYPDLC, S2APPNAM, S2INITS1, S2INITS2, S3TYPE, S3MAXCHR, S3RESTO, S3RETTO, S3TYPDLC, S3APPNAM, S3INITS1, S3INITS2
Keywords relate to	**Keyword** **Relates to keywords** **NAME** PAR&FORW (parameter 1)

Vector format	FORWPRF NAME=*xxxxxxxx*, [BLKFAC=*xx*,] [DSPRTY=*x*,] [ALTSESS=*x*,] S1TYPE=*xxxx*, [S2TYPE=*xxxx*,] [S3TYPE=*xxxx*,] [S1MAXCHR=*xxx*,] [S2MAXCHR=*xxx*,] [S3MAXCHR=*xxx*,] [S1RESTO=*xxxx*,] [S2RESTO=*xxxx*,] [S3RESTO=*xxxx*,] [S1RETTO=*xxxx*,] [S2RETTO=*xxxx*,] [S3RETTO=*xxxx*,] S1TYPDLC=*xxxxx*, [S2TYPDLC=*xxxxx*,] [S3TYPDLC=*xxxxx*,] S1APPNAM=*xxxxxxxx*, [S2APPNAM=*xxxxxxxx*,] [S3APPNAM=*xxxxxxxx*,] S1INITS1=*xxxxx*, [S2INITS1=*xxxxx*,] [S3INITS1=*xxxxx*,] S1INITS2=*xxxxx*, [S2INITS2=*xxxxx*,] [S3INITS2=*xxxxx*]

Keyword **Description**

NAME Is a required keyword to define the name of the forwarding server profile. The name can have a maximum of eight alphanumeric characters, plus the special characters $, %, #, and @.

BLKFAC Is an optional keyword to define the maximum number of transaction records grouped together in one host transaction.

The parameter value must be a numeric field from 1 to 99. The default is **1**.

If a number other than 1 is specified, the forwarding server automatically blocks the data. A block can contain the records from one store-for-forwarding server data set only. The blocking of records is limited by the available buffer space for building the blocks.

If the number of records to be transmitted is less than the blocking factor, a short block is sent. If records or blocks are longer than 4096 bytes or exceed the maximum size of the chain element specified, the records are sent as a SNA chained message. The host application that receives the forwarded data must be able to receive chained messages and to recover the individual records.

For more information about the block transmission, refer to the *LANDP Servers and System Management* book.

DSPRTY Is an optional keyword to define if you want to assign priorities to each forwarding data set.

The parameter value can be either Y (Yes) or N (No). The default is **Y**.

ALTSESS Is an optional keyword to define the transmission priority for all of the three forwarding sessions. The parameter value can be either Y (Yes) or N (No). Select Y if you want the same priority for each; they will automatically alternate. If you select N, transmission will be in the order of the session numbers (1, 2, 3). The default is **Y**.

In the following keywords, S1 means session 1, S2 session 2, and S3 session 3.

Keyword **Description**

S1TYPE S1TYPE is a required keyword to define the type of host application for session 1. The parameter must be either IMS or CICS. It cannot be omitted.

S2TYPE and S3TYPE
Are optional keywords to define the type of host application programs for sessions 2 and 3. The parameter value can be either IMS, CICS, or N/A.

If S2TYPE is N/A, it deletes the definition of session 2. In this case, you cannot define session 3. The default is **N/A**.

If S3TYPE is N/A, it deletes the definition of session 3. The default is **N/A**.

S1MAXCHR, S2MAXCHR, and S3MAXCHR
Are optional keywords to define the maximum number of bytes sent in a SNA chain element.

The parameter value is a numeric field from 1 to 4096 bytes. If this keyword is omitted, the default is **256**.

This keyword cannot be specified, if the respective session was N/A.

S1RESTO, S2RESTO, and S3RESTO
Are optional keywords to define the time out value for the forwarding server to wait for the SNA session to be established.

The parameter value is a numeric field from 1 to 1500. If this keyword is omitted, the default is **30** seconds.

This keyword cannot be specified, if the respective session was N/A.

S1RETTO, S2RETTO, and S3RETTO
Are optional keywords to define the time period after an unsuccessful attempt to establish a SNA session, before another attempt is made.

The parameter value is a numeric field from 1 to 1500. If this keyword is omitted, the default is **300** seconds.

This keyword cannot be specified, if the respective session was N/A.

S1TYPDLC, S2TYPDLC, and S3TYPDLC

S1TYPDLC, S2TYPDLC, and S3TYPDLC are keywords to define the type of data link control.

Parameter values are SDLC (synchronous), TRDLC (token ring), DCADLC (device cluster adapter), or X25DLC.

This keyword is required for each session that is NOT N/A.

S1APPNAM, S2APPNAM, and S3APPNAM

S1APPNAM, S2APPNAM, and S3APPNAM are keywords to define the name of the host application program for the INITSELF command.

The parameter values are fields with a length from 8 to 16 hexadecimal characters.

The SNA server builds the complete INITSELF string using this name. If this keyword is specified for a session, SxINITSx keyword must be omitted.

Any session that is not N/A requires either APPNAM or SxINITS1 (and SxINITS2). The INITSELF string must be specified using either one or the other.

This keyword cannot be specified, if the respective session was N/A.

S1INITS1, S2INITS1 and S3INITS1

S1INITS1, S2INITS1, and S3INITS1 are keywords to define the first part of the INITSELF string, starting with X'010681' or X'810681'. The length of the complete string must be greater than or equal to 38 digits and less than or equal to 94 digits.

Up to 48 hexadecimal digits can be specified as part 1 of the INITSELF string.

If these keywords are specified for the respective session, SxAPPNAM keywords must be omitted.

This keyword cannot be specified, if the respective session was N/A.

S1INITS2, S2INITS2 and S3INITS2

S1INITS2, S2INITS2, and S3INITS2 are keywords to define the INITSELF string continuation. Up to 46 hexadecimal digits can be specified as part 2 of the INITSELF string.

If these keywords are specified for the session, SxAPPNAM keyword must be omitted.

This keyword cannot be specified, if the respective session was N/A.

FORWPRF vector example

```
/* Vectors FORWPRF and FORWDS Examples */

         FORWPRF  NAME=FORWPRF1,
                  BLKFAC=1,
                  DSPRTY=Y,
                  ALTSESS=Y,
                  S1TYPE=CICS,
                  S1MAXCHR=256,
                  S1RESTO=30,
                  S1RETTO=300,
                  S1TYPDLC=SDLC,
                  S1APPNAM=C1C3C4F1,
                  S2TYPE=IMS,
                  S2MAXCHR=256,
                  S2RESTO=30,
                  S2RETTO=300,
                  S2TYPDLC=SDLC,
                  S2APPNAM=C1C3C2F2

         FORWDS   SESSION=1,
                  TRANSACT=TR01,
                  PRTY=03,
                  EBCDICXL=Y,
                  AUTTRANS=Y

         FORWDS   SESSION=2,
                  TRANSACT=TR02,
                  PRTY=02,
                  EBCDICXL=N,
                  AUTTRANS=Y
```

KBD3270 vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT

Defines a keyboard table for the 3270 Emulator. Specify one vector for each 3270 keyboard table you want to use in your workgroups.

The 3270 keyboard table allows you to associate function codes or data with the SCAN/ASCII code combination generated by every key on the keyboard.

There is a help facility for calling the scan and ASCII codes for each key on your keyboard. From the EHCCUS subdirectory, enter:

```
EHCSC
```

and you will get an interactive help panel where you can enter any key to find out the specified values.

The KBD3270 vector relates to the vector that defines the translation table generated by the keyboard ASCII codes with the EBCDIC host translation, XLATETBL TYPE=AE3270. You associate the two vectors by defining the same value for keyword EXTEN.

There is a limit on the number of keys that can be defined by using the KBD3270 vector. This limit depends on the size of the data strings associated with the keys and is, typically, about 80 or 90 key definitions. If you need to define more keys than this, use the KBD3270X vector to provide the extra definitions.

The following figure contains the scan and ASCII codes for the default key assignments for the 3270 Emulator program functions, when using an English keyboard. This is the default KBD3270 vector (KBD3270 vector with EXTEN=KBD), the source for which can be generated using GENSPEC after a CREATE. You can then change or add values.

Table 10 (Page 1 of 3). Keyboard Key Codes

Scan code	**ASCII code**	**Keyboard key**	**Function**
01	00	Alt+Esc	End emulation
01	1B	Esc	Reset
0E	08	Backspace	Cursor left and delete
0F	00	SHIFT+TAB	Backtab
0F	09	Tab	Tab
10	00	Alt+Q	Alternate cursor
1C	0D	Enter	New line
1E	00	Alt+A	Attention
1F	00	Alt+S	System request
20	00	Alt+D	Dup
21	00	Alt+F	Field mark
2E	00	Alt+C	Cursor selection
3B	00	F1	PF1
3C	00	F2	PF2
3D	00	F3	PF3
3E	00	F4	PF4
3F	00	F5	PF5
40	00	F6	PF6
41	00	F7	PF7
42	00	F8	PF8

Table 10 (Page 2 of 3). Keyboard Key Codes

Scan code	**ASCII code**	**Keyboard key**	**Function**
43	00	F9	PF9
44	00	F10	PF10
47	00	HOME (numeric pad)	Home
47	E0	HOME	Home
48	00	↑ (numeric pad)	Cursor up
48	E0	↑	Cursor up
4A	2D	–	Clear
4B	00	← (numeric pad)	Cursor left
4B	E0	←	Cursor left
4D	00	→ (numeric pad)	Cursor right
4D	E0	→	Cursor right
4E	2B	+	Enter
4F	00	End (numeric pad)	Erase EOF
4F	E0	End	Erase EOF
50	00	↓ (numeric pad)	Cursor down
50	E0	↓	Cursor down
52	00	Insert (numeric pad)	Toggle insert
52	E0	Insert	Toggle insert
53	00	Del (numeric pad)	Delete
53	E0	Del	Delete
54	00	SHIFT+F1	PF13
55	00	SHIFT+F2	PF14
56	00	SHIFT+F3	PF15
57	00	SHIFT+F4	PF16
58	00	SHIFT+F5	PF17
59	00	SHIFT+F6	PF18
5A	00	SHIFT+F7	PF19
5B	00	SHIFT+F8	PF20
5C	00	SHIFT+F9	PF21
5D	00	SHIFT+F10	PF22
68	00	Alt+F1	PA1
69	00	Alt+F2	PA2
6C	00	Alt+F5	End emulation
85	00	F11	PF11

Table 10 (Page 3 of 3). Keyboard Key Codes

Scan code	ASCII code	Keyboard key	Function
86	00	F12	PF12
87	00	SHIFT+F11	PF23
88	00	SHIFT+F12	PF24
E0	0D	Enter (numeric pad)	Enter

You can edit the KBD3270 vector to use the following scan code values for additional emulated 3270 keyboard keys.

Table 11. Emulated Keyboard Key Codes

Scan code	ASCII code	Keyboard key
01	FF	Left CTRL
02	FF	Left ALT
04	FF	Right CTRL
08	FF	Right ALT
10	FF	Scroll Lock
20	FF	Num Lock
40	FF	Caps Lock
80	FF	Sys Req

You can combine these by OR-ing the values together. For example:

```
Left CTRL + Scroll Lock = 11,FF
```

A quick reference

Vector position	None
List of keywords	EXTEN, KEY
Keywords relate to	**Keyword** **Relates to keywords** **EXTEN** EXTEN (XLATETBL vector), SES&3270 (parameter 7), EXTEN (KBD3270X vector)
Vector format	KBD3270 EXTEN=*xxx*, KEY=( )

Keyword	Description
EXTEN	Is a required keyword to give an identification to a 3270 keyboard definition table. The parameter value is a 3-character alphanumeric string, and may contain the special characters #, $, %, and @. It must be unique among all KBD3270 vectors. When defining a 3270 emulator, parameter 7 of the SES&3270 keyword is a pointer to this identification.

KEY Is a required keyword to define keys. There must be one KEY keyword for every key you want to assign function codes or data to.

This keyword has three parameters:

1. The first parameter specifies the scan code for the key. The parameter value is a 1-byte field.
2. The second parameter specifies the ASCII code for the key. The parameter value is a 1-byte field.
3. The third parameter value specifies the associated function code or data for the key.

   Mixing function and data strings is allowed when linking them with a plus (+) sign. The whole parameter is defined between single quotes ('). Data strings must be defined between double quotes ("). Function codes can match functions as described in Table 10 on page 349. The following can also be specified:

   Erase input
   Reset
   Release emulator
   Insert off
   PA3

   For example:

   ```
   'HOME+"data string"+PF4'
   ```

   If single or double quotes are part of a data string, they must be doubled.

KBD3270 vector example

```
/* Vector KBD3270 Default Example */

         KBD3270  EXTEN=KBD,
                  KEY=(48,E0,'CURSOR UP'),
                  KEY=(50,E0,'CURSOR DOWN'),
                  KEY=(4B,E0,'CURSOR LEFT'),
                  KEY=(4D,E0,'CURSOR RIGHT'),
                  KEY=(52,E0,'TOGGLE INSERT'),
                  KEY=(53,E0,'DELETE'),
```

```
KEY=(47,E0,'HOME'),
KEY=(4F,E0,'ERASE EOF'),
KEY=(48,00,'CURSOR UP'),
KEY=(50,00,'CURSOR DOWN'),
KEY=(4B,00,'CURSOR LEFT'),
KEY=(4D,00,'CURSOR RIGHT'),
KEY=(52,00,'TOGGLE INSERT'),
KEY=(53,00,'DELETE'),
KEY=(47,00,'HOME'),
KEY=(4F,00,'ERASE EOF'),
KEY=(1C,0D,'NEW LINE'),
KEY=(E0,0D,'ENTER'),
KEY=(4E,2B,'ENTER'),
KEY=(4A,2D,'CLEAR'),
KEY=(01,1B,'RESET'),
KEY=(01,00,'END EMULATION'),
KEY=(6C,00,'END EMULATION'),
KEY=(0F,09,'TAB'),
KEY=(0F,00,'BACKTAB'),
KEY=(0E,08,'CURSOR LEFT+DELETE'),
KEY=(3B,00,'PF1'),
KEY=(3C,00,'PF2'),
KEY=(3D,00,'PF3'),
KEY=(3E,00,'PF4'),
KEY=(3F,00,'PF5'),
KEY=(40,00,'PF6'),
KEY=(41,00,'PF7'),
KEY=(42,00,'PF8'),
KEY=(43,00,'PF9'),
KEY=(44,00,'PF10'),
KEY=(85,00,'PF11'),
KEY=(86,00,'PF12'),
KEY=(55,00,'PF14'),
KEY=(56,00,'PF15'),
KEY=(57,00,'PF16'),
KEY=(58,00,'PF17'),
KEY=(59,00,'PF18'),
KEY=(5A,00,'PF19'),
KEY=(5B,00,'PF20'),
KEY=(5C,00,'PF21'),
KEY=(5D,00,'PF22'),
KEY=(87,00,'PF23'),
KEY=(88,00,'PF24'),
KEY=(68,00,'PA1'),
KEY=(69,00,'PA2'),
KEY=(20,00,'DUP'),
KEY=(21,00,'FIELD MARK'),
KEY=(10,00,'ALTERNATE CURSOR'),
KEY=(2E,00,'CURSOR SELECTION'),
KEY=(1F,00,'SYSTEM REQUEST'),
KEY=(54,00,'PF13'),
KEY=(1E,00,'ATTENTION')
```

KBD3270X vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT

Defines additional keys for a keyboard table for the 3270 emulator. Specify one vector for each of your 3270 keyboard tables that need more key definitions.

When you need more key definitions for a keyboard table, specify a KBD3270X vector with the same value for its EXTEN keyword as you gave to the table that requires the additional keys.

A quick reference

Vector position	None
List of keywords	EXTEN, KEY
Keywords relate to	**Keyword** **Relates to keywords** **EXTEN** EXTEN (KBD3270 vector), EXTEN (XLATETBL vector) SES&3270 (parameter 7)
Vector format	KBD3270X EXTEN=*xxx*, KEY=( )

Keyword **Description**

EXTEN Is a required keyword to give an identification to a 3270 keyboard definition table. A KBD3270 vector providing the first half of the keyboard definition table must also exist in the COMMON.SPC file.

The parameter value is a 3-character alphanumeric string, and may contain the special characters #, $, %, and @. It must match the EXTEN value of a KBD3270 vector.

When defining a 3270 emulator, parameter 7 of the SES&3270 keyword is a pointer to this identification.

KEY Is a required keyword to define keys. There must be one KEY keyword for every key you want to assign function codes or data to.

This keyword has three parameters:

1. The first parameter specifies the scan code for the key. The parameter value is a 1-byte field.
2. The second parameter specifies the ASCII code for the key. The parameter value is a 1-byte field.
3. The third parameter value specifies the associated function code or data for the key.

   Mixing function and data strings is allowed when linking them with a plus (+) sign. The whole parameter is defined between single quotes

('). Data strings must be defined between double quotes ("). Function codes can match functions as described in Table 10 on page 349. The following can also be specified:

Erase input
Reset
Release emulator
Insert off
PA3

For example:

```
'HOME+"data string"+PF4'
```

If single or double quotes are part of a data string, they must be doubled.

KBD3270X vector example

```
/* Vector KBD3270X Example */

         KBD3270X EXTEN=KBD,
                  KEY=(1E,00,'HOME+"HOST1"+ENTER'),
                  KEY=(3E,00,'HOME+"HOST2"+ENTER')
```

KBDBIWP vector

- PC/Integrator
- PC Integrator/2

Defines a keyboard table for the Banking Interactive Workstation Program (BIWP). Specify one vector for each BIWP keyboard table you want to use in your workgroups.

The BIWP keyboard table allows you to associate function codes or data with the SCAN/ASCII code combination generated by every key on the keyboard.

There is a help facility for calling the scan and ASCII codes for each key on your keyboard. From the EHCCUS subdirectory, enter:

```
EHCSC
```

and you will get an interactive help panel where you can enter any key to find out the specified values.

The keyboard definition vector is associated with three other vectors. The first two are optional, the third is required.

1. The vector that defines the table used to simulate a keystroke sent from keyboard to 4700 (KSTRBIWP vector).
2. The vector that defines the table used to simulate a scan code sent from keyboard to 4700 (KSCCBIWP vector).

3. The vector that defines the table to translate the ASCII codes generated by the keyboard with EBCDIC sent to 4700 (XLATETBL TYPE=AEBIWP).

You associate the four vectors by defining the same value for the keyword EXTEN.

The keyboard table used by the BIWP can be changed from the 4700 application with the SWAPTT instructions. When the active keyboard table is changed, the active tables for all the group of vectors with the same EXTEN keyword will also be changed accordingly.

The following figure shows the defaults provided for the default keyboard table. You can change or add values.

Table 12 (Page 1 of 2). Keyboard Key Defaults

Function	**Scan/ASCII Codes**	**Keyboard key**
JUMP (see notes below the table)	01 00 (see notes below the table)	[Alt] + [Esc]
RESET	01 1B	[Esc]
BACKSPACE + SEG DELETE	0E 08	[Backspace]
MSR/E CANCEL	12 00	[Alt] + [E]
PIN pad CANCEL	19 00	[Alt] + [P]
EOM, X'FF'	1C 0D	[Return]
CANCEL	2E 00	[Alt] + [C]
MONITOR	32 00	[Alt] + [M]
BACKSPACE	4B 00	[Left arrow]
BACKSPACE	4B E0 (see notes below the table)	[Left arrow]
ADVANCE	4D 00	[Right arrow]
ADVANCE	4D E0 (see notes below the table)	[Right arrow]
EOM, X'FF'	4E 2B	[+]
END OF INPUT	4F 00	[End]
END OF INPUT	4F E0 (see notes below the table)	[End]
TOG.SEG. INSERT	52 00	[Ins]
TOG.SEG. INSERT	52 E0 (see notes below the table)	[Ins]
SEGMENT DELETE	53 00	[Del]
SEGMENT DELETE	53 E0 (see notes below the table)	[Del]
JUMP (see notes below the table)	6C 00	[Alt] + [F5]
JUMP (see notes below the table)	6D 00	[Alt] + [F6]

Table 12 (Page 2 of 2). Keyboard Key Defaults

Function	Scan/ASCII Codes	Keyboard key
JUMP (see notes below the table)	6E 00	[Alt] + [F7]
JUMP (see notes below the table)	6F 00	[Alt] + [F8]
JUMP (see notes below the table)	70 00	[Alt] + [F9]
ERASE END OF SEG.	75 E0 (see notes below the table)	[Ctrl] + [End]
EOM, X'FF'	E0 0D (see notes below the table)	[Enter]

Notes:

1. 101/102-key keyboard extended function.
2. The JUMP function is provided for DOS only. In OS/2, Task Manager handles this function. The key combinations defined for the JUMP function are not operational under OS/2.

A quick reference

Vector position	None
List of keywords	EXTEN, APPLIC, EAEXTEN, KEY
Keywords relate to	**Keyword** **Relates to keywords** **EXTEN** EXTEN (KSCCBIWP, KSTRBIWP, and XLATETBL vectors), PAR&BIWP (parameter 14) **EAEXTEN** EXTEN (XLATETBL vector)
Vector format	KBDBIWP EXTEN=*xxx*, [APPLIC=*xxxxxxxx*,] [EAEXTEN=*xxx*,] KEY=( )

Keyword **Description**

EXTEN Is a required keyword to give an identification to a BIWP keyboard definition table.

The parameter value is a 3-character alphanumeric string, and may contain the special characters #, $, %, and @. It must be unique among all KBDBIWP vectors.

The parameter value cannot be UNI, since it is a reserved name.

When defining a BIWP emulator, parameter 18 of the SES&BIWP keyword is a pointer to this identification.

APPLIC Is an optional keyword that defines the application identification that will be used with the 4700 SWAPTT instruction.

The parameter value can have a maximum of eight alphanumeric characters, plus the special characters $, %, #, and @. It must be unique among all KBDBIWP vectors.

It will be translated to EBCDIC using the international code pages 850 and 500.

EAEXTEN Is an optional keyword to define the extension of the EABIWP table that is included in the group specified with the APPLIC keyword. This table, together with the KSTRBIWP and KSCCBIWP vectors, will be changed by the 4700 SWAPTT instruction.

If you want to change this table, you must define it.

The parameter value is a 3-character string and must adhere to DOS specifications for file extensions. It must match the parameter of the EXTEN keyword in a XLATETBL vector (TYPE=EABIWP).

KEY Is a required keyword to define keys. There must be one KEY keyword for every key definition.

This keyword has three parameters:

1. The first parameter specifies the scan code for the key. The parameter value is a two hexadecimal field.
2. The second parameter specifies the ASCII code for the key. The parameter value is a two hexadecimal field.
3. The third parameter value specifies the associated function code or data for the key.

   The whole parameter is defined between single quotes ('). Mixing function code and data is allowed when linking them by using a plus (+) sign.

   ASCII data strings are defined between double quotes ("). Hexadecimal or EBCDIC data strings are defined between double quotes and are preceded by a colon (:). They will be a group of hexadecimal pairs with the EBCDIC values.

   Function codes can match functions as described in Table 12 on page 356.

   The following can also be specified:

   EOMA
   EOMB
   EOM, EID
   SEG. INSERT OFF
   NOP

   If single or double quotes or colons are part of a data string, they must be doubled.

When defining the function codes for the functions

EOM
EOMA
EOMB
EOM, EID

the function code must be followed by a plus (+) sign and two hexadecimal digits corresponding to the EOM mask.

Note that when defining the function code for the EOM, EID function, you can specify only one byte. As SMSEID is a 1-byte field, the customization program requires that you enter a 1-byte character string before assigning the EOM, EID function to a key. This 1-byte character string is placed in SMSEID. To generate more than one character, you must define two concatenated character strings: one with the byte to be placed in SMSEID, and one with the rest of the characters.

For example, to define an EOM, EID key that generates EBCDIC *C2* in the SMSEID field and EBCDIC *C1* in the user segment, with the EOM mask being *FF*, the value for the third parameter must be:

```
'":C1"+":C2". +,EOM,EID+FF‘
```

KBDBIWP vector example

```
/* Vector KBDBIWP Default Example */

         KBDBIWP  EXTEN=KBD,
                  APPLIC=DEFAULT,
                  EAEXTEN=DIS,
                  KEY=(01,1B,'RESET'),
                  KEY=(70,00,'JUMP'),
                  KEY=(2E,00,'CANCEL'),
                  KEY=(32,00,'MONITOR'),
                  KEY=(E0,0D,'EOM+FF'),
                  KEY=(1C,0D,'EOM+FF'),
                  KEY=(4B,00,'BACKSPACE'),
                  KEY=(4D,00,'ADVANCE'),
                  KEY=(4D,E0,'ADVANCE'),
                  KEY=(4B,E0,'BACKSPACE'),
                  KEY=(53,00,'SEG. DELETE'),
                  KEY=(52,00,'TOG.SEG. INSERT'),
                  KEY=(53,E0,'SEG. DELETE'),
                  KEY=(52,E0,'TOG.SEG. INSERT'),
                  KEY=(0E,08,'BACKSPACE+SEG. DELETE'),
                  KEY=(19,00,'PIN PAD CANCEL'),
                  KEY=(12,00,'MSR/E CANCEL'),
                  KEY=(4F,E0,'END OF INPUT'),
                  KEY=(75,E0,'ERASE END OF SEG'),
                  KEY=(4E,2B,'EOM+FF'),
                  KEY=(4F,00,'END OF INPUT'),
                  KEY=(01,00,'JUMP'),
                    KEY=(6C,00,'JUMP'),
                    KEY=(6D,00,'JUMP'),
                    KEY=(6E,00,'JUMP'),
                    KEY=(6F,00,'JUMP')
```

KSCCBIWP vector

- PC/Integrator
- PC Integrator/2

Defines an issue-keyboard-scan code translation table for the Banking Interactive Workstation Program.

This vector is used to match scan codes and their associated shift states, which are used by the application when using the BIWP issue-keyboard-scan code function with the keyboard definition vector (KBDBIWP). You associate a KSCCBIWP vector with a KBDBIWP vector by defining the same value for the keyword EXTEN in both vectors.

Define as many KSCCBIWP vectors as different BIWP keyboard definition vectors you are going to use through the issue-keyboard-scan code function.

If this vector is omitted, an empty table will be created.

A quick reference

Vector position	None
List of keywords	EXTEN, KEY
Keywords relate to	**Keyword** **Relates to keywords** **EXTEN** EXTEN (KBDBIWP, KSTRBIWP, and XLATETBL vectors)
Vector format	KSCCBIWP EXTEN=*xxx*, KEY=( )

Keyword **Description**

EXTEN Is a required keyword to give an identification to a BIWP issue-keyboard-scan code translation table.

The parameter value is a 3-character alphanumeric string, and may contain the special characters #, $, %, and @. It must be the same as defined in the associated KBDBIWP vector.

KEY Is a required keyword for scan code definition. There must be one keyword for every shift state/scan code combination defined.

The keyword has four parameters.

The first and second parameters define the shift state and scan code values that will be used by the application when using the BIWP issue-keyboard-scan code function. The shift state/scan code combination must be unique in the table.

- The first parameter defines the 1 hexadecimal byte character shift state used by the application. Its values can be:

  U For not shifted
  S For shifted
  A For alternate

- The second parameter defines the 1-byte scan code value used by the application.

The third and fourth parameters define the SCAN and ASCII codes of a key defined in the associated KBDBIWP vector.

- The third parameter is a SCAN code of the associated key defined in the KBDBIWP vector.
- The fourth parameter is the ASCII code of the associated key defined in the KBDBIWP vector.

KSCCBIWP vector example

```
/* Vector KSCCBIWP Example */

         KSCCBIWP EXTEN=KBD,
                  KEY=(S,06,4D,E0),
                  KEY=(U,32,4B,00),
                  KEY=(U,33,E0,0D),
                  KEY=(U,34,1C,0D)
```

KSTRBIWP vector

- PC/Integrator
- PC Integrator/2

Defines a present-keystroke translation table for banking interactive workstation program.

This vector is used to define how the keystrokes, used by the application when using the present-keystroke function, match with the keyboard definition vector (KBDBIWP). You associate a KSTRBIWP vector with a KBDBIWP vector by defining the same value for the keyword EXTEN in both vectors. Specify one vector for each BIWP keyboard definition vector you are going to use through the present-keystroke function.

If the KSTRBIWP vector is omitted, an empty table will be generated.

The following figure shows the values defined in the default KSTRBIWP vector (KSTRBIWP vector with keyword EXTEN=KBD).

Table 13. Present-keystroke defaults

Function	Scan/ASCII Codes	Keystroke Values
EOM, FF	1C 0D	0D
RESET	01 1B	00
BACKSPACE	4B 00	03
ADVANCE	4D 00	04
BACKSPACE + SEGMENT DELETE	0E 08	08

A quick reference

Vector position	None
List of keywords	EXTEN, KEY
Keywords relate to	**Keyword** **Relates to keywords** **EXTEN** EXTEN (KBDBIWP, KSCCBIWP, and XLATETBL vectors)
Vector format	KSTRBIWP EXTEN=*xxx*, KEY=( )

Keyword **Description**

EXTEN Is a required keyword to identify a BIWP present-keystroke table.

The parameter value is a 3-character alphanumeric string, and may contain the special characters #, $, %, and @. It must be the same as defined in the associated KBDBIWP vector, in the EXTEN keyword.

KEY Is a required keyword to define a keystroke. There must be one KEY keyword for every keystroke defined.

The keyword has three parameters:

1. The first parameter defines the 1-byte value that will be used by the application with the BIWP present-keystroke function. It must be unique in the table.
2. The second parameter is the 1-byte SCAN code of the associated key defined in the KBDBIWP vector.
3. The third parameter is the 1-byte ASCII code of the associated key defined in the KBDBIWP vector.

KSTRBIWP vector example

```
/* Vector KSTRBIWP  Default Example */

         KSTRBIWP EXTEN=KBD,
                  KEY=(0D,1C,0D),
                  KEY=(03,4B,00),
                  KEY=(04,4D,00),
                  KEY=(08,0E,08),
                  KEY=(00,01,1B)
```

LUPOOL vector

- LANDP for DOS

Defines LUs for host communication using the SNA server with LU pooling on a DOS workstation. One vector is required for each LU to be pooled. All the LUs to be used by the SNA server must be in the same LUPOOL table - in other words, they must have the same NAME parameter value.

A quick reference

Vector position	None
List of keywords	NAME, GROUPID, DLC, LUNUMBER
Keywords relate to	**Keyword** **Relates to keywords** **NAME** PAR&SNA **GROUPID** SES&SNA

Vector format	LUPOOL NAME=*xxxxxxxx*,
	GROUPID=*xx*,
	DLC=*xxx*,
	LUNUMBER=*xxx*

Keyword **Description**

NAME Is a required keyword to specify the name of the LUPOOL table.

The parameter value is a string of up to eight alphanumeric characters and may contain the special characters #, _, %, or @.

GROUPID Is a required keyword to identify the logical group of LUs.

The parameter value is a string of two characters: the first character must be alphabetical, and the second one numerical.

DLC Is a required keyword to specify the DLC type used with the LU.

For DLCs other than X.25, the parameter value can be:

SDL Synchronous Data Link Control (SDLC)
TKR Token-ring
DCA Device Cluster Attachment (DCA)

For X.25 DLCs, the parameter value ranges from 1 to 999, and corresponds to the virtual circuit definition identification.

LUNUMBER

Is a required keyword to specify the LU number. This corresponds to the LOCADDR on the host VTAM definition.

The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit.

LUPOOL vector example

```
/* Vector LUPOOL Example */

         LUPOOL   NAME=LUPOOL1,
                  GROUPID=P1,
                  DLC=SDL,
                  LUNUMBER=1
```

MSRINTBL vector

- PC/Integrator
- PC Integrator/2

Defines an input translation table for the Banking Interactive Workstation Program when reading the MSR/E. Define one MSRINTBL vector for every MSR/E input table that you want to use in your workgroups.

A quick reference

Vector position	None
List of keywords	EXTEN, TRKOPER, DELAY, TR2CHR0 to TR2CHRF, TR3CHR0 to TR3CHRF
Keywords relate to	**Keyword** **Relates to keywords** **EXTEN** PAR&BIWP (parameter 17)
Vector format	MSRINTBL EXTEN=*xxx*, [TRKOPER=( ),] [DELAY=*xx*,] [TR2CHR0=( ),] [TR3CHR0=( )]

Keyword **Description**

EXTEN Is a required keyword to give an identification to an input translation table for the MSR/E.

The parameter value is a 3-character alphanumeric string, and may contain the special characters #, $, %, and @. It must be unique among all MSRINTBL vectors.

When defining a BIWP server, one of the parameters of the PAR&BIWP keyword is a pointer to this identification.

TRKOPER Is an optional keyword to define the operational MSR/E tracks.

The keyword has two parameters:

1. The first parameter specifies which tracks are going to be read. The possible values are:

   2 Only track 2 will be read
   3 Only track 3 will be read
   B Both tracks 2 and 3 will be read

   The default is **2**.

2. The second parameter specifies whether track 2 will be written. The parameter value can be either Y (Yes) or N (No). The default is **Y**.

DELAY Is an optional keyword to specify the maximum amount of time, in seconds, between MSR/E reads.

The parameter value ranges from 1 to 60. The default is **5**.

TR2CHR0 to TR2CHRF
Are optional keywords to associate data or end of message indicators to every character (0 to F), read from track 2. TR2CHR0 is the keyword related to track 2 character 0, TR2CHR1 to track 2 character 1, and so on. You must write the corresponding TR2CHRx for every character in track 2 that you want to generate data or EOM indicator. If the keyword is omitted for a character in track 2, no data or EOM indicator will be sent to the 4700 when it is read.

Each keyword has two parameters:

1. The first parameter specifies the 1-byte value to be sent. It must be enclosed between double quotes ("). If the character is an EBCDIC value, it must be preceded by a colon (:), and written in hexadecimal notation.

   The default is no defined character.

2. The second parameter specifies the end of message indicator associated with the character, and takes the following values:

EOM	If character is associated with EOM
CEOM	If character is associated with conditional EOM
EID	If character is associated with EID
CEID	If character is associated with conditional EID
NONE	If character is not associated with any indicator

   The default is **NONE**.

TR3CHR0 to TR3CHRF
Are optional keywords associated with track 3. If specified, they will follow the same rules as TR2CHR0 to TR2CHRF.

MSRINTBL vector example

```
/* Vector MSRINTBL  Default Example */
        MSRINTBL EXTEN=MSR,
                TRKOPER=(2,Y),
                TR2CHR0=(":F0",NONE),
                TR2CHR1=(":F1",NONE),
                TR2CHR2=(":F2",NONE),
                TR2CHR3=(":F3",NONE),
                TR2CHR4=(":F4",NONE),
                TR2CHR5=(":F5",NONE),
                TR2CHR6=(":F6",NONE),
                TR2CHR7=(":F7",NONE),
                TR2CHR8=(":F8",NONE),
                TR2CHR9=(":F9",NONE),
                TR2CHRC=(":7C",EOM),
                TR2CHRD=(":7D",NONE),
                TR2CHRE=(":7E",NONE),
                TR3CHR0=(":F0",NONE),
                TR3CHR1=(":F1",NONE),
                TR3CHR2=(":F2",NONE),
                TR3CHR3=(":F3",NONE),
                TR3CHR4=(":F4",NONE),
                TR3CHR5=(":F5",NONE),
                TR3CHR6=(":F6",NONE),
                TR3CHR7=(":F7",NONE),
                TR3CHR8=(":F8",NONE),
                TR3CHR9=(":F9",NONE),
                TR3CHRA=(":FA",NONE),
                TR3CHRC=(":7C",NONE),
                TR3CHRE=(":7E",NONE)
```

MSROUTBL vector

- PC/Integrator
- PC Integrator/2

Defines an output translation table for the Banking Interactive Workstation Program when writing to the MSR/E. Define one MSROUTBL vector for every MSR/E output table that you want to use in your workgroups.

A quick reference

Vector position	None
List of keywords	EXTEN, OPTIONS, OUTCHR0 to OUTCHRF
Keywords relate to	**Keyword** **Relates to keywords** **EXTEN** PAR&BIWP (parameter 18)
Vector format	MSROUTBL EXTEN=*xxx*, [OPTIONS=( ),] [OUTCHR0=( )]

Keyword **Description**

EXTEN
Is a required keyword to define an output table for the MSR/E.

The parameter value is a 3-character alphanumeric string, and may contain the special characters #, $, %, and @. It must be unique among all MSROUTBL vectors.

When defining a BIWP, one of the parameters of the PAR&BIWP keyword is a pointer to this identification.

OPTIONS
Is an optional keyword to define parameters to perform output operations. If it is omitted, defaults will be assumed. The keyword has three parameters:

1. The first parameter specifies the encoding mode. It can take the values DIN, IBM, or ISO. The default is **IBM**.
2. The second parameter specifies whether the track will be written using single (1) or double (2) record encoding. The default is **2**.
3. The third parameter specifies whether or not to use the suppress heading A option with the IBM encoding mode.

   The parameter value can be either Y (Yes) or N (No). The default is **N**.

OUTCHR0 to OUTCHRF
Are optional keywords to associate the characters received from the 4700 application with the characters that will be encoded to the MSR/E (0 to F). OUTCHR0 is the keyword related to character 0, OUTCHR1 to character 1, and so on.

You must write the corresponding OUTCHRx for every character you want to be written to the MSR/E. If the keyword is omitted for a character, it will not be written. Each keyword has three parameters:

1. The first parameter specifies the 1-byte value that will be received from the 4700 application. The value must be a pair of hexadecimal digits and must be unique in the vector.
2. The second parameter specifies whether the character is operational or not. The parameter value must be either Y (the character will be translated and encoded) or N (the character will be ignored). The default is **Y**.
3. The third parameter specifies whether the character is associated with an EOM indicator or not. The parameter value can be either Y (Yes) or N (No). The default is **N**.

MSROUTBL vector example

```
/* Vector MSROUTBL  Default Example */

         MSROUTBL EXTEN=MSR,
                  OPTIONS=(IBM,2,N),
                  OUTCHR0=(F0,Y,N),
                  OUTCHR1=(F1,Y,N),
                  OUTCHR2=(F2,Y,N),
                  OUTCHR3=(F3,Y,N),
                  OUTCHR4=(F4,Y,N),
                  OUTCHR5=(F5,Y,N),
                  OUTCHR6=(F6,Y,N),
                  OUTCHR7=(F7,Y,N),
                  OUTCHR8=(F8,Y,N),
                  OUTCHR9=(F9,Y,N),
                  OUTCHRA=(00,N,N),
                  OUTCHRB=(00,N,N),
                  OUTCHRC=(7C,Y,Y),
                  OUTCHRD=(7D,Y,N),
                  OUTCHRE=(7E,Y,N),
                  OUTCHRF=(00,N,N)
```

P3287ATT vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT

Defines an attribute table for the 3287 emulator.

One vector has to be specified for each 3287 emulator attribute table that you want to use in your workgroups.

A quick reference

Vector position	None
List of keywords	EXTEN, TIMEOUT, LSTDELIM, DEFCCHR, RESET
Keywords relate to	**Keyword** **Relates to keywords** **EXTEN** PAR&3287 (parameter 3)
Vector format	P3287ATT EXTEN=*xxx*, [TIMEOUT=*xxxx*,] [LSTDELIM=*xxxxxxxx*,] [DEFCCHR=*xx*,] [RESET=( )]

Keyword **Description**

EXTEN Is a required keyword to give an identification to a 3287 attribute table, if more than one table is used. Otherwise it can be omitted and the default is **TAB**. The keyword does not apply to DBCS mode.

The parameter value is a 3-character alphanumeric string, and may contain the special characters #, $, %, and @. It must be unique among all P3287ATT vectors.

When defining a 3287 printer emulator, one of the parameters of the PAR&3287 keyword is a pointer to this identification.

TIMEOUT Is an optional keyword to define the number of seconds the 3287 printer emulator will wait before assuming that a listing has finished. At the end of the time interval, the emulator gives control to the next remote service or changes from local to remote (or remote to local) depending on operator's requirements. If the time-out expires without a print request having arrived, a listing is considered to have ended.

The parameter value is a field with up to four characters and can take the values between 0 and 3600. The default is **15** seconds.

LSTDELIM Is an optional keyword to define a name that the host application program provides to identify the beginning and the end of listings.

The parameter value is a string of up to eight alphanumeric characters.

The application sends a LISTL (LISTL=*xxxxxxxx*) that identifies the beginning of the identified listing. The emulator program considers the listing to be finished when the corresponding ENDL (ENDL=*xxxxxxxx*) is received. Normally, the application sends a form feed control character following the ENDL. If the ENDL is not received, the following happens:

1. If the operator interface is loaded, the 3287 printer emulator program waits for a command from the operator.
2. If the operator interface is not loaded, the emulator assumes that the listing has finished at the specified timeout.

DEFCCHR Is an optional keyword to define the 1-byte ASCII value of the character that replaces any invalid control character. The default is **00**.

RESET Is an optional keyword to define the control characters used to reset the printer before starting a new listing.

The parameter value is a field of up to 45 hexadecimal bytes. The hexadecimal bytes must be separated with a comma (,).

You only need to define this keyword if you use a printer that uses different reset control characters from those for the printers explicitly supported by the 3287 emulator.

P3287ATT vector example

```
/* Vector P3287ATT  Default Example */

         P3287ATT EXTEN=TAB,
                  TIMEOUT=15,
                  DEFCCHR=00
```

PINPTBL vector

- PC/Integrator
- PC Integrator/2

Defines an input translation table to be used by BIWP when reading the PIN pad. Define one PINPTBL vector for every PIN pad input table that you want to use in your workgroups.

A quick reference

Vector position	None
List of keywords	EXTEN, FILLER, KEY0 to KEY9, KEYSCP, KEYSEP, KEYENDP
Keywords relate to	**Keyword** **Relates to keywords** **EXTEN** PAR&BIWP (parameter 16)
Vector format	PINPTBL EXTEN=*xxx*, [FILLER=*x*,] [KEY0=( ),] [KEYSCP=( ),] [KEYSEP=( ),] [KEYENDP=( )]

Keyword **Description**

EXTEN Is a required keyword to define a PIN pad input table.

The parameter value is a 3-character alphanumeric string, and may contain the special characters #, $, %, and @. It must be unique among all MSROUTBL vectors.

When defining a BIWP, one of the parameters of the PAR&BIWP keyword is a pointer to this identification.

FILLER Is an optional keyword to define the ASCII character that will appear on the screen when any PIN pad key is pressed.

The parameter value is one ASCII character. Special characters have to be put between single quotes (''). The default is an asterisk (*****).

KEY0 to KEY9

Are optional keywords to associate the data or end of message indicators to every character (0 to 9), read from PIN pad. KEY0 is the keyword related to character 0, KEY1 to character 1, and so on.

You must write the corresponding KEYx for every character in PIN pad for which you want to generate data or EOM indicator. If the keyword is omitted for a character, no data will be sent to the 4700 when this character is read. Each keyword has three parameters:

1. The first parameter specifies the character that will be sent to the 4700 when the key is pressed. This parameter is required, if the keyword has been specified.

   The value must be a 2-byte digit and must be unique in the vector.

2. The second parameter specifies whether the key is operational or not. The parameter value must be either H (the key is operational and hidden) or N (the key is not operational). The default is **N**.

3. The third parameter specifies whether the key is associated with an EOM indicator or not. The parameter value can be either Y (Yes) or N (No). The default is **N**.

KEYSCP Is an optional keyword to define the 2-byte value of the character to be sent to the 4700 when the start of the clear PIN indicator is read from the PIN pad. You can also associate an EOM indicator to it.

If the keyword is omitted, the defaults will be assumed.

The keyword has three parameters:

1. The first parameter specifies the character that will be sent to the 4700 when the start of the clear PIN indicator is read.

   The value must be a 2-byte digit and must be unique in the vector.

   The default is **7F**.

2. The second parameter specifies whether the key is operational or not. The parameter value must be either H (the key is operational and hidden) or N (the key is not operational). The default is **N**.

3. The third parameter specifies whether the key is associated with an EOM indicator or not. The parameter value can be either Y (Yes) or N (No). The default is **N**.

KEYSEP Is an optional keyword to define the 2-byte value of the character to be sent to the 4700 when the start of the encrypted PIN indicator is read from the PIN pad. You can also associate an EOM indicator to it.

If the keyword is omitted, the defaults will be assumed.

The keyword has three parameters:

1. The first parameter specifies the character that will be sent to the 4700 when the start of the encrypted PIN indicator is read.

   The value must be a 2-byte digit and must be unique in the vector.

   The default is **7E**.

2. The second parameter specifies whether the key is operational or not. The parameter value must be either H (the key is operational and hidden) or N (the key is not operational). The default is **N**.
3. The third parameter specifies whether the key is associated with an EOM indicator or not. The parameter value can be either Y (Yes) or N (No). The default is **N**.

KEYENDP Is an optional keyword to define the 2-byte value of the character to be sent to the 4700 when the end of PIN indicator is read from the PIN pad. You can also associate an EOM indicator to it.

If the keyword is omitted, the defaults will be assumed.

The keyword has three parameters:

1. The first parameter specifies the character that will be sent to the 4700 when the end of PIN indicator is read.

   The value must be a 2-byte digit and must be unique in the vector.

   The default is **7F**.

2. The second parameter specifies whether the key is operational or not. The parameter value must be either H (the key is operational and hidden) or N (the key is not operational). The default is **N**.
3. The third parameter specifies whether the key is associated with an EOM indicator or not. The parameter value can be either Y (Yes) or N (No). The default is **N**.

PINPTBL vector example

```
/* Vector PINPTBL  Default Example */

         PINPTBL  EXTEN=PIN,
                  FILLER=*,
                  KEY0=(F0,H,N),
                  KEY1=(F1,H,N),
                  KEY2=(F2,H,N),
                  KEY3=(F3,H,N),
                  KEY4=(F4,H,N),
                  KEY5=(F5,H,N),
                  KEY6=(F6,H,N),
                  KEY7=(F7,H,N),
                  KEY8=(F8,H,N),
                  KEY9=(F9,H,N),
                  KEYSCP=(7F,H,N),
                  KEYSEP=(7E,H,N),
                  KEYENDP=(7F,H,N)
```

RCMSLNF vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT

Defines remote change management services (RCMS) logical names according to NetView DM conventions.

RCMS is primarily concerned with activities between the host and workstation, such as storing files sent from the host to the workstation, and sending files from the workstation to the host.

Because the host and workstation use different code formats, RCMS needs an EBCDIC-to-ASCII translation table to convert the strings received from the host, and an ASCII-to-EBCDIC table for messages to the host. Both translation tables are defined in the XLATETBL vector for types AERCMS and EARCMS.

You also need a logical name, relating the file identification in the host to the path and file name in the workstation.

Files that are visible to the host and can be distributed by Netview Distribution Manager (DM) must follow predefined conventions. Editing the RCMS logical name file allows you to add, delete, and update host-to-workstation filename mapping. Thus, Netview DM can manage and control distribution of software and data resources. Identical logical names and workstation file paths are not permitted.

One vector has to be defined for each RCMS logical name you want to specify.

A quick reference

Vector position	None
List of keywords	NAME, PATH
Vector format	RCMSLNF NAME=*xxxxxxxx*, PATH=*path*

Keyword **Description**

NAME Is a required keyword to define the logical name that will be used in the host to identify a file or group of files in workgroup workstations.

The parameter value is a string of up to eight characters starting with an alphabetical character. It must be unique among all RCMS logical names.

PATH Is a required keyword to define the path and file name in the workstation.

The parameter value is a string of up to 64 characters and must adhere to DOS specifications for path and file names. It must be unique among all RCMLNF vectors.

RCMSLNF vector example

```
/* Vector RCMSLNF Example */

         RCMSLNF  NAME=NAME001,
                  PATH=C:\LEVEL1\LEVEL2\LEVEL3\
```

For more information on RCMS see *LANDP Servers and System Management*.

RECDEF vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- LANDP for AIX

Defines a record structure. Every record you define has a name and can contain several fields. Every field in the record has a name and associated parameters to enable validation when the data is updated. Up to 92 fields may be defined in the RECFIELD vector.

The record definition facility supports the following record formats:

- Fixed format. All data fields in the record are of fixed length.
- Delimiter format. The length of at least one field is variable, and defined through a delimiter.

The record definition facility defines the record structures used by the following LANDP family servers:

- Electronic journal server
- Store-for-forwarding server
- Forwarding server
- System manager server
- System manager operator
- Shared-file server

The applications and the user servers can also use the record structures defined with this vector.

LANDP for AIX electronic journal and store-for-forwarding servers use the record structures defined with LANDP for AIX record definition utility.

A quick reference

Vector position	None. Can be followed by up to 92 RECFIELD vectors.
List of keywords	NAME, DELIMIT, DECSEP
Keywords relate to	**Keyword** **Relates to keywords** **NAME** RECNAME (EJOUREC, SHFLDBD, and SFORWREC vectors)
Vector format	RECDEF NAME=*xxxxxxxx*, [DELIMIT=*x*,] [DECSEP=*x*]

Keyword **Description**

NAME Is a required keyword to define the name of a record structure.

The parameter value is a string with up to eight alphanumeric characters. It must be unique for all the RECDEF vectors.

This name can be referenced by the electronic journal server (EJOUREC vector), the shared-file server (SHFLDBD vector), or the store-for-forwarding server (SFORWREC vector).

DELIMIT Is an optional keyword to define the character used to mark the end of a variable length field. A delimiter is required if the field is variable length.

Parameter values are the special characters \, /, $, @, &, and %. They must be enclosed between single quotes. The default is **'/'**.

If you specify pure DBCS or mixed, SBCS and DBCS, field format in an associate RECFIELD vector, characters \ and @ are not supported as delimiters.

DECSEP Is an optional keyword to define the decimal separator, that is, the character to separate the integer part of a numeral from the fractional (decimal) part.

The parameter value can either be the period (.) or the comma (,). They must be enclosed between single quotes. The default is the comma (**,**).

RECDEF vector example

```
/* Vectors RECDEF and RECFIELD Examples */

/* RECORD1 Definition */
        RECDEF   NAME=RECORD1,
                 DELIMIT='/',
                 DECSEP=','
        RECFIELD NAME=REC1FL01,
                 LENGTH=4,
                 FORMAT=C,
                 DESCRIPT='TRANSACTION'
        RECFIELD NAME=FILLER1,
                 LENGTH=2,
                 FORMAT=C,
                 DESCRIPT='2 BYTE FILLER'
        RECFIELD NAME=REC1FL02,
                 LENGTH=10,
                 FORMAT=N,
                 DECIMALS=2,
                 DESCRIPT='AMOUNT'
        RECFIELD NAME=REC1FL03,
                 LENGTH=8,
                 FORMAT=N,
                 DECIMALS=0,
                 DESCRIPT='CLIENT NUMBER'

/* RECORD2 Definition */
        RECDEF   NAME=RECORD2,
                 DELIMIT='/',
                 DECSEP=','
        RECFIELD NAME=REC2FL01,
                 LENGTH=10,
                 FORMAT=N,
                 DECIMALS=2,
                 DESCRIPT='BALANCE'
        RECFIELD NAME=REC2FL02,
                 LENGTH=4,
                 FORMAT=P,
                 DECIMALS=0,
                 DESCRIPT='CHECK TOTAL'
        RECFIELD NAME=REC2FL03,
                 LENGTH=4,
                 FORMAT=P,
                 DECIMALS=0,
                 DESCRIPT='CASH TOTAL'
```

RECFIELD vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- LANDP for AIX

Defines a field inside a record structure.

At least one RECFIELD vector must be specified following a RECDEF vector. Up to 92 RECFIELD vectors can follow a RECDEF vector.

A quick reference

Vector position	Follows RECDEF vector. This vector can exist up to 92 times after a RECDEF vector.
List of keywords	NAME, DECIMALS, DESCRIPT, FORMAT, LENGTH
Keywords relate to	**Keyword** **Relates to keywords** **NAME** KEYxx (EJOUPRF, SHFLDBD, and SFORWPRF vectors), KEYFIELD (SHFLPCB vector)
Vector format	RECFIELD NAME=*xxxxxxxx*, [DECIMALS=*x*,] [DESCRIPT=' ',] [FORMAT=*xx*,] LENGTH=*xxxx*

Keyword **Description**

NAME Is a required keyword to define the field name.

The parameter value is a string of up to eight alphanumeric characters plus the special characters $, %, #, and @. It must start with an alphanumeric character and must be unique within one RECDEF vector, but can be repeated in different RECDEF vectors.

DECIMALS

Is an optional keyword to define the number of decimal places in a field. This parameter only applies to the field format types N (ASCII numeric), SN (signed ASCII numeric), P (Packed), and SP (signed packed).

The parameter value can be a number from 0 to 9, and must not be greater than the field length. The default is **0**.

DESCRIPT

Is an optional keyword that contains the logical description of the field.

The parameter value is a string with up to 30 alphanumeric characters. As this field may contain blanks or special characters, it must be enclosed between quotes. If quotes are part of the description, they must be doubled.

FORMAT Is an optional keyword to define the field formats.

The parameter values are:

B	Inverted binary
C	Character
CD	DBCS characters (pure DBCS)
CM	SBCS and DBCS characters (mixed, SBCS and DBCS)
H	Hexadecimal
I	Integer
SI	Signed Integer
L	Long Integer
SL	Signed long integer
N	ASCII numeric
SN	Signed ASCII numeric
P	Packed
SP	Signed packed

Values CD and CM apply only to DBCS mode. The default is **C**.

LENGTH Is a required keyword to define the length of the field.

The parameter values are numeric characters from 0 to 4096. The allowed length depends on the field format:

B, I, and SI
: 2 (can be omitted)

L, SL
: 4 (can be omitted)

C, CD, CM, H, N, and SN
: A number from 0 to 4096 (cannot be omitted). If you specify 0, the field has variable length, and the delimiter is used to indicate the end of the field. Type CD requires an even number.

P and SP
: A number from 1 to 4096. The length of a packed field (P) is the display length excluding the decimal separator. If this packed field is signed (SP), you have to add one byte. The storage length of a packed field is the display length divided by two and rounded up.

Variable fields in a record will be considered length 1. At run time, the sum of all lengths in a record cannot exceed 4096 bytes.

RECFIELD vector example

```
/* Vectors RECDEF and RECFIELD Examples */

/* RECORD1 Definition */
         RECDEF   NAME=RECORD1,
                  DELIMIT='/',
                  DECSEP=','
         RECFIELD NAME=REC1FL01,
                  LENGTH=4,
                  FORMAT=C,
                  DESCRIPT='TRANSACTION'
         RECFIELD NAME=FILLER1,
                  LENGTH=2,
                  FORMAT=C,
                  DESCRIPT='2 BYTE FILLER'
         RECFIELD NAME=REC1FL02,
                  LENGTH=10,
                  FORMAT=N,
                  DECIMALS=2,
                  DESCRIPT='AMOUNT'
         RECFIELD NAME=REC1FL03,
                  LENGTH=8,
                  FORMAT=N,
                  DECIMALS=0,
                  DESCRIPT='CLIENT NUMBER'

/* RECORD2 Definition */
         RECDEF   NAME=RECORD2,
                  DELIMIT='/',
                  DECSEP=','
         RECFIELD NAME=REC2FL01,
                  LENGTH=10,
                  FORMAT=N,
                  DECIMALS=2,
                  DESCRIPT='BALANCE'
         RECFIELD NAME=REC2FL02,
                  LENGTH=4,
                  FORMAT=P,
                  DECIMALS=0,
                  DESCRIPT='CHECK TOTAL'
         RECFIELD NAME=REC2FL03,
                  LENGTH=4,
                  FORMAT=P,
                  DECIMALS=0,
                  DESCRIPT='CASH TOTAL'
```

SFORWPRF vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- LANDP for AIX

Defines a store-for-forwarding server profile. Define one SFORWPRF vector for each store-for-forwarding profile you want to specify.

When you define a DOS, OS/2 or Windows NT workstation, you must define as many records as you need in vectors RECDEF and RECFIELD.

When you define a RS/6000 workstation, the records are defined using a record definition utility program, shown in *LANDP Servers and System Management*.

A quick reference

Vector position	None. Can be followed by at least one SFORWREC vector.
List of keywords	NAME, DATASETS, SEPSESS, MAXACC, SHFLPRF, DBDPATH, SPLIT, KEY02 to KEY15
Keywords relate to	**Keyword** **Relates to keywords** **NAME** PAR&SFOR (parameter 1) **SHFLPRF** SHFLPRF (SHFLDBD vector) **KEYxx** NAME (RECFIELD vector)
Vector format	SFORWPRF NAME=*xxxxxxxx*, [DATASETS=*xx*,] [SEPSESS=*x*,] [MAXACC=*xxxxx*,] SHFLPRF=*xxxxxxxx*, DBDPATH=*path*, SPLIT=*length*, [KEY02=*xxxxxxxx*,] : [KEY15=*xxxxxxxx*]

Keyword **Description**

NAME Is a required keyword to define the name of the store-for-forwarding profile. The parameter value is a field with up to eight alphanumeric characters plus the special characters $, %, #, and @. It must be unique among all SFORWPRF vectors.

DATASETS
Is an optional keyword to define the number of different store-for-forwarding data sets to be used.

The parameter value is a field with up to two characters, ranging from 1 to 64. The default is **3**.

If you are using the forwarding server, this value must be greater than or equal to the number of FORWDS vectors defined for the forwarding profile that will be used together with this profile. The other data sets will not be forwarded and can be accessed by the applications.

SEPSESS LANDP for DOS, OS/2, and Windows NT: Is an optional keyword to define the method for providing integrity for data access to the store-for-forwarding server records.

Specify Y (Yes) if you want session integrity to be handled by the store-for-forwarding server, independently of your own shared-file server session. The default is **N** (No).

MAXACC LANDP for DOS, OS/2, and Windows NT: Is an optional keyword to define the maximum number of accesses in search operations. This is the maximum number of searches made before the search operation is ended and control is returned to the application.

The parameter value is a string of up to 5 numeric characters, which can range from 1 to 32767. The default is **200**.

SHFLPRF LANDP for DOS, OS/2, and Windows NT: Is a required keyword to define the name of the shared-file profile to be used. You can use the name of an already defined shared-file profile in the SHFLPRF keyword of the SHFLDBD vector, or a new name. If you provide a new profile name, the corresponding shared-file profile is created by the LANDP customization.

In any case, the customization adds to the shared-file profile the DBD and PCB definitions required for this store-for-forwarding profile.

The same shared-file profile can only be used by one store-for-forwarding profile. If you have the store-for-forwarding server and the electronic journal server in the same workstation, you must use the same shared-file profile for both of them.

The parameter value is a field with up to eight alphanumeric characters. It must be unique among the SFORWPRF vectors.

DBDPATH LANDP for DOS, OS/2, and Windows NT: Is a required keyword to define the path where the store-for-forwarding data and index files are located. This path must exist prior to running the LANDP family programs in a workstation with the store-for-forwarding server services.

The parameter value is the complete path (drive plus subdirectories). The number of subdirectories must not exceed four, and has to end with a back slash (\). It must at least contain three characters.

Note: LANDP for AIX store-for-forwarding server uses the path defined in Informix.

SPLIT LANDP for DOS, OS/2, and Windows NT: Is a required keyword to define the length of the user record that can be kept in one store-for-forwarding record.

Parameter values range from 62 to 4096 minus the size of the header that is appended. The header length is the length of all the defined keys (including the store-for-forwarding hidden key SFHIDKEY which is 8 bytes long), plus 36.

If the record size exceeds the number specified, it will be split into two or more segments.

KEY02 to KEY15
LANDP for DOS, OS/2, and Windows NT: Are optional parameters to define the names of the fields used as keys. These field names must be defined in at least one of the records specified in the SFORWREC vectors. If fields specified as keys are defined in more than one record, the characteristics of the field must match.

Fields in a record can be selected as store-for-forwarding keys only if they are in character, unsigned ASCII numeric, or hexadecimal, and their length is fixed and less than 50 bytes.

The keys must be specified sequentially; for example, KEY05 cannot be specified if KEY04 was omitted. The first key (KEY01) is not defined. It is always the store-for-forwarding hidden key SFHIDKEY.

SFORWPRF vector example

```
/* Vectors SFORWPRF and SFORWREC Examples */

         SFORWPRF NAME=SFFPRF01,

                  DATASETS=10,
                  SEPSESS=N,
                  MAXACC=200,
                  SHFLPRF=SFPROF01,
                  DBDPATH=C:\TELLER\SFORW\,
                  SPLIT=1024,
                  KEY02=REC1FL03,
                  KEY03=REC2FL02

        SFORWREC  RECNAME=RECORD1
        SFORWREC  RECNAME=RECORD2
```

SFORWREC vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- LANDP for AIX

Defines the names of the records defined with the RECDEF vector which will be used as store-for-forwarding records.

At least one SFORWREC vector must be specified following a SFORWPRF vector.

A quick reference

Vector position	Follows SFORWPRF vector
List of keywords	RECNAME
Keywords relate to	**Keyword** **Relates to keywords** **RECNAME** NAME (RECDEF vector)
Vector format	SFORWREC RECNAME=*xxxxxxxx*

Keyword **Description**

RECNAME

Is a required keyword to specify a name of a record which will be used as an store-for-forwarding record.

The parameter value is a field with up to eight alphanumeric characters, and must match the name given in keyword NAME of RECDEF vector.

If a record has a field with name AND or OR, the record cannot be selected as store-for-forwarding record.

SFORWREC vector example

```
/* Vectors SFORWPRF and SFORWREC Examples */

         SFORWPRF NAME=SFFPRF01,

                  DATASETS=10,
                  SEPSESS=N,
                  MAXACC=200,
                  SHFLPRF=SFPROF01,
                  DBDPATH=C:\TELLER\SFORW\,
                  SPLIT=1024,
                  KEY02=REC1FL03,
                  KEY03=REC2FL02

        SFORWREC  RECNAME=RECORD1
        SFORWREC  RECNAME=RECORD2
```

SHFLDBD vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- LANDP for AIX

Defines a shared-file database description (DBD) for the shared-file server. You can define up to 2048 DBDs.

For each shared-file server you create:

- The shared-file server structure, this means, as many database descriptions (DBDs) as you want to define using this vector.
- As many program control blocks (PCBs) as you want to define, using the SHFLPCB vector.

A shared-file profile is made up of all the SHFLDBD vectors which have the same identification in keyword SHFLPRF.

Define one SHFLDBD vector for each shared-file database that you want to use in a shared-file profile. A shared-file profile is defined or referenced in the SHFLPRF keyword of this vector.

A quick reference

Vector position	None. Can be followed by as many SHFLPCB vectors as needed.
List of keywords	SHFLPRF, DBDNAME, RECNAME, COLLKEYS, PHYSREC, RECLEN, DBDPATH, DBFLNAME, KEY01 to KEY15
Keywords relate to	**Keyword** **Relates to keywords** **SHFLPRF** SHFLPRF (EJOUPRF and SFORWPRF vectors), NAME (COLSQTBL vector), PAR&SHFL (parameter 1) **RECNAME** NAME (RECDEF vector)
Vector format	SHFLDBD SHFLPRF=*xxxxxxxx*, DBDNAME=*xxxxxxxx*, [RECNAME=*xxxxxxxx*,] [COLLKEYS=*x*,] [PHYSREC=*xxxxx*,] [RECLEN=*xxxxx*,] DBDPATH=*path*, DBDPATH6=*path*, DBFLNAME=*xxxxxxxx*, [KEY01=( )], . . [KEY15=( )], [MINLOCKS=*xxxxx*,], [TLOCKS=*xxxxx*,]

Keyword **Description**

SHFLPRF Is a required keyword to define the name of the shared-file profile for this DBD. The parameter value is a string with up to eight alphanumeric characters.

If you specify to use an alternate collating sequence table, you can define the table and reference the SHFLPRF name in the keyword NAME of the COLSQTBL vector.

DBDNAME

Is a required keyword to define the name of one DBD in a shared-file profile. The parameter value is a string with up to eight alphanumeric characters. It must be unique in the profile.

Do not use names starting with FBEJ, since they are reserved for electronic journal DBDs. Those starting with FBSF are reserved for store-for-forwarding DBDs.

Applications refer to a DBD by using one of the related program control blocks (SHFLPCB vectors).

RECNAME

Is an optional keyword with the name of a record structure (defined in the RECDEF vector) to be used as a pattern for name, length, and offset of the key fields.

If specified, it must match the name given in a RECDEF vector, keyword NAME.

If no record structure is referenced, the characteristics of each key field must be defined (KEY01 to KEY15). Records with variable length fields cannot be used as a pattern.

COLLKEYS

Is an optional keyword to define whether an alternate collating sequence table will be used or not.

The parameter values are:

Y (Yes) - the table will be used
N (No) - the table will not be used

The default is **N**.

This definition is valid for the third parameter of the key definitions (keywords KEYxx). If you select Y, you can define for each key whether the alternate collating sequence table will be used or not. If you use the default, the keys must also be defaulted to N.

PHYSREC Is an optional keyword to define the record split size. This enables to use variable length records inside the shared file server databases. No record can be split in more than 200 physical records.

The parameter value can be between 44 (minimum length) and the data record length specified in the RECLEN keyword.

Each physical record contains 5 control bytes. Thus, the number of user data bytes in a physical record will be the parameter value you specify in the PHYSREC keyword minus 5.

RECLEN Is a keyword to define the record length, that is, the number of bytes in each data record.

If you have specified to use a previously defined record structure (keyword RECNAME), the keyword does not apply.

If you have not specified it, the keyword is required. The parameter value can be between 44 (minimum length) and 26624 (maximum length).

DBDPATH Is a required keyword to define the path where the data and index files will be located.

The parameter value is the complete path (drive plus subdirectories). It must not exceed 56 characters, and contain at least 3 characters. Always include the final \ (backslash) after the last directory.

The path must exist prior to running the LANDP family programs in a workstation with the shared-file server.

DBDPATH6

LANDP for AIX: Is a required keyword to define the path where the data and index files will be located in an AIX system.

The parameter value is the complete path including the subdirectories. It must not exceed 56 characters, and must be enclosed within single quotes ('). Each subdirectory name included in the path must have one character at least, and 30 characters at most.

As with the DBDPATH keyword, the path must exist prior to running the LANDP family programs in a workstation with the shared file server.

If you have speceified a DBDPATH keyword, you cannot also specify a DBDPATH6 keyword. Similarly, if you have specified DBDPATH6, then you cannot specify DBDPATH.

DBFLNAME

Is a required keyword to define the name of the data and index files.

The parameter value is a field with up to eight characters for the file name. The file extension must be omitted. They will be DAT for data, or IX*n* (or I*nn* for those *nn* greater than or equal to 10) for key index files.

KEY01 to KEY15

Are optional keywords to define the key fields.

- The keys must be specified sequentially; for example, KEY05 cannot be specified if KEY04 was omitted.
- If keyword KEY01 is omitted, only direct or sequential access will be allowed for this DBD.
- If you have not defined a record name in keyword RECNAME, all key parameters have to be specified.

  If RECNAME has been defined, the name, key class, and the use of an alternate collating sequence table parameters must be specified. Key length and key offset, however, must not be specified.
- If you have selected the default (N) for an alternate collating sequence table (keyword COLLKEYS), do not specify the third parameter of these keywords.

Each keyword has five parameters:

1. The first parameter specifies the name of the key field. If RECNAME was specified, this name must reference a field in the record structure definition. The name will be used to refer to this key in the SHFLPCB vector.

   The parameter value is an alphanumeric field with up to eight characters.
2. The second parameter specifies the key class. The parameter value can be:

   P Cannot be changed, unique

S Can be changed, not unique
M Cannot be changed, not unique
D Can be changed, unique

The default is **S**.

3. The third parameter specifies information related to alternate collating sequence. The parameter value can be:

   N The key does not require alternate collating sequence.

   Y The key requires alternate collating sequence, using a table with extension SEQ.

   X The key requires alternate collating sequence, using a table with extension SE*X*, where *X* can be any hexadecimal character, except for 0.

   D The key requires DBCS-style alternate collating sequence.

   The parameter does not apply to segmented keys. The parameter value S applies only to DBCS mode. The default is **N**.

4. The fourth parameter specifies either segmented key or the key field length, that is, the number of bytes the field contains.

   If you plan to define a segmented key, specify value 0. To specify a non-segmented key field length, the parameter value can be between 1 and 255.

5. The fifth parameter specifies the key field offset, that is, the byte in the data record that denotes the beginning of a key. Byte numbering in the data record begins with 0.

   The parameter does not apply to segmented keys. The parameter value can be between 0 and 26623. The offset specified for a key plus the key length must not exceed the data record length.

MINLOCKS
Is an optional keyword that has no effect. It is provided as a migration aid from LANDP Version 1.0. If specified, it must be in the range 1 through 10922.

TLOCKS Is an optional keyword that has no effect. It is provided as a migration aid from LANDP Version 1.0. If specified, it must be in the range 1 through 10922.

SHFLDBD vector example

```
/* Vectors SHFLDBD, SHFLPCB, and SHFLSGM Examples */

         SHFLDBD  SHFLPRF=SAMPLE1,
                  DBDNAME=SAMP1DBD,
                  COLLKEYS=N,
                  RECLEN=100,
                  DBDPATH=D:\,
                  DBFLNAME=SAMP1FIL,
                 KEY01=(F1P,P,,0),
                 KEY02=(F1S,S,N,15,5)

        SHFLPCB  PCBNAME=PCBS1
        SHFLPCB  PCBNAME=PCBSF1P,
                 KEYFIELD=F1P
        SHFLPCB  PCBNAME=PCBSF1S,
                 KEYFIELD=F1S
        SHFLSGM  SGMKEY=1,
                 SGM01=(YEAR,D,2,34),
                 SGM02=(MONTH,D,2,32),
                 SGM03=(DAY,D,2,30)
```

SHFLPCB vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- LANDP for AIX

Defines a program control block (PCB) to access a previously defined DBD. You can define up to 2048 PCBs.

At least one SHFLPCB vector has to be specified for each access to a key field you are going to define.

A quick reference

Vector position	Follows SHFLDBD vector
List of keywords	PCBNAME, KEYFIELD
Keywords relate to	**Keyword** **Relates to keywords** **KEYFIELD** NAME (RECFIELD vector)
Vector format	SHFLPCB PCBNAME=*xxxxxxxx*, [KEYFIELD=*xxxxxxxx*]

Keyword **Description**

PCBNAME

Is a required keyword to define a name of a program control block (PCB) to be used by the application to refer to a particular DBD.

The parameter value is an alphanumeric field with up to eight characters. It must be unique among the SHFLPCB vectors for this profile.

KEYFIELD Is an optional keyword to define the name of the key field used to access the DBD for this PCB. It is used for indexed sequential access and indexed direct access.

The parameter value is the name given in the first parameter of the KEYxx keyword in the SHFLDBD vector.

All key fields defined in the DBD must have a corresponding PCB, if you want the record to be accessed through this key.

If the key name is omitted, the PCB is used for sequential or direct access (not indexed).

SHFLPCB vector example

```
/* Vectors SHFLDBD, SHFLPCB, and SHFLSGM Examples */

         SHFLDBD  SHFLPRF=SAMPLE1,
                  DBDNAME=SAMP1DBD,
                  COLLKEYS=N,
                  RECLEN=100,
                  DBDPATH=D:\,
                  DBFLNAME=SAMP1FIL,
                  KEY01=(F1P,P,,0),
                  KEY02=(F1S,S,N,15,5)

         SHFLPCB  PCBNAME=PCBS1
         SHFLPCB  PCBNAME=PCBSF1P,
                  KEYFIELD=F1P
         SHFLPCB  PCBNAME=PCBSF1S,
                  KEYFIELD=F1S
         SHFLSGM  SGMKEY=1,
                  SGM01=(YEAR,D,2,34),
                  SGM02=(MONTH,D,2,32),
                  SGM03=(DAY,D,2,30)
```

SHFLSGM vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- LANDP for AIX

Defines a segmented key. Specify one SHFLSGM vector for each segmented key specified through the KEYxx keywords in the SHFLDBD vector.

A quick reference

Vector position	Follows SHFLPCB vectors
List of keywords	SGMKEY, SGM01 to SGM32
Keywords relate to	**Keyword** **Relates to keywords** **SGMKEY** KEYxx (SHFLDBD vector) **SGMxx** RECNAME (SHFLDBD vector)
Vector format	SHFLSGM SGMKEY=*xx*, SGM01=( ), [SGM02=( ),] . . [SGM32=( )]

Keyword **Description**

SGMKEY Is a required keyword to specify the segmented key.

The parameter value is a two-digits string that ranges from 1 to 15, and must match the two last digits of the KEYxx keyword of the SHFLDBD vector

The total length of all the segments of a segmented key cannot exceed 255.

SGM01 to SGM32

Are keywords to define the segments of the segmented key. SGM01 is required. Keywords from SGM02 to SGM32 are optional, and can not be used if the previous one in the sequence has not been specified.

Each keyword has six parameters:

1. The first parameter specifies the name of the segment field. If RECNAME was specified in the SHFLDBD vector, this name must reference a field in the record structure definition.

   The parameter value is an alphanumeric field with up to eight characters.

2. The second parameter specifies the segment type. The parameter value can be:

   D Data
   I Indicator
   L Locally nullable
   G Globally nullable

   The default is **D**.

3. The third parameter specifies the segment field length, that is, the number of bytes the field contains.

   If RECNAME was specified in the SHFLDBD vector, the parameter does not apply. The parameter value can be between 1 and 255.

4. The fourth parameter specifies the segment field offset, that is, the byte in the data record that denotes the beginning of the segment. Byte numbering in the data record begins with 0.

   If RECNAME was specified in the SHFLDBD vector, the parameter does not apply. The parameter value can be between 0 and 26623.

   The offset specified for a segment plus the segment length must not exceed the data record length (RECLEN keyword in SHFLDBD vector).

5. The fifth parameter specifies information related to alternate collating sequence. The parameter value can be:

   N The segment does not require alternate collating sequence.

   Y The segment requires alternate collating sequence, using a table with extension SEQ.

   X The segment requires alternate collating sequence, using a table with extension SE*X*, where *X* can be any hexadecimal character, except for 0.

   S The segment requires DBCS-style alternate collating sequence.

   The parameter value S applies only to DBCS mode. The default is **N**.

6. The sixth parameter specifies the segment nullable value, that is, the character for comparison to make null a segment.

   The parameter does not apply to data or indicator segments, and is required for locally or globally nullable segments. The parameter value is a string of two hexadecimal digits.

SHFLSGM vector example

```
/* Vectors SHFLDBD, SHFLPCB, and SHFLSGM Examples */

        SHFLDBD  SHFLPRF=SAMPLE1,
                 DBDNAME=SAMP1DBD,
                 COLLKEYS=N,
                 RECLEN=100,
                 DBDPATH=D:\,
                 DBFLNAME=SAMP1FIL,
                 KEY01=(F1P,P,,0),
                 KEY02=(F1S,S,N,15,5)

        SHFLPCB  PCBNAME=PCBS1
        SHFLPCB  PCBNAME=PCBSF1P,
                 KEYFIELD=F1P
        SHFLPCB  PCBNAME=PCBSF1S,
                 KEYFIELD=F1S
        SHFLSGM  SGMKEY=1,
                 SGM01=(YEAR,D,2,34),
                 SGM02=(MONTH,D,2,32),
SGM01=(DAY,D,2,30)
```

SMGRPRF vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT

Defines a system manager server profile. One vector has to be defined for each system manager server profile you want to use.

There are two conditions that determine if record structures (defined in vector RECDEF) are used by the system manager server:

1. LAN common data validation

   If the system manager server is to validate the LAN common data, the customization creates the LAN common data, based on the record structure of record FBSSGLUS. This record must be created before defining a system manager server profile.

2. Application data validation

   If the system manager server is to validate the application data, a record structure named FBSSUSPR must exist and is used to build the application data. You must create this record structure before you define a system manager server profile.

The records FBSSGLUS and FBSSUSPR can have up to 1024 bytes and cannot contain variable length fields.

A quick reference

Vector position	None. If you define that system security will be used (SECLVL=Y), this vector must be followed by SMGRUSER vectors that define the authorization level of each user.
List of keywords	NAME, LANID, LOGSUPP, SECLVL, AUTSYNC, COMDTVAL, APPDTVAL, LOGFSIZE, LOGRECL, LEVELS, LEVELF, LEVELR, LEVELOA, LEVELM, MSGOPRCV, ALRHNDLR
Keywords relate to	**Keyword** **Relates to keywords** **NAME** PAR&SMGR (parameter 4) **MSGOPRCV** USERID (SMGRUSER vector)

Vector format	SMGRPRF NAME=*xxxxxxxx*, [LANID=*xxxxxxxx*,] [LOGSUPP=*x*,] [SECLVL=*x*,] [AUTSYNC=*x*,] [COMDTVAL=*x*,] [APPDTVAL=*x*,] [LOGFSIZE=*xxx*,] [LOGRECL=*xxxx*,] [LEVELS=*x*,] [LEVELF=*x*,] [LEVELR=*x*,] [LEVELOA=*x*,] [LEVELM=*x*,] [MSGOPRCV=*x*,] [ALRHNDLR=*x*]

Keyword **Description**

NAME Is a required keyword to define the system manager profile name.

The parameter value is a string with up to eight alphanumeric characters plus the special characters $, %, #, and @. It must start with an alphabetical character, and must be unique among all SMGRPRF vectors.

LANID Is an optional keyword to identify the workgroup. It is used by the system manager server to complete the hierarchy subvector of the alert as a service point (SP).

The parameter value is a string with up to eight alphanumeric characters plus the special characters $, %, #, and @. The default is **LANID**.

It is strongly recommended that the value matches the PU name defined in VTAM for the workstation providing SNA services for alerts. For OS/2 and Windows NT workstations, it is also recommended that the value matches SNA local node name (control point alias) defined in the communications provider configuration file, so that the RUNCMD files sent from the host computer can reach the LANDP workgroup.

LOGSUPP Is an optional keyword to define whether you want to have system manager support for the system and user LOG (Y). The default is no (**N**).

The LOG file is created and maintained by the LANDP System Manager server. It can only be deleted when the LANDP program is not running.

If you select Y for this keyword, LOGFSIZE and LOGRECL keywords must be specified, or the default values will be taken. If you select N, these keywords do not apply.

SECLVL Is an optional keyword to define if you want the system manager to control access to the LANDP family resources.

The parameter value can be either Y (Yes) or N (No). The default is **Y**.

If you select Y for this keyword, you must choose which authorization levels you want the system manager to control. You do this by using the LEVELx keywords of the same vector. You must also have an user assigned as system manager administrator, so the keyword LEVELOA must be Y.

If you select N, no LEVELx=Y can be specified.

AUTSYNC Is an optional keyword to define if the system manager will provide automatic synchronization of date and time in all personal computer and PS/2 workstations in the workgroup.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

COMDTVAL

Is an optional keyword to define if the system manager performs validation of the LAN common data. This validation requires the FBSSGLUS record, defined in the RECDEF vector.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

APPDTVAL

Is an optional keyword to define if the system manager performs validation of the application data. This is data about the user, not about user transactions. This validation requires the FBSSUSPR record, defined in the RECDEF vector.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

LOGFSIZE

Is an optional keyword to define the log file size. When the log file is full, it wraps and begins overlaying records at the beginning of the file. It is only applicable, if LOGSUPP=Y.

The parameter values range from 10 to 500KB. The default is **100**.

LOGRECL Is an optional keyword to define the log record size. It is only applicable, if LOGSUPP=Y.

The parameter values range from 128 to 1024 bytes of memory (that is 960 bytes of data). The default is **128**.

LEVELS Is an optional keyword to define whether the system manager server controls access to the debug and trace tools (DDT) and log files (LOG). This is security level **S**.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

LEVELF Is an optional keyword to define whether the system manager server controls access to the LANDP family resources. This is security level **F**.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

If you have defined your workgroup to use LEVELF security, a user must be signed on with LEVELF authority to use any applications from a LANDP workstation.

LEVELR Is an optional keyword to define whether the system manager server controls access to the shared DOS directory. This is security level **R**.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

LEVELOA Is an optional keyword to define whether the system manager server controls access to administrator and operator functions. Authorization for access for system manager operators is security level **O**, for system manager administrators it is **A**.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

If you select Y for this keyword, at least one LEVELA=Y user must be specified in the SMGRUSER vector.

LEVELM Is an optional keyword to define whether the system manager server controls access for administrator and operator functions from an application instead of from system manager operator. This is security level **M**.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

Select Y if you want to designate one or more users of the applications to have access to the administrator and operator functions.

MSGOPRCV

Is an optional keyword to define the user identification of the operator messages receiver. This user must have authorization level **O**. The name must be referenced in keyword USERID in one of the SMGRUSER vectors.

The parameter value is a string of up to eight alphanumeric characters.

If only one LEVELO=Y has been specified in the SMGRUSER vector, or if LOGSUPP=N has been defined, this keyword can be omitted. Otherwise, it must be specified.

ALRHNDLR

Is an optional keyword to define the Alert Handler User Exit.

The parameter value is a string of up to eight alphanumeric characters. The default is **EHCALRH**.

SMGRPRF vector example

```
/* Vectors SMGRPRF and SMGRUSER Examples */

       SMGRPRF  NAME=SMGRPRF1,
                LANID=MYLAN,
                LOGSUPP=Y,
                SECLVL=Y,
                AUTSYNC=Y,
                COMDTVAL=Y,
                APPDTVAL=Y,
                LOGFSIZE=200,
                LOGRECL=512,
                LEVELS=Y,
                LEVELF=Y,
                  LEVELR=N,
                  LEVELOA=Y,
                  LEVELM=Y,
                  MSGOPRCV=BOBSM


         SMGRUSER USERID=BOBSM,
                  PASSWORD=BOBSMITH,
                  NAME='Bob Smith',
                  LANGUAGE=33,
                  LEVELS=Y,
                  LEVELR=N,
                  LEVELF=Y,
                  LEVELO=Y,
                  LEVELA=Y,
                  LEVELM=N,
                  APPLVL='12345ABCDE'
```

SMGRUSER vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT

Defines a system manager user profile inside a system manager server profile. One SMGRUSER vector has to be defined for each system manager user profile you want to specify.

If you have defined your workgroup to use LEVELF security, a user must be signed on with LEVELF authority to use any applications from a LANDP workstation.

If you plan to use LEVELF security in a workgroup that includes an RS/6000 workstation, you need to add SMGRUSER entries for each user ID that you plan to use on that workstation. See “Security in a mixed-system workgroup” on page 208 for further details.

If no SMGRUSER vector is specified when keyword SECLVL=Y in SMGRPRF vector, the defaults will be USERID=**SYSTEM** and PASSWORD=**MANAGER**.

A quick reference

Vector position	Follows SMGRPRF vector
List of keywords	USERID, PASSWORD, PSWCRYPT, NAME, LANGUAGE, LEVELS, LEVELR, LEVELF, LEVELO, LEVELA, LEVELM, APPLVL
Keywords relate to	**Keyword** **Relates to keywords** **USERID** MSGOPRCV (SMGRPRF vector)

Vector format	SMGRUSER USERID=*xxxxxxxx*, PASSWORD=*xxxxxxxx*, [PSWCRYPT=password encrypted,] [NAME=' ',] [LANGUAGE=*xx*,] [LEVELS=*x*,] [LEVELR=*x*,] [LEVELF=*x*,] [LEVELO=*x*,] [LEVELA=*x*,] [LEVELM=*x*,] [APPLVL=' ']

Keyword **Description**

USERID Is a required keyword to identify the system manager user.

The parameter value is a string with up to eight alphanumeric characters plus the special characters $, %, #, and @. It must be unique within a system manager profile.

PASSWORD
Is a required keyword to define the user password to control user logon.

The parameter value is a string with four to eight alphanumeric characters plus the special characters $, %, #, and @.

PSWCRYPT
This is not an input keyword. The PASSWORD specification will be converted by the GENSPEC program into its encrypted form, and will from then on appear under the PSWCRYPT keyword, which substitutes the PASSWORD keyword.

This is a security provision. No changes can be made to the encrypted password. In order to change an user password in an already generated profile, you have to substitute the PSWCRYPT keyword by a PASSWORD keyword with the new information.

NAME Is an optional keyword to define the user name.

The parameter value is an alphanumeric field with a maximum of 30 characters. Since blanks are allowed in this field, it must be enclosed within quotes. If quotes are part of the name, they must be doubled.

LANGUAGE
Is an optional keyword to define a language code which will be defined in and used by your own applications.

The parameter value is a numeric field with two characters, or blanks. For example, you can define language code 01 for Italian and assign this code to various users.

LEVELS Is an optional keyword to define whether the user has access to the debug and trace tools (DDT) and log files (LOG).

The parameter value can be either Y (Yes) or N (No). The default is **N**.

LEVELR Is an optional keyword to define whether the user has access to the shared DOS directory.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

LEVELF Is an optional keyword to define whether the user has access to the LANDP family resources.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

LEVELO Is an optional keyword to define whether the user has access to system manager operator services.

The parameter value can be either Y (Yes) or N (No). The default is **N**.

If you want an user to receive the operator messages, you must define at least one user with authorization level **O**.

LEVELA Is an optional keyword to define whether the user has access to system manager operator services for administrator functions.

The parameter value can be either Y (Yes) or N (No). The default is **Y**.

LEVELM Is an optional keyword to define whether the user has authorization to access operator and administrator functions from an application (instead of from the system manager operator).

The parameter value can be either Y (Yes) or N (No). The default is **N**.

APPLVL Is an optional keyword to define application authorization levels. Ten different authorization levels are allowed. The user application can use this information to control access to its user application functions.

The parameter value is a field of up to 10 characters.

SMGRUSER vector example

```
/* Vectors SMGRPRF and SMGRUSER Examples */

       SMGRPRF  NAME=SMGRPRF1,
                LANID=MYLAN,
                LOGSUPP=Y,
                SECLVL=Y,
                AUTSYNC=Y,
                COMDTVAL=Y,
                APPDTVAL=Y,
                LOGFSIZE=200,
                LOGRECL=512,
                LEVELS=Y,
                LEVELF=Y,
                LEVELR=N,
                LEVELOA=Y,
                 LEVELM=Y,
                 MSGOPRCV=BOBSM

        SMGRUSER USERID=BOBSM,
                 PASSWORD=BOBSMITH,
                 NAME='Bob Smith',
                 LANGUAGE=33,
                 LEVELS=Y,
                 LEVELR=N,
                 LEVELF=Y,
                 LEVELO=Y,
                 LEVELA=Y,
                 LEVELM=N,
                 APPLVL='12345ABCDE'
```

SOFTPACK vector

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- LANDP for AIX
- PC/Integrator
- PC Integrator/2

Defines a list of non-LANDP files that are to be distributed with the LANDP runtime files. These files can be application programs, additional device drivers, bitmaps, or any other files required in the LANDP workstation.

This vector can define two types of pack:

- Pack of files to be distributed. Such a pack can contain any number of files. A reference to the pack (in the SOFTPACK keyword of the LWSCONF or WSMODEL vector) causes the distribution of all the files in the pack.
- Pack of packs. Such a pack is a list of predefined packs of the first type. Such a pack can contain any number of predefined packs. A reference to a pack of packs (in the SOFTPACK keyword of the LWSCONF or WSMODEL vector) causes the distribution of all the files in each pack it lists.

A quick reference

Vector position	None
List of keywords	PACKNAME, FILENAME, FILEPATH
Keywords relate to	
Vector format	SOFTPACK PACKNAME=*xxxxxxxx*, FILENAME=*xxxxxxxx[.xxx]*, [FILEPATH=*path*]

Keyword **Description**

PACKNAME
Is a required keyword to define the softpack name. This name is used to group all components of the same pack and to provide the name that is used when the pack is referenced by an LWSCONF or WSMODEL vector.

FILENAME
Is a required keyword, to specify either:

- The filename and file extension of a file that is to be distributed, or
- The name of a predefined pack of files

FILEPATH Is a optional keyword that definesthe full path of the files location in the customization workstation. The name includes the drive, and all directory names.

If FILENAME specifies a predefined pack of files, this keyword must be omitted.

SOFTPACK vector example

```
/* Sample of packs of files */

  SOFTPACK PACKNAME=PACKO1,
           FILENAME=FILE101.EXE,
           FILEPATH=C:\SUBDIR1\SUBDIR2\

  SOFTPACK PACKNAME=PACKO1,
           FILENAME=FILE102.EXE,
           FILEPATH=C:\SUBDIR1\SUBDIR2\

  SOFTPACK PACKNAME=PACKO2,
           FILENAME=FILE201.EXE,
           FILEPATH=C:\SUBDIR1\SUBDIR2\

  SOFTPACK PACKNAME=PACKO2,
           FILENAME=FILE202.EXE,
           FILEPATH=C:\SUBDIR1\SUBDIR2\

/* Sample of pack of packs */

  SOFTPACK PACKNAME=GPACKO1,
           FILENAME=PACK01

  SOFTPACK PACKNAME=GPACKO1,
           FILENAME=PACK02
```

XLATETBL vector

- LANDP for DOS
- LANDP for OS/2

- LANDP for Windows NT
- LANDP for AIX
- PC/Integrator
- PC Integrator/2

Defines an ASCII to EBCDIC or an EBCDIC to ASCII translation table. One separate vector has to be specified for each table you are going to define.

For each type of table, a default table will be created by CREATE. The vectors for these default tables can be generated using GENSPEC.

A quick reference

Vector position	None
List of keywords	TYPE, EXTEN, DATA0X to DATAFX
Keywords relate to	**Keyword** **Relates to keywords** **EXTEN** SES&3270 (parameters 6 and 7), PAR&3287 (parameter 2), SES&BPP (parameter 5), PAR&BIWP (parameters 14 and 15), PAR&RCMS (parameters 8 and 9), PAR&FORW (parameter 2), EXTEN (KBDBIWP, KBD3270, and KBD3270X vectors), EAEXTEN (KBDBIWP vector), CHSTRID (BPPPARM vector)

Vector format	XLATETBL TYPE=*xxxx*,
	EXTEN=*xxx*,
	DATA0X=( ),
	:
	DATAFX=( )

Keyword **Description**

TYPE Is a required keyword to define the table type. It does not apply to DBCS mode.

The parameter values can be one of the following:

AE3270 ASCII to EBCDIC translation for 3270 Emulation
EA3270 EBCDIC to ASCII translation for 3270 Emulation
AEBIWP ASCII to EBCDIC translation for BIWP Emulation
EABIWP EBCDIC to ASCII translation for BIWP Emulation
AERCMS ASCII to EBCDIC translation for RCMS
EARCMS EBCDIC to ASCII translation for RCMS
AEFORW ASCII to EBCDIC translation for Forwarding
EA47X2 EBCDIC to ASCII translation for BPP Emulation
EA3287 EBCDIC to ASCII translation for 3287 Emulation

EXTEN Is a required keyword to define the translation table identification.

The parameter value is a 3-character alphanumeric string, and may contain the special characters #, $, %, and @. It must be unique among all translation tables of the same type.

DATA0X to DATAFX
Are required keywords to define the table values. The parameter value is a string of 16 parameters, each one with a 1-byte digit from '00' to 'FF'.

DATA0X must contain the translation information for digits '00' to '0F', DATA1X for '10' to '1F', and so on.

For translation tables with the type EA3270, EA47X2, EA3287, and EABIWP, the keywords DATA0X, DATA1X, DATA2X, and DATA3X must not be specified.

XLATETBL vector example

```
/* Vector XLATETBL (TYPE=AE3270) Default Example */

   XLATETBL TYPE=AE3270,
            EXTEN=KBD,
            DATA0X=(00,01,02,03,04,05,06,07,08,09,0A,0B,0C,0D,0E,0F),
            DATA1X=(10,11,12,13,14,15,16,17,18,19,1A,1B,1C,1D,1E,1F),
            DATA2X=(40,4F,7F,7B,5B,6C,50,7D,4D,5D,5C,4E,6B,60,4B,61),
            DATA3X=(F0,F1,F2,F3,F4,F5,F6,F7,F8,F9,7A,5E,4C,7E,6E,6F),
            DATA4X=(7C,C1,C2,C3,C4,C5,C6,C7,C8,C9,D1,D2,D3,D4,D5,D6),
            DATA5X=(D7,D8,D9,E2,E3,E4,E5,E6,E7,E8,E9,4A,E0,5A,5F,6D),
            DATA6X=(79,81,82,83,84,85,86,87,88,89,91,92,93,94,95,96),
            DATA7X=(97,98,99,A2,A3,A4,A5,A6,A7,A8,A9,C0,BB,D0,A1,41),
            DATA8X=(68,DC,51,42,43,44,47,48,52,53,54,57,56,58,63,67),
            DATA9X=(71,9C,9E,CB,CC,CD,DB,DD,DF,EC,FC,70,B1,80,41,B4),
            DATAAX=(45,55,CE,DE,49,69,9A,9B,AB,AF,BA,B8,B7,AA,8A,8B),
            DATABX=(41,41,41,41,41,65,62,64,41,41,41,41,41,B0,B2,41),
            DATACX=(41,41,41,41,41,41,46,66,41,41,41,41,41,41,41,9F),
            DATADX=(8C,AC,72,73,74,9F,75,76,77,41,41,41,41,6A,78,41),
            DATAEX=(EE,59,EB,ED,CF,EF,A0,AE,8E,FE,FB,FD,8D,AD,BC,BE),
            DATAFX=(CA,8F,BF,89,B6,B5,41,9D,90,BD,B3,41,FA,EA,41,41)
```

XLAT2TBL vector

- PC/Integrator

Defines an ASCII to EBCDIC and an EBCDIC to ASCII translation table for the XLATE function of the PC/Integrator.

One vector has to be specified for each table you want to use in your workgroups. All the parameters for this vector have to be specified, there is no default for them.

A quick reference

Vector position	None

List of keywords	EXTEN, EADATA0X to EADATAFX, AEDATA0X to AEDATAFX
Keywords relate to	

Vector format	XLAT2TBL EXTEN=*xxx*,
	EADATA0X=( ),
	:
	EADATAFX=( ),
	AEDATA0X=( ),
	:
	AEDATAFX=( )

Keyword **Description**

EXTEN Is a required keyword to define a translation table.

The parameter value is a 3-character alphanumeric string, and may contain the special characters #, $, %, and @. It must be unique among all translation tables.

EADATA0X to EADATAFX
Are required keywords to define the EBCDIC to ASCII table values. The parameter value is a string of 16 parameters, each one with a 1-byte digit from '00' to 'FF'. These digits specify the ASCII value to be translated.

EADATA0X must contain the translation information for digit '00' to '0F', EADATA1X for '10' to '1F', and so on.

AEDATA0X to AEDATAFX
Are required keywords to define the ASCII to EBCDIC table values. The parameter value is a string of 16 parameters, each one with a 1-byte digit from '00' to 'FF'. These digits specify the EBCDIC value to be translated.

AEDATA0X must contain the translation information for digit '00' to '0F', AEDATA1X for '10' to '1F', and so on.

XLAT2TBL vector example

```
/* Vector XLAT2TBL Example */

 XLAT2TBL EXTEN=TB1,
          EADATA0X=(00,01,02,03,37,2D,2E,2F,16,05,25,0B,0C,0D,0E,0F),
          EADATA1X=(10,11,12,13,3C,3D,32,26,18,19,3F,27,1C,1D,1E,1F),
          EADATA2X=(40,4F,7F,7B,5B,6C,50,7D,4D,5D,5C,4E,6B,60,4B,61),
          EADATA3X=(F0,F1,F2,F3,F4,F5,F6,F7,F8,F9,7A,5E,4C,7E,6E,6F),
          EADATA4X=(7C,C1,C2,C3,C4,C5,C6,C7,C8,C9,D1,D2,D3,D4,D5,D6),
          EADATA5X=(D7,D8,D9,E2,E3,E4,E5,E6,E7,E8,E9,4A,E0,5A,5F,6D),
          EADATA6X=(79,81,82,83,84,85,86,87,88,89,91,92,93,94,95,96),
          EADATA7X=(97,98,99,A2,A3,A4,A5,A6,A7,A8,A9,C0,6A,D0,A1,07),
          EADATA8X=(20,21,22,23,24,15,06,17,28,29,2A,2B,2C,09,0A,1B),
          EADATA9X=(30,31,1A,33,34,35,36,08,38,39,3A,3B,04,14,3E,E1),
          EADATAAX=(41,42,43,44,45,46,47,48,49,51,52,53,54,55,56,57),
          EADATABX=(58,59,62,63,64,65,66,67,68,69,70,71,72,73,74,75),
          EADATACX=(76,77,78,80,8A,8B,8C,8D,8E,8F,90,9A,9B,9C,9D,9E),
EADATADX=(9F,A0,AA,AB,AC,AD,AE,AF,B0,B1,B2,B3,B4,B5,B6,B7),
EADATAEX=(B8,B9,BA,BB,BC,BD,BE,BF,CA,CB,CC,CD,CE,CF,DA,DB),
EADATAFX=(DC,DD,DE,DF,EA,EB,EC,ED,EE,EF,FA,FB,FC,FD,FE,FF),
AEDATA0X=(00,01,02,03,9C,09,86,7F,97,8D,8E,0B,0C,0D,0E,0F),
AEDATA1X=(10,11,12,13,9D,85,08,87,18,19,92,8F,1C,1D,1E,1F),
AEDATA2X=(80,81,82,83,84,0A,17,1B,88,89,8A,8B,8C,05,06,07),
AEDATA3X=(90,91,16,93,94,95,96,04,98,99,9A,9B,14,15,9E,1A),
AEDATA4X=(20,A0,A1,A2,A3,A4,A5,A6,A7,A8,5B,2E,3C,28,2B,21),
AEDATA5X=(26,A9,AA,AB,AC,AD,AE,AF,B0,B1,5D,24,2A,29,3B,5E),
AEDATA6X=(2D,2F,B2,B3,B4,B5,B6,B7,B8,B9,7C,2C,25,5F,3E,3F),
AEDATA7X=(BA,BB,BC,BD,BE,BF,C0,C1,C2,60,3A,23,40,27,3D,22),
AEDATA8X=(C3,61,62,63,64,65,66,67,68,69,C4,C5,C6,C7,C8,C9),
AEDATA9X=(CA,6A,6B,6C,6D,6E,6F,70,71,72,CB,CC,CD,CE,CF,D0),
AEDATAAX=(D1,7E,73,74,75,76,77,78,79,7A,D2,D3,D4,D5,D6,D7),
AEDATABX=(D8,D9,DA,DB,DC,DD,DE,DF,E0,E1,E2,E3,E4,E5,E6,E7),
AEDATACX=(7B,41,42,43,44,45,46,47,48,49,E8,E9,EA,EB,EC,ED),
AEDATADX=(7D,4A,4B,4C,4D,4E,4F,50,51,52,EE,EF,F0,F1,F2,F3),
AEDATAEX=(5C,9F,53,54,55,56,57,58,59,5A,F4,F5,F6,F7,F8,F9),
AEDATAFX=(30,31,32,33,34,35,36,37,38,39,FA,FB,FC,FD,FE,FF)
```

X25DIR vector

- LANDP for OS/2

Defines an entry in a X.25 directory table. These tables contain communications provider directories and their associated X.25 subscriber numbers. A table can contain up to 100 entries.

The customization program creates an ASCII run-time file, named SNAX25D.CFG, which contains that information.

One vector has to be specified for each entry in the table. All the parameters for this vector have to be specified. There is no default for them.

A quick reference

Vector position	None
List of keywords	TBLNAME, DIRNAME, SUBSNUMB
Keywords relate to	**Keyword** **Relates to keywords** **TBLNAME** SBSX25

Vector format	X25DIR TBLNAME=*xxxxxxxx*,
	DIRNAME=*xxxxxxxx*,
	SUBSNUMB=*xxxxxxxxxxxxxxx*

Keyword **Description**

TBLNAME Is a required keyword to specify the name of the X.25 directory table that contains the entry.

The parameter value is a string of up to eight alphanumeric characters, and must be unique among all X.25 directory tables.

DIRNAME Is a required keyword to specify the name of the subscriber number in the communications provider definitions.

The parameter value is a string of up to eight alphanumeric characters, and must be unique among all the entries in the X.25 directory table.

SUBSNUMB
Is a required keyword to specify the subscriber address. For international calls, the subscriber number must be preceded by the country identification and the country subcode.

The parameter value is a string of up to 15 digits, and must be unique among all the entries in the X.25 directory table.

X25DIR vector example

```
/* Vector X25DIR Example */

 X25DIR   TBLNAME=X25TBL01,
          DIRNAME=PARTNER1,
          SUBSNUMB=123456789012345
```

Appendix D. Editing configuration data

OS/2 DOS 6000 NT

This chapter provides information about:

- Data that is specific for each workgroup.

  This is called workgroup configuration data, and is specified in vector format and stored in the LANCONF.SPC files.

  Each workgroup configuration requires a separate LANCONF.SPC file, which must be located in a separate directory. See “Customization directory structure” on page 37.

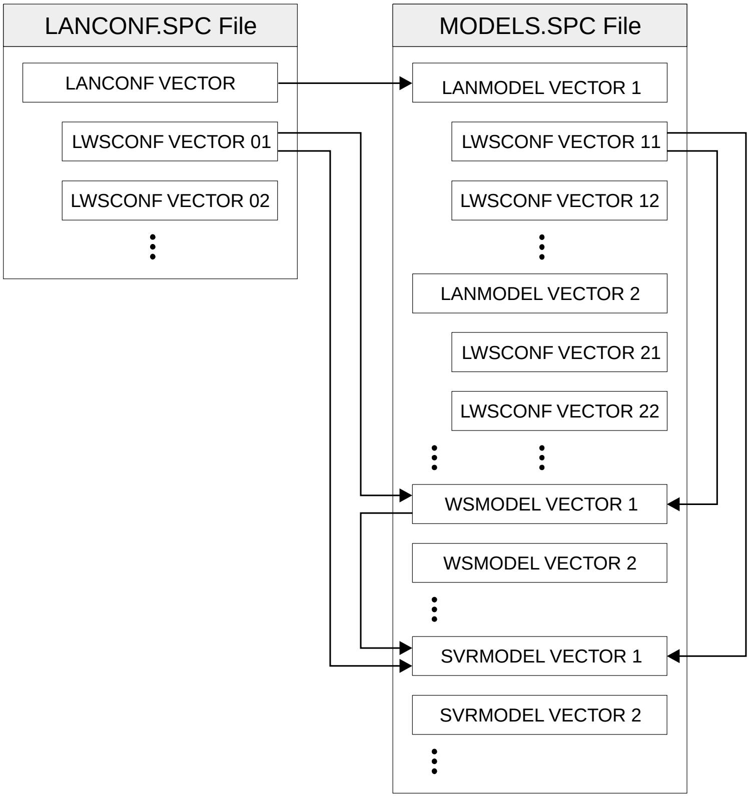
- Data for models.

  This is called model configuration data, and is specified in vector format and stored in the MODELS.SPC file located in the EHCCUS directory.

For information on how to store customization data in specification files, refer to “Customization data structure” on page 36, especially if you plan to use *embedded files* (see page 39) or *partial files* (see page 39).

The chapter also explains how to use models to define your workgroup configurations.

The following graphic shows the vectors in the .SPC files, and the references that can be specified to use the model configurations. The references are specified at *keyword* level.



Creating and editing configuration vectors

Before you start creating and editing configuration vectors, consider the following:

1. The customization program provides a procedure to edit specification files.

   You can apply this procedure to the LANCONF.SPC or MODELS.SPC files, and display and modify the configuration vectors.

   If you want to display *online information* about this procedure, from the EHCCUS directory, enter:

   ```
   EDITSPC ?
   ```

   To start the procedure for a workgroup configuration file (LANCONF.SPC), from the EHCCUS directory enter:

   ```
   EDITSPC xxxxxxxx LAN
   ```

   where *xxxxxxxx* is the name of the subdirectory in the EHCCUS directory where the LANCONF.SPC file is located.

   If the file does not exist, the procedure creates a subdirectory with the name you provide, and an empty file named LANCONF.SPC located in that subdirectory.

If the subdirectory where the LANCONF.SPC file is located is not in the EHCCUS directory, but in an other directory at the same level, you have to specify also this directory. In this case, the command to enter is:

```
EDITSPC  yyyyyyyy xxxxxxxx LAN
```

where *yyyyyyyy* is the name of a directory at the same level as EHCCUS.

If the file or the directories do not exist, the procedure creates the directories with the names you provide, and an empty file named LANCONF.SPC located in that path.

To start the procedure for the model configuration file (MODELS.SPC), from the EHCCUS directory, enter:

```
EDITSPC MODELS
```

If you use the OS/2 Enhanced Editor with the LANDP customization editing tool, the **LANDP** choice is listed in the action bar of the editor window. For information about that tool, select the **View doc** option in the pull-down that appears when you choose **LANDP**.

2. The customization program provides a procedure to generate configuration vectors from the configuration data in the internal repository.

   The configuration vectors are generated either in the LANCONF.SPC files, or in the MODELS.SPC file, located in the EHCCUS directory. The order in which the vectors appear in the files is not relevant.

   To start the procedure for the workgroup configuration vectors, from the EHCCUS directory, enter:

   ```
   GENSPEC LAN
   ```

   To start the procedure for the model configuration vectors, from the EHCCUS directory, enter:

   ```
   GENSPEC MODELS
   ```

   For further information on that procedure, refer to “Generating data” on page 243.

Editing workgroup configuration data

To edit workgroup configuration data:

- Start the EDITSPC procedure to edit a workgroup configuration file in the path you specify (see “Creating and editing configuration vectors” on page 408)

  or

- Open a file with filename LANCONF and extension SPC (LANCONF.SPC), using any text editor. The LANCONF.SPC file *must* be located in one of the CONF*XX* directories (see “Customization directory structure” on page 37).
- Write the workgroup configuration vectors on that file.
  - Edit a LANCONF vector to include information about a workgroup configuration. You will have to write one LANCONF vector for each workgroup.

The LANCONF vector can contain actual definitions, and a reference to a LANMODEL vector.

– Edit one LWSCONF vector for each workstation in the workgroup configuration, to include information about the workstation and the servers in that workstation. Those vectors *must* be placed immediately after the corresponding LANCONF vector. You will have to write one LWSCONF vector set for each workgroup.

  The LWSCONF vector can contain actual definitions, a reference to a WSMODEL vector, and some references to SVRMODEL vectors.

When editing a configuration vector, if you refer to a model configuration vector, the definitions in the model configuration vector become the default parameter values for the configuration vector you are editing. To change some values, include the corresponding keywords in the vector and assign to them the new parameter values. Note that you can include keywords that are not included in the model configuration vector.

When defining a workgroup configuration and referring to a workgroup model configuration, you may have to do some of the tasks in the following table. The table shows the tasks and the actions you have to take.

Task	**Action**
Delete a workstation from a workgroup model configuration	• Include the LWSCONF vector corresponding to that workstation, specifying DELETE as the workstation model configuration name
Add a workstation to a workgroup model configuration	• Include the WSNAMES keyword in the LANCONF vector, specifying that workstation, and • Include the LWSCONF vector corresponding to that workstation, containing the workstation information
Delete a server from a workstation model configuration	• Include the SERVER keyword in the LWSCONF vector, specifying that server, and • Specify DELETE as the server model configuration name
Add a server to a workstation model configuration	• Include the SERVER keyword in the LWSCONF vector, specifying that server • Include the server information in the SERVER keyword and, if needed, in the following keywords – PAR*XXXXX* – SES*XXXXX*
Delete access to the services provided by a functional area from a workstation model configuration	• Include the LWSCONF vector corresponding to that workstation, specifying DELETE as the alias of the resource in the CLIENT keyword

Task	Action
Delete access to the services provided by a functional area to a workstation model configuration	• Include the CLIENT keyword in the LWSCONF vector, specifying that server • Include the server information in the CLIENT keyword and, if needed, in the SES*XXXXX* keyword

When you generate the LANCONF.SPC file through the generation procedure, using the customization data stored in the internal repository, the order in which the vectors appear in the file may be different from the order in which you wrote them.

Workgroup configuration vectors - descriptions

This section provides information about each workgroup configuration vector, including examples.

LANCONF vector

Contains workgroup information, and must be the first vector in the workgroup configuration file. An example for this vector is provided under “LANCONF vector examples” on page 416.

A quick reference

Vector position	First vector in the LANCONF.SPC file. It can be followed by up to 250 LWSCONF vectors.
List of keywords	GROUP, NAME MODEL DEFAULTS, LANADAPT, LANGUAGE, RPLOAD, SUFFIX, XPORT, TCPPORT, ADAPTNUM, NETBUFF, WSNAMES, PARLIP, NWSIDDUP
Keywords relate to	**Keyword** **Relates to keywords** **MODEL** MODNAME (LANMODEL vector) **ADAPTNUM** ADAPTNUM (LWSCONF vector) **DEFAULTS** NAME (DEFAULTS vector) **WSNAMES** NAME (LWSCONF vector) **NETBUFF** NETBUFF (LWSCONF vector)

Vector format	LANCONF [GROUP=*xxxxxxxx*,] NAME=*xxxxxxxx*, [MODEL=*xxxxxxxx*,] [DEFAULTS=*xxxxxxxx*,] [LANADAPT=*x*,] [LANGUAGE=*xxx*,] [RPLOAD=*x*,] [SUFFIX=*x*,] [XPORT=*x*,] [TCPPORT=*xxxxx*,] [ADAPTNUM=*x*,] [NETBUFF=*xx*,] [NWSIDDUP=*x*,] [PARLIP=*x*,] WSNAMES=( )

The keywords in the first line in the list identify the workgroup configuration. The MODEL keyword refers to a workgroup model configuration. Other keywords define the workgroup configuration.

Keyword **Description**

GROUP Is an optional keyword to specify the name of the group of workgroup configurations to which the workgroup configuration belongs. This name must match the name of the directory where the workgroup configuration directory is located. For information on directory structure, refer to “Customization directory structure” on page 37.

The combination of the name assigned in the GROUP keyword and the name assigned in the NAME keyword must be unique for the whole installation.

The parameter value is a name of up to 8 alphanumeric characters. The default is **EHCCUS**.

NAME Is a required keyword to specify the name of the workgroup configuration. This name must match the name of the directory where the LANCONF.SPC file is located. For information on directory structure, refer to “Customization directory structure” on page 37.

The combination of the name assigned in the GROUP keyword and the name assigned in the NAME keyword must be unique for the whole installation.

The parameter value is a name of up to 8 alphanumeric characters.

MODEL Is an optional keyword to specify the name of the workgroup model configuration to be used to define the workgroup configuration.

The parameter value must match the value assigned in the MODNAME keyword in a LANMODEL vector.

DEFAULTS

Is an optional keyword to specify the set of defaults to be used in the workgroup.

The parameter value must match the value assigned in the NAME keyword in a DEFAULTS vector. The default is **GENERAL**.

LANADAPT
Is an optional keyword to specify the type of hardware adapter to be used in the workgroup. The keyword applies only when there is at least one DOS workstation in the workgroup, and NetBIOS is used as transport protocol.

The parameter value can be:

T	Token-ring
P	PC network
O	Other types (for example, Ethernet)

The default is **T**. LANDP always uses the primary adapter, not the alternate one.

LANGUAGE
Is an optional keyword to specify the identifier of the language to be used. The customization program copies the files that contain messages in the corresponding language.

The parameter value can be:

001	English
034	Spanish
086	Simplified Chinese
090	Turkish

The default is **001**. When working in DBCS mode, only English and Simplified Chinese are supported. Note that you can only specify 086 if the DBCSCTRY entry on the DEFAULTS vector is also set to 086.

RPLOAD Is an optional keyword to specify whether 4700 virtual volume remote program load is used in the workgroup. The keyword applies only to DOS workstations, and when the PC/Integrator virtual volume support is loaded in a workstation in the workgroup.

The parameter value can be Y, to use virtual volume remote program load, or N, not to use it. The default is **N**.

SUFFIX Is an optional keyword to specify that the workgroup name is to be added, as a suffix to the workstation IDs to identify the workstations when establishing the LANDP session.

If the suffix is to be used, the workgroup name must be unique for each LANDP workgroup in the same network.

The suffix enables to have workstations with the same name in different LANDP workgroups, which belong to the same network. This suffix can be changed at run-time using the LAN variable run-time parameters program (VARPARM.EXE).

The parameter value can be Y, to add the suffix, or N, not to add it. The default is **N**.

XPORT LANDP for DOS, OS/2, Windows NT, and AIX: Is an optional keyword to specify the transport protocol used to carry requests/replies through the workgroup.

The parameter value can be:

N NetBIOS
P TCP/IP (for DOS workstations, the PC/TCP implementation is used)
T TCP/IP (for DOS workstations, the IBM TCP/IP implementation is used)

The default is **N**.

For a DOS workstation:

- When the parameter value is P, customization generates code to distribute EHCLIP.PCT (renamed to EHCLIP.EXE) to the DOS workstation.
- When the parameter value is T, customization generates code to distribute EHCLIP.IBM (renamed to EHCLIP.EXE) to the DOS workstation.

For non-DOS workstations, P and T values both have the same effect as T for a DOS workstation.

TCPPORT LANDP for DOS, OS/2, Windows NT, and AIX: Is an optional keyword to specify the TCP/IP port number used by LANDP. It applies only when you assign the P or T parameter value to the XPORT keyword.

The parameter value ranges from 1024 to 65535. The default is **52699**.

ADAPTNUM

LANDP for DOS, OS/2, and Windows NT: Is an optional keyword to specify the adapter number used by all the workstations in the workgroup.

You can also specify the adapter number at workstation level, through the ADAPTNUM keyword in the LWSCONF vector. The value specified at workstation level overrides the value specified at workgroup level.

The parameter value ranges from 0 to 3. The default is provided at workstation level.

Values 2 and 3 apply only to OS/2 and Windows NT workstations. If you specify value 2 or 3 and there are DOS workstations in the workgroup, you have to modify the adapter number for these workstations in the corresponding LWSCONF vector.

For OS/2 and Windows NT workstations, you can also specify parameter values in the form: (*x*,*x*...), where *x* can range from 0 to 3, but where no number is repeated. This enables you to specify a range of adapters, thus supporting up to four adapters in a single workstation.

NETBUFF LANDP for DOS: Is an optional keyword to specify the size of the NetBIOS buffer for all the DOS workstations in the workgroup.

You can also specify the size of the NetBIOS buffer at workstation level, through the NETBUFF keyword in the LWSCONF vector. The value specified at workstation level overrides the value specified at workgroup level.

The parameter value ranges from 1 to 56. The default is provided at workstation level.

WSNAMES

Is a required keyword to specify the names of the workstations in the workgroup configuration. The keyword can specify up to 250 parameters, because this is the maximum number of workstations in a workgroup supported by the LANDP family.

Each parameter is the name of a workstation in the workgroup configuration. The parameter value is a name of up to 2 alphanumeric characters. The value must match the value assigned in the NAME keyword in the LWSCONF vector that contains information about the workstation.

NWSIDDUP

Is an optional keyword that can be used to avoid receiving return codes which indicate that a duplicate workstation identifier (WSID) has been defined.

The parameter values are Y (to suppress duplicate WSID return codes) and N (to display these return codes). The default is N.

All WSIDs across all workgroups must be unique if you distribute your LANDP software by using NetView/DM. In this case, you must take action on the return codes issued by VALSPEC which indicate that a duplicate WSID has been issued. The first six characters of each workgroup name (NAME) must be unique. The two-character workstation name (WSNAME) is appended to create the eight-character WSID. Avoid the return codes by ensuring that the first six characters of all workgroup names are unique.

If you do not distribute your LANDP software by using NetView/DM, you can suppress return codes by setting the appropriate VALSPEC parameter. Be aware, however, that this also suppresses other return codes that you might need to respond to. Alternatively, set the NWSIDDUP parameter to Y to suppress return codes for duplicate WSIDs.

PARLIP Is an optional keyword to define LANDP Internet Protocol parameters.

You can also specify these parameters at workstation level, through the PARLIP keyword in the LWSCONF vector. The value specified at workstation level overrides the value specified at workgroup level.

You can specify up to 3 parameters:

1. Availability probe datagrams.

   This parameter specifies whether availability probe datagrams will be sent at regular intervals when a session has no normal traffic.

   The parameter value can be Y, to send availability probe datagrams, or N, not to send them. The default is **Y**.

2. Space for retransmission table.

   This parameter specifies the storage, in bytes, to be allocated for the retransmission table.

   This table is used by the LANDP Internet Protocol to save information about datagrams sent, in case they should be retransmitted. The information is deleted when the receiver acknowledges reception.

   The parameter applies only to DOS workstations. The parameter value ranges from 256 to 65000.

   The default is defined using the following formula:

   ```
     Default value = 910 * ((n * 40) / (n + 34))
   ```

   where *n* is the number of related workstations.

3. LANDP Internet Protocol address checking.

   This parameter specifies whether checking for related workstations with undefined IP addresses will be carried out.

   The parameter value can be Y or N. Y checks that all LANDP Internet Protocol addresses are defined; processing stops if any undefined addresses are found. N allows some addresses to be undefined at this time; processing continues. The default is **Y**.

LANCONF vector examples

```
/* Following is the EXAMPLE 1, which corresponds to keyword      */
/* specifications for a workgroup that uses NetBIOS as transport */
/* protocol.                                                      */



         LANCONF  GROUP=EHCCUS,
                  NAME=LAN1,
                  WSNAMES=(AA,BB,CC)


/* Following is the EXAMPLE 2, which corresponds to keyword    */
/* specifications for a workgroup that uses TCP/IP as transport */
/* protocol.                                                    */

         LANCONF  GROUP=EHCCUS,
                  NAME=TCPIP1,
                  WSNAMES=(AA,BB),
                  DEFAULTS=GENERAL,
      /* Transport protocol through workgroup is TCP/IP */
                  XPORT=T,
      /* TCP/IP port used by LANDP is port 52699 */
                  TCPPORT=52699,
                  SUFFIX=Y
```

LWSCONF vector

Contains information about a particular workstation in the workgroup configuration, including functional area definitions. The keywords that define the functional areas are explained in the corresponding sections. Those sections provide examples of the keyword definitions that can be included in the LWSCONF vectors for the respective functional areas.

A quick reference

Vector position	None. Follows a LANCONF vector. There can be up to 250 LWSCONF vectors in the same LANCONF.SPC file.
List of keywords	NAME WSID, MODEL TYPE, SYSLVL, PRODLVL, SOFTPACK, FRAME, RPLOAD, POOLSIZE, PARLIP, PARLIPEX, LANDPDCE, ADAPTNUM, NETBUFF, DBCSXLAT, PARWIN, SERVER, CLIENT, PAR*xxxxx*, SES*xxxxx*
Keywords relate to	**Keyword** **Relates to keywords** **ADAPTNUM** ADAPTNUM (LANCONF vector) **MODEL** MODNAME (WSMODEL vector) **NETBUFF** NETBUFF (LANCONF vector) MODNAME (SVRMODEL vector) **PARLIP** XPORT (LANCONF vector) **PARLIPEX** XPORT (LANCONF vector)
Vector format	LWSCONF NAME=*xx*, [WSID=*xxxxxxxx*,] [MODEL=*xxxxxxxx*,] [TYPE=( ),] [SYSLVL=*x*,] [PRODLVL=*xxx*,] [SOFTPACK=*xxxxxxxx*,] [FRAME=*xxxx*,] [RPLOAD=*x*,] [POOLSIZE=*xxx*,] [PARLIP=( ),] [PARLIPEX=( ),] [LANDPDCE=*x*,] [ADAPTNUM=*x*,] [NETBUFF=*xx*,] [DBCSXLAT=*x*,] [PARWIN=*xx*,] [SERVER=( ),] [CLIENT=( ),] [PARxxxxx=( ),] [SESxxxxx=( ),]

The NAME keyword identifies the workstation in the workgroup configuration. The MODEL keyword refers to a workstation model configuration.

Next keywords define the workstation in the workgroup configuration. The PAR*xxxxx* and SES*xxxxxx* keywords specify the parameters used to define some functional areas.

Keyword **Description**

NAME Is a required keyword to specify the name of the workstation in the workgroup configuration. The name becomes the workstation ID, and the name of the directory where the customization information about the workstation is located after generating run-time files. For information on directory structure, refer to "Customization directory structure" on page 37.

The parameter value is a name of up to 2 alphanumeric characters. The value is used as the name of the workstation when you need to specify that name in other keywords, for example, because the services provided by this workstation are required in the configuration you are defining.

See the NWSIDDUP parameter of LANCONF for details of how to avoid receiving return codes about duplicate workstation identifiers.

WSID Is an optional keyword to specify the name of the workstation when distributing using a distributed server. (See Chapter 10, "Distributing software using a distribution server" on page 85).

The parameter value is a name of up to 8 alphanumeric characters, and should be unique across all workstations and workgroups. The default is the first 6 characters of the workgroup name, concatenated with the workstation name.

MODEL Is an optional keyword to specify the name of the workstation model configuration to be used to define the workstation in the workgroup configuration.

To point to a workstation model configuration, the parameter value must match the value assigned in the MODNAME keyword in a WSMODEL vector. To delete a workstation from a workgroup model configuration, the parameter value must be DELETE.

TYPE Is an optional keyword to specify the operating system installed in the workstation.

You can specify up to 2 parameters:

1. Operating system.

   The parameter value can be:

   DOS One of the following operating systems, or later versions, depending on the country:

   IBM DOS T2000 (Traditional Chinese)
   IBM DOS H2000 (Korean)
   IBM DOS P2000 (Simplified Chinese)

   OS/2 One of the following operating systems, or later versions, depending on the country:

   IBM OS/2 Warp V4

IBM OS/2 T4.0
IBM OS/2 H4.0
IBM OS/2 P4.0

NT One of the following operating systems:

Microsoft Windows NT Version 4
Microsoft Windows 2000

RS/6000 IBM AIX V. 3.2

The default is **DOS**.

2. Windows 3.1 support.

   This parameter specifies whether Windows 3.1/3.11 support is to be used. It applies only to DOS or OS/2 workstations.

   To use Windows 3.1 support, the parameter value must be WIN. To use Windows for Workgroups support, the parameter value must be WFW. This specifies that no additional LAN support needs to be added for this workstation.

   The parameter is optional. It is *not* supported for TYPE=NT (Windows NT).

SYSLVL Is an optional keyword to specify the operating system version installed in the workstation. The keyword applies only to DOS workstations.

If it is specified, the parameter value must be 5, meaning V. 5.xx.

PRODLVL Is an optional keyword to specify the LANDP version installed in the workstation.

The parameter value can be:

L50 LANDP Version 5.0 for DOS, if you define a TYPE=DOS workstation.
LANDP Version 5.0 for OS/2, if you define a TYPE=OS/2 workstation.
LANDP Version 5.0 for Windows NT, if you define a TYPE=NT workstation.

L40 LANDP Version 4.0 for DOS, if you define a TYPE=DOS workstation.
LANDP Version 4.0 for OS/2, if you define a TYPE=OS/2 workstation.
LANDP Version 4.0 for Windows NT, if you define a TYPE=NT workstation.

L30 LANDP Version 3.0 for DOS, if you define a TYPE=DOS workstation.
LANDP Version 3.0 for OS/2, if you define a TYPE=OS/2 workstation.

L21 LANDP Version 2.1 for AIX, if you define a TYPE=RS/6000 workstation.

L20 LANDP Version 2.0 for DOS, if you define a TYPE=DOS workstation.
LANDP Version 2.0 for OS/2, if you define a TYPE=OS/2 workstation.

L10 LANDP Version 1.0 for DOS, if you define a TYPE=DOS workstation.
LANDP Version 1.0 for OS/2, if you define a TYPE=OS/2 workstation.
LANDP Version 1.0 for AIX, if you define a TYPE=RS/6000 workstation.

F31 FBSS Version 3.1, if you define a TYPE=DOS workstation.

F11 FBSS/2 Version 1.1, if you define a TYPE=OS/2 workstation.

The default is the latest LANDP product version for the operating system installed in the workstation you define.

SOFTPACK
Is an optional keyword to specify the name of a pack of files to be distributed with the LANDP runtime files for this workstation. This name must be defined in the PACKNAME parameter of a SOFTPACK vector in COMMON.SPC.

FRAME Is an optional keyword to specify the segment address of a 64KB frame of free memory space that is needed to load servers in *expanded memory*. The keyword applies only to DOS workstations.

The parameter value ranges from C000 to E000, in increments of 400(hex). The default is **C000**.

RPLOAD Is an optional keyword to specify whether 4700 virtual volume remote program load is used in the workstation. The keyword applies only to DOS workstations, and when the PC/Integrator virtual volume support is loaded in a workstation in the workgroup.

The parameter value can be Y, to use virtual volume remote program load, or N, not to use it. The default is **N**.

POOLSIZE
LANDP for DOS: Is an optional keyword to specify the size of the buffer pool, in KB. This means the number of KB that should be reserved for the internal buffer pool to allocate incoming/outgoing requests from/to remote workstations, and requests from servers loaded into expanded memory to servers that are also loaded into expanded memory.

The parameter value ranges from 7 to 512. The value is limited by the available memory.

The default is [1.25 x *n*], where *n* is the number of workstations that receive services from the workstation you are defining, or provide services to it.

For workstations that provide or receive shared DOS directory services, the customization program recalculates the default accordingly.

PARLIP LANDP for DOS, LANDP for OS/2, LANDP for Windows NT, LANDP for AIX: Is an optional keyword to define LANDP Internet Protocol parameters. This keyword applies only if you have selected TCP/IP as transport protocol in the LANCONF vector (XPORT=T).

You can specify up to 3 parameters:

1. Availability probe datagrams.

   This parameter specifies whether availability probe datagrams will be sent at regular intervals when a session has no normal traffic.

   The parameter value can be Y, to send availability probe datagrams, or N, not to send them. The default is **Y**.

2. Space for retransmission table.

   This parameter specifies the storage, in bytes, to be allocated for the retransmission table.

   This table is used by the LANDP Internet Protocol to save information about datagrams sent, in case they should be retransmitted. The information is deleted when the receiver acknowledges reception.

   The parameter applies only to DOS workstations. The parameter value ranges from 256 to 65000.

   The default is defined using the following formula:

   ```
   Default value = 910 * ((n * 40) / (n + 34))
   ```

   where *n* is the number of related workstations.

3. LANDP Internet Protocol address checking.

   This parameter specifies whether checking for related workstations with undefined IP addresses will be carried out.

   The parameter value can be Y or N. Y checks that all LANDP Internet Protocol addresses are defined; processing stops if any undefined addresses are found. N allows some addresses to be undefined at this time; processing continues. The default is **Y**.

PARLIPEX

LANDP for DOS: Is an optional keyword to define LANDP Internet Protocol parameters. This keyword applies only if you have selected TCP/IP as transport protocol in the LANCONF vector (XPORT=T).

You can specify one parameter:

1. Expanded memory selection.

   The parameter value can be:

   Y The LANDP Internet Protocol is loaded in *expanded memory*
   N The LANDP Internet Protocol is loaded in *conventional memory*

   The default is **N**.

LANDPDCE

LANDP for AIX: Is an optional keyword to specify whether the workstation provides LANDP-DCE interoperability.

The parameter value can be Y, to provide LANDP-DCE interoperability, or N, not to provide it. The default is **N**.

ADAPTNUM

LANDP for DOS, LANDP for OS/2, LANDP for Windows NT: Is an optional keyword to specify the adapter number used by the workstation.

You can also specify the adapter number at workgroup level, through the ADAPTNUM keyword in the LANCONF vector. The value specified at workstation level overrides the value specified at workgroup level.

The parameter value ranges from 0 to 3. The default is **0**, or the value specified at workgroup level.

Values 2 and 3 apply only to OS/2 and Windows NT workstations. If you specify value 2 or 3 at workgroup level, when you define a DOS workstation in the workgroup, you have to specify value 0 or 1 through this keyword.

For OS/2 and Windows NT workstations, you can also specify parameter values in the form: `(`*x*`,`*x*`...)`, where *x* can range from 0 to 3, but where no number is repeated. This enables you to specify a range of adapters, thus supporting up to four adapters in a single workstation.

NETBUFF LANDP for DOS: Is an optional keyword to specify the size of the NetBIOS buffer, in KB. This means the number of KB that should be reserved for the internal buffer to be shared with the NetBIOS manager layer in order to receive data from remote workstations.

You can also specify the size of the NetBIOS buffer at workgroup level, through the NETBUFF keyword in the LANCONF vector. The value specified at workstation level overrides the value specified at workgroup level.

The parameter value ranges from 1 to 56. The default is **4**, or the value specified at workgroup level.

DBCSXLAT

LANDP for DOS (DBCS mode): Is an optional keyword to specify whether the ASCII-EBCDIC translation server will be loaded.

The parameter value can be:

N The server will not be loaded.
Y The server will be loaded in conventional memory.
E The server will be loaded in expanded memory.

The default is **N**.

PARWIN LANDP for DOS: Is an optional keyword to specify the maximum storage, in KB, required by Windows 3.1 parameters and user data.

The parameter value ranges from 4 to 56. The customization data provides no default.

SERVER Is an optional keyword to define some functional areas to be loaded in a workstation. Each functional area requires a SERVER keyword. At least one SERVER or CLIENT keyword is required in each LWSCONF vector.

You can specify up to 4 parameters:

1. Server name used by the applications.

   The parameter is required. The parameter value is a string of up to 8 alphanumeric characters.

   Refer to the table in “Vectors - a quick reference” on page 54 for the values (string in parenthesis) corresponding to the functional areas.

   The following servers require that the server suffix (##) is substituted by the corresponding value to completely identify the server:

   EHCMQ##
   EHCODB##
   EHCSFD##
   EHCSFR##
   EHCSQL##
   ELECJO##
   MSRE47##
   PINP47##
   SHFILE##

   Note: If the PIN pad or MSR/E server is loaded in an RS/6000 workstation, the server suffix in PINP47## or MSRE47## must not be substituted.

   The following servers require a suffix to identify the session to be used:

   BIWP*x*
   BPP*x*
   EMU3270*x*
   LDA7*x*

   Note that BIWP and LDA 7 program need the suffix only when they are installed in a DOS workstation.

   If the following servers are to be run in OS/2 MVDM, the server names to be specified are:

BIWP	VBIWP*x*
LDA 7 program	VLDA7*x*

   where *x* is the suffix to identify the session.

   To specify user servers, the parameter value must be the value specified in the NAME keyword of the DEFSERV vector. Those vectors are located in the COMMON.SPC file. See Appendix B, “User servers” on page 275.

   This also applies to DCE servers, which are defined as user servers. All the DCE servers must be specified as if they were to be loaded in the RS/6000 type workstation in the LANDP workgroup that provides LANDP access to DCE services.

2. Model name.

The parameter is optional. The parameter value is a string of up to 8 alphanumeric characters, and must match the value assigned in the MODNAME keyword in a SVRMODEL vector.

To delete a server from a workstation model, the parameter value must be DELETE.

3. Expanded memory selection.

   The parameter applies only to DOS workstations. It is optional. The parameter value can be:

   Y The server is loaded in *expanded memory*
   N The server is loaded in *conventional memory*

   The default is **N**.

4. Loading parameters.

   This parameter specifies a string that will be added to the loading statement of the server.

   The parameter is optional. The parameter value is a string of up to 40 characters, enclosed between quotes.

CLIENT Is an optional keyword to define a functional area to be used by a workstation. At least one SERVER or CLIENT keyword is required in each LWSCONF vector.

You can specify up to 3 parameters:

1. Name of the resource providing services. The name specified can be used by the applications, when requesting services.

   The following resources are always local, and do not require this parameter:

   BIWP
   BMLS
   BPP
   DCADLC
   DDT
   EHCCOMP
   EHCLRMG
   EHCSAM
   EHCTCP
   EHCTRACW
   EHCVDMGR
   EMU3270
   EMU3287
   FORWARD
   LDA7
   OPER
   PBMS
   PRTMGR
   PT4721

RCMS
RDVVOLS
SDLC
SFQUERY
SMOP
TRDLC
VFILE
X25DLC
X25DLC2

In any other case, the parameter is required. The parameter value is a string of up to 8 alphanumeric characters.

Refer to the table in “Vectors - a quick reference” on page 54 for the values (string in parenthesis) corresponding to the functional areas.

The following servers require that the server suffix (##) is substituted by the corresponding value to completely identify the server:

EHCODB##
EHCSFD##
EHCSFR##
EHCSQL##
ELECJO##
MSRE47##
PINP47##
PR47X2##
PR4770##
SHFILE##

In the following resources, the suffix identifies the session used by the server:

EHCMQ##
MSRE47##
PINP47##
PR47X2##
PR4748##
PR4770##
SNA##
X25NAT##

In the LANDP for AIX PPC#### resource, the first two # positions identify the group ID. The last two # positions correspond to the values for the dynamically used and chosen application conversation IDs.

To specify user servers, the parameter value must be the value specified in the NAME keyword of the DEFSERV vector. Those vectors are located in the COMMON.SPC file. See Appendix B, “User servers” on page 275.

This also applies to DCE servers, which are defined as user servers.

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. To customize for the modified interface, use a server name of just SNA, instead of SNA followed by the session identifier, in the CLIENT keyword of the LWSCONF vector. Only one CLIENT=(SNA,xx) keyword is required, no matter how many sessions are to be used

2. Name of the workstation where the resource is located.

   The parameter is required. The parameter value must match the value assigned in the NAME keyword in the LWSCONF vector corresponding to the workstation where the server is loaded.

   When you define a DCE client, the LANDP resource must be specified as if it was located in the RS/6000 type workstation in the LANDP workgroup that provides DCE access to LANDP services.

   When you define a workstation model configuration, and some services are locally provided, you can specify &W as the name of the workstation where the corresponding server is loaded. When you assign the &W value in the CLIENT keyword in a WSMODEL vector, the customization program substitutes the value &W for the name of the workstation that uses the model.

3. Alias of the resource. It can be used by the applications, when requesting services, as if it were the name of the resource.

   The parameter applies only to LANDP for OS/2, Windows NT, and AIX. workstations. The parameter is optional. The parameter value is a string of up to 8 alphanumeric characters. It can be **neither** the name of a resource **nor** the alias of another resource used in the workstation.

   This parameter is also used to delete a client from a workstation model. The parameter value must be DELETE, and is applicable to any type of workstation in this case.

PAR*xxxxx*, SES*xxxxx*

Specify the parameters used to define some functional areas. The *xxxxx* string stands for the functional area identifier; for example, &3270 identifies the 3270 emulator.

The PAR*xxxxxx* keywords are used to specify all the parameters for those servers that are loaded only once in the workstation, and handle a single resource. For servers that are loaded more than once in the workstation or handle multiple resources, the PAR*xxxxx* keywords only specify the common information. In this case, the information specific for each particular server or resource is specified using the SES*xxxxxx* keywords.

The PAR*xxxxx* and SES*xxxxx* keywords corresponding to all the functional areas are explained in the following sections.

LWSCONF vector examples

```
/* Following is EXAMPLE 1, which corresponds to keyword     */
/* specifications for a workstation that runs DOS.          */

         LWSCONF  NAME=D1,
                  TYPE=DOS,
                  PRODLVL=L50,
                  SOFTPACK=SHDIRC,
                  FRAME=C000,
                  SERVER=DDT,
                  SERVER=(EMU32701,,Y),
                  PAR&3270=(N,N),
                  SES&3270=(D1,1,P1,,AT1,DI1,KBD),
                  SERVER=(EMU32702,,Y),
                  SES&3270=(D1,2,P1,,AT1,DIS,KBD,HOST2,43,80),
                  SERVER=(EMU32703,,Y),
                  SES&3270=(O1,3,P1,,ATR,DIS,KBD,HOST3,27,132),
                  SERVER=(EMU32704,,Y),
                  SES&3270=(O1,4,P2,,ATR,DIS,KBD,HOST4),
                  SERVER=(OPER,,Y),
                  SERVER=SDLC,
                  PAR&SDLC=(20,01,N,Y,N,03D,01234,TURN,65,C1C1C1C1C1),
                  SERVER=(SNA##,,Y),
                  PAR&SNA=(APPL,SRV,,,LUPOOL1,Y)

/* Following is EXAMPLE 2, which corresponds to keyword     */
/* specifications for a workstation that runs OS/2.         */

         LWSCONF  NAME=O1,
                  TYPE=(OS/2),
                  PRODLVL=L50,
                  SERVER=SNA##,
                  PAR&SNA=(APPL,SRV),
                  SERVER=(EHCVDMGR,,N),
                  SERVER=EMU32701,
                  PAR&3270=(N,Y),
                  SES&3270=(O1,1,,,ATR,DIS,KBD,HOST1),
                  SERVER=EMU32702,
                  SES&3270=(O1,2,,,ATR,DIS,KBD,HOST2)
```

BIWP definitions

- PC/Integrator
- PC Integrator/2

The Banking Interactive Workstation Program (BIWP) requires one PAR&BIWP keyword.

If BIWP will run in an OS/2 MVDM, the name of the server in the SERVER keyword must be VBIWP*x* instead of BIWP*x*. Additionally, the keyword to be included in the

LWSCONF, WSMODEL, or SVRMODEL vector, must be PAR&VBIW instead of PAR&BIWP. The parameters you can specify in a PAR&VBIW keyword are the same as those that can be specified in a PAR&BIWP keyword.

Keyword **Description**

PAR&BIWP

Is a required keyword to define BIWP parameters, when BIWP is specified in a SERVER keyword.

You can specify up to 19 parameters:

1. Name of the workstation providing SNA services.

   The parameter is required. The parameter value is a string of up to 2 alphanumeric characters.

2. BIWP emulated session number.

   When BIWP is installed in a DOS workstation or is used in an OS/2 MVDM, the parameter is required. The parameter value ranges from 1 to 5, and must be unique for the workstation where BIWP is installed.

   The parameter must be omitted when BIWP is installed in a OS/2 workstation. The parameter value is defaulted to **1**.

3. SNA LU pool ID or LU number.

   To use the SNA session with LU pooling support, specify the LU pool ID. The parameter value is a string of two characters: the first character must be alphabetical, and the second one alphanumeric.

   To use the session without LU pooling support, when the SNA services are provided from a DOS workstation, specify the LU number. The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit.

   Omit the parameter to use the session without LU pooling support when the SNA services are provided from an OS/2, Windows NT, or RS/6000-type workstation.

4. DLC type used by the workstation providing SNA services.

   The parameter applies only to DOS workstations. In this case, it is required except for workstations that have one DLC only.

   The parameter value can be:

   SDL Synchronous Data Link Control (SDLC)
   TKR Token-ring
   DCA Device Cluster Attachment (DCA)

5. Cursor usage.

   This parameter defines how the cursor will work on the screen.

   The parameter is optional. The parameter value can be:

Y To make the cursor remain on the screen from the first I/O operation until switching off the screen

R To make the cursor remain on the screen while a read operation is in progress

N Not to use the cursor

The default is **R**.

6. Read operation usage.

   This parameter defines how a read operation will be posted when the input segment is full and a character is typed.

   The parameter is optional. The parameter value can be Y, to post the read operation complete without status, or N, to post it with a wrong length status. The default is **Y**.

7. Default end of message (EOM) mask.

   The parameter is optional. The parameter value is a hexadecimal number of two digits. The default is **FF**.

8. Indicator 1 alarm selection.

   The parameter is optional. The parameter value can be Y, to make the alarm sound when the indicator 1 of the BIWP display is turned on, or N, not to make it sound. The default is **N**.

9. Indicator 2 alarm selection.

   The parameter is optional. The parameter value can be Y, to make the alarm sound when the indicator 2 of the BIWP display is turned on, or N, not to make it sound. The default is **N**.

10. Indicator 3 alarm selection.

    The parameter is optional. The parameter value can be Y, to make the alarm sound when the indicator 3 of the BIWP display is turned on, or N, not to make it sound. The default is **N**.

11. SYSTEM indicator alarm selection.

    The parameter is optional. The parameter value can be Y, to make the alarm sound when the SYSTEM indicator of the BIWP display is turned on, or N, not to make it sound. The default is **N**.

12. MSR/E ready alarm selection.

    The parameter applies only when BIWP uses IBM 4717 MSR/E. It is optional.

    The parameter value can be Y, to make the alarm sound when MSR/E is ready to encode, or N, not to make it sound. The default is **N**.

13. Display color attributes.

    This parameter identifies the color attributes table used for the workstation display.

The parameter value is a string of three alphanumeric characters, which is the identifier of the color attributes table. It must match the value assigned in the EXTEN keyword in a DISPLATT vector with TYPE = BIWP.

14. Display translation table.

    This parameter identifies the table that is used to translate from 4700 system EBCDIC to personal computer system ASCII, when receiving from the 4700 system.

    The parameter value is a string of three alphanumeric characters, which is the identifier of the translation table. It must match the value assigned in the EXTEN keyword in a XLATETBL vector with TYPE = EABIWP.

15. Keyboard translation table.

    This parameter identifies the translation table used for the workstation keyboard.

    The parameter value is a string of three alphanumeric characters, which is the identifier of the translation table. It must match the value assigned in the EXTEN keyword in a KBDBIWP vector.

16. PIN pad input table.

    This parameter identifies the table used by BIWP to read the input provided through the PIN pad.

    The parameter applies only when BIWP uses IBM 4718 PIN Pad. The parameter value is a string of three alphanumeric characters, which is the identifier of the PIN pad input table. It must match the value assigned in the EXTEN keyword in a PINPTBL vector.

17. MSR/E input table.

    This parameter identifies the table used by BIWP to read the input provided through the MSR/E.

    The parameter applies only when BIWP uses IBM 4717 MSR/E. The parameter value is a string of three alphanumeric characters, which is the identifier of the MSR/E input table. It must match the value assigned in the EXTEN keyword in a MSRINTBL vector.

18. MSR/E output table.

    This parameter identifies the table used by BIWP to encode data to the MSR/E.

    The parameter applies only when BIWP uses IBM 4717 MSR/E. The parameter value is a string of three alphanumeric characters, which is the identifier of the MSR/E output table. It must match the value assigned in the EXTEN keyword in a MSROUTBL vector.

19. Application session identification.

This parameter identifies the session that is used for the application program.

The parameter is optional. If specified, it overrides the value assigned in the E3270HKx keyword in the DEFAULTS vector. The parameter value is a string of up to 8 alphanumeric characters.

BIWP example

```
/* Banking Interactive Workstation Program server definition.
   Example. */
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  PRODLVL=L50,
                  SERVER=(SNA##,,N),
                  PAR&SNA=(APPL,SRV),
                  SERVER=(TRDLC),
                  PAR&TKR=(48,04,04,00999999,00111111,017,00000),
                  SERVER=(BIWP1),
                  PAR&BIWP=(AA,1,013,TKR,R,Y,FF,
                           N,N,N,N,N,ATR,DIS,KBD)

         LWSCONF  NAME=BB,
                  TYPE=OS2,
                  PRODLVL=L50,
/* Banking Interactive Workstation Program server definition.
   Example. */
                  SERVER=(BIWP1),
                  PAR&BIWP=(AA,1,014,TKR,R,Y,FF,
                           N,N,N,N,N,ATR,DIS,KBD)
```

Banking printer program definitions

- PC/Integrator
- PC Integrator/2

The banking printer program (BPP) requires one SES&BPP keyword per session.

Keyword **Description**

SES&BPP Is a required keyword to define a banking printer program (BPP) session, when BPP is specified in a SERVER keyword.

You can specify up to 7 parameters:

1. Name of the workstation providing SNA services.

   The parameter is required. The parameter value is a string of up to 2 alphanumeric characters.

2. Printer port number.

When BPP is installed in a DOS workstation, the parameter value ranges from 1 to 4. When BPP is installed in a OS/2 workstation, the parameter value ranges from 1 to 3.

3. SNA LU pool ID or LU number.

   To use the SNA session with LU pooling support, specify the LU pool ID. The parameter value is a string of two characters: the first character must be alphabetical, and the second one alphanumeric.

   To use the session without LU pooling support, when the SNA services are provided from a DOS workstation, specify the LU number. The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit.

   Omit the parameter to use the session without LU pooling support when the SNA services are provided from an OS/2, Windows NT, or RS/6000-type workstation.

4. DLC type used by the workstation providing SNA services.

   The parameter applies only to DOS workstations. In this case, it is required except for workstations that have one DLC only.

   The parameter value can be:

   SDL Synchronous Data Link Control (SDLC)
   TKR Token-ring
   DCA Device Cluster Attachment (DCA)

5. Translation table.

   This parameter identifies the table that is used to translate from 4700 system EBCDIC to ASCII, when transmitting to the printer.

   The parameter value is the extension of the file containing the EBCDIC to ASCII translation table. It must match the value assigned in the EXTEN keyword in the XLATETBL vector.

6. Translation table dynamic change selection.

   The parameter is optional. The parameter value can be Y, to use this capability, or N, not to use it. The default is **N**.

7. Host application session identification.

   The parameter is optional. If specified, it overrides the value assigned in the BPPSIx keyword in the DEFAULTS vector. The parameter value is a string of up to 8 alphanumeric characters.

Banking printer program example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  PRODLVL=L50,
                  SERVER=(SNA##,,N),
                  PAR&SNA=(APPL,SRV),
                  SERVER=(TRDLC),
                  PAR&TKR=(48,04,04,00999999,00111111,017,00000),
                  SERVER=(PR47X2##),
                  PAR&47X2=(N,C,9600,4722,N,C,9600,4722,Y),
/* Banking Printer Program server definition. Example. */
                  SERVER=(BPP1),
                  SES&BPP=(AA,1,019,TKR,017,N)

         LWSCONF  NAME=BB,
                  TYPE=OS/2,
                  PRODLVL=L50,
                  SERVER=(PR47X2##),
                  PAR&47X2=(N,C,9600,4722,N,C,9600,4722,Y),
/* Banking Printer Program server definition. Example. */
                  SERVER=(BPP1),
                  SES&BPP=(AA,1,020,TKR,017,N)
```

Batch machine loader server definitions

- LANDP for OS/2

The batch machine loader server requires one PAR&BMLS keyword.

Keyword **Description**

PAR&BMLS

Is a required keyword to define the batch machine loader server parameters, when the server is specified in a SERVER keyword.

You can specify up to 3 parameters:

1. Program name.

   This parameter specifies the name of the program called by the batch machine loader server when a message is pending in the message queue.

   The parameter is required. The parameter value has the following format:

   ```
   nnnnnnnn.eee
   ```

   where:

   nnnnnnnn Is the filename

   eee Is the extension of the file that contains the program.

2. Public object post box user name.

   The parameter is optional. The parameter value is a string of up to 8 alphanumeric characters. The default is the filename of the client program (parameter 1).

3. Path.

   This parameter specifies the path where the client program runs.

   The parameter value is a string of up to 30 alphanumeric characters. The format must be:

   ```
   D:\[directory1\[directory2\[directory3\]]]
   ```

   A maximum of three levels is permitted for the path.

Batch machine loader server example

```
         LWSCONF  NAME=AA,
                  TYPE=OS/2,
                  SERVER=SMGR,
                  PAR&SMGR=(,,,SMGRPRF,C:\SMGR\,C:\SMGR\,C:\SMGR\,C:\SMGR\),
/* Batch machine loader server definition.  Example */
                  SERVER=BMLS,
                  PAR&BMLS=(MESSAGP.EXE,MESSAGEP,C:\PATHM\),
/* Object post box client definition. Example */
                  CLIENT=(OPBS,BB)

         LWSCONF  NAME=BB,
                  TYPE=OS/2,
                  SERVER=SHFILE01,
                  PAR&SHFL=(PROFSFS,3,8,Y,0),
/* Object post box server definition. Example */
                  SERVER=OPBS,
                  PAR&OPBS=(C:\OPBSDB\),
                  CLIENT=(SHFILE01,BB)
```

Device cluster attachment DLC definitions

- PC/Integrator

The DCADLC server requires one PAR&DCA keyword.

Keyword **Description**

PAR&DCA Is a required keyword to define the DCADLC server parameters, when the server is specified in a SERVER keyword.

You can specify up to 3 parameters:

1. Number of buffers.

   The parameter is optional. The parameter value ranges from 10 to 100. (1 buffer = 272 bytes in memory, 256 bytes of data.) The default is **64**.

2. Maximum number of retries.

   This parameter specifies the number of retries after negative responses that are admitted before resetting the DCA card.

   The parameter is optional. The parameter value ranges from 1 to 100. The default is **10**.

3. Time-out.

   This parameter specifies the seconds after a non-answered request, that are admitted before resetting the DCA card.

   The parameter is optional. The parameter value ranges from 1 to 60. The default is **30**.

Device cluster attachment DLC example

```
         LWSCONF  NAME=AA,
                  TYPE=OS/2,
                  PRODLVL=L150,
                  SERVER=(SNA##,,N),
                  PAR&SNA=(APPL,SRV),
                  CLIENT=(SNA01,AA),
                  SES&SNA=(AA,01,010,DCA),
/* DCA server definition. Example. */
                  SERVER=(DCADLC),
                  PAR&DCA=(64,10,30)
```

Electronic journal server definitions

- LANDP for DOS
- LANDP for OS/2
- LANDP for AIX
- LANDP for Windows NT

The electronic journal server requires one PAR&EJOU keyword.

Keyword **Description**

PAR&EJOU
Is a required keyword to define the electronic journal server parameters, when the server is specified in a SERVER keyword.

You can specify up to 2 parameters:

1. Profile name.

   If only one electronic journal server profile is defined, the parameter is optional. The parameter value is a string of up to 8 alphanumeric characters, and must match the value assigned in the NAME keyword in the EJOUPRF vector.

2. Maximum record size.

   The parameter is optional. The parameter value ranges from 1 to 4, meaning the size in KBs. If that keyword is not specified, the default is **1**.

Electronic journal server example

```
         LWSCONF  NAME=AA,
                  TYPE=OS/2,
                  SERVER=SMGR,
                  PAR&SMGR=(,,,SMGRPRF,C:\SMGR\,C:\SMGR\,C:\SMGR\,C:\SMGR\),
                  SERVER=(SHFILEBA),
                  PAR&SHFL=(PROFSFS,10,3,Y),
                  SERVER=(SFQUERY),
                  CLIENT=(SHFILEBA,AA),
/* Electronic journal server definition. Example. */
                  SERVER=(ELECJO01),
                  PAR&EJOU=(PROFEJOU,1)

         LWSCONF  NAME=BB,
                  TYPE=DOS,
/* Electronic journal client definition. Example. */
                  CLIENT=(ELECJO01,AA)
```

Forwarding server definitions

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- LANDP for AIX

The forwarding server requires one PAR&FORW keyword, and as many SES&FORW keywords as forwarding sessions are defined in the profile used.

Keyword **Description**

PAR&FORW

Is a required keyword to define the forwarding server parameters, when the server is specified in a SERVER keyword.

You can specify up to 3 parameters:

1. Profile name.

   If only one forwarding server profile is defined, the parameter is optional. The parameter value is a string of up to 8 alphanumeric characters, and must match the value assigned in the NAME keyword in the FORWPRF vector.

2. Translation table.

   This parameter identifies the table that is used to translate from personal computer system ASCII to host computer EBCDIC, when transmitting from the workstation to the host computer.

   The parameter does not apply to DBCS mode. It is optional, if only one translation table is defined. The parameter value is the name of the translation table. It must match the value assigned in a XLATETBL vector with TYPE = AEFORW.

3. Maximum record size.

   The parameter is optional. The parameter value ranges from 1 to 4, meaning the size in KBs. The default is **1**.

SES&FORW

Is a required keyword to define a forwarding server session, when the server is specified in a SERVER keyword. One keyword is required for each forwarding session defined.

You can specify up to 5 parameters:

1. Name of the workstation providing SNA services.

   The parameter is required. The parameter value is a string of up to 2 alphanumeric characters.

2. Forwarding session number.

   The parameter is required. The parameter value ranges from 1 to 3, and must match the value assigned in the SESSION keyword in the FORWDS vector.

3. SNA LU pool ID or LU number.

   To use the SNA session with LU pooling support, specify the LU pool ID. The parameter value is a string of two characters: the first character must be alphabetical, and the second one alphanumeric.

   To use the session without LU pooling support, when the SNA services are provided from a DOS workstation, specify the LU number. The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit.

   Omit the parameter to use the session without LU pooling support when the SNA services are provided from an OS/2, Windows NT, or RS/6000-type workstation.

4. DLC type used by the workstation providing SNA services for forwarding session.

   The parameter applies only to DOS workstations. In this case, it is required except for workstations that have one DLC only that is not X.25.

   For workstations with more than one DLC other than X.25, to specify the type of DLC used, the parameter value can be:

   SDL Synchronous Data Link Control (SDLC)
   TKR Token-ring
   DCA Device Cluster Attachment (DCA)

   For workstations with DLC being X.25, the parameter value is the number you have assigned as the virtual circuit definition identification in a SES&SPVC keyword or SES&SSVC keyword.

5. Session device name.

This parameter specifies the name of the device on the AS/400 that is used for communication purposes.

The parameter applies only when the SNA services are provided by a OS/400 system.

The parameter is optional. The parameter value is a string of up to 10 alphanumeric characters.

The default is **QCFBZZFOR***x*. Note that *x* is the forwarding session number, ranging from 1 to 3, specified in the second parameter.

Forwarding server example

```
         LWSCONF  NAME=AA,
                  TYPE=OS/2,
                  SERVER=(SNA##,,N),
                  PAR&SNA=(APPL,SRV),
                  SERVER=(SHFILEBA),
                  PAR&SHFL=(PROFSFS,10,3,Y),
                  CLIENT=(SHFILEBA,AA),
                  SERVER=SMGR,
                  PAR&SMGR=(,,,SMGRPRF,C:\SMGR\,C:\SMGR\,C:\SMGR\,C:\SMGR\),
                  SERVER=(SFQUERY),
                  SERVER=(SFORFORW),
                  PAR&SFOR=(PROFSTOR,1),
                  CLIENT=(SFORFORW,AA),
/* Forwarding server definition. Example. */
                  SERVER=(FORWARD),
                  PAR&FORW=(PROFFORW,OUT,1),
                  SES&FORW=(AA,1)
```

Logical device address 7 program definitions

- PC/Integrator
- PC Integrator/2

The logical device address (LDA) 7 program requires one PAR&LDA7 keyword.

If the LDA 7 program will run in an OS/2 MVDM, the name of the server in the SERVER keyword must be VLDA7*x* instead of LDA7*x*. Additionally, , the keyword to be included in the LWSCONF, WSMODEL, or SVRMODEL vector must be PAR&VLDA instead of PAR&LDA7. The parameters you can specify in a PAR&VLDA keyword are the same as those that can be specified in a PAR&LDA7 keyword.

Keyword **Description**

PAR&LDA7

Is a required keyword to define the LDA 7 program parameters, when the program is specified in a SERVER keyword.

You can specify up to 4 parameters:

1. Name of the workstation providing SNA services.

   The parameter is required. The parameter value is a string of up to 2 alphanumeric characters.

2. BIWP emulated session number.

   When the LDA 7 program is installed in a DOS workstation or is used in OS/2 MVDM, the parameter is required.

   The parameter value ranges from 1 to 5, and must be unique for the workstation where BIWP and the LDA 7 program are installed. It must match the value assigned as the BIWP emulated session number (parameter 2) in the PAR&BIWP keyword.

   When the LDA 7 program is installed in a OS/2 workstation, the parameter must be omitted. The default is **1**.

3. SNA LU pool ID or LU number.

   To use the SNA session with LU pooling support, specify the LU pool ID. The parameter value is a string of two characters: the first character must be alphabetical, and the second one alphanumeric.

   To use the session without LU pooling support, when the SNA services are provided from a DOS workstation, specify the LU number. The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit.

   Omit the parameter to use the session without LU pooling support when the SNA services are provided from an OS/2, Windows NT, or RS/6000-type workstation.

4. DLC type used by the workstation providing SNA services.

   The parameter applies only to DOS workstations. In this case, it is required except for workstations that have one DLC only.

   The parameter value can be:

SDL	Synchronous Data Link Control (SDLC)
TKR	Token-ring
DCA	Device Cluster Attachment (DCA)

Logical device address 7 program example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  PRODLVL=L50,
                  SERVER=(SNA##,,N),
                  PAR&SNA=(APPL,SRV),
                  SERVER=(TRDLC),
                  PAR&TKR=(48,04,04,00999999,00111111,017,00000),
                  SERVER=(BIWP3),
                  PAR&BIWP=(AA,3,013,TKR,R,Y,FF,
                           N,N,N,N,N,ATR,DIS,KBD),
/* Logical Device Address 7 Program server definition. Example. */
                  SERVER=(LDA73),
                  PAR&LDA7=(AA,3,011,TKR)

         LWSCONF  NAME=BB,
                  TYPE=DOS,
                  PRODLVL=L50,
                  SERVER=(BIWP3),
                  PAR&BIWP=(AA,3,014,TKR,R,Y,FF,
                           N,N,N,N,N,ATR,DIS,KBD),
/* Logical Device Address 7 Program server definition. Example. */
                  SERVER=(LDA73),
                  PAR&LDA7=(AA,3,012,TKR)
```

MQSeries Link server definitions

- LANDP for OS/2
- LANDP for Windows NT

The LANDP MQSeries Link server requires one PAR&MQ keyword.

Keyword **Description**

PAR&MQ Is a required keyword to define the MQSeries server parameters, when the server is specified in a SERVER keyword.

You can specify up to 6 parameters:

1. MQSeries queue manager name.

   This parameter specifies the name of the queue manager to which the server is to connect. The parameter is a string of up to 24 bytes. All alphabetical characters are processed as upper case. Special characters '_' and '.' can be used in the name. Alternatively, this parameter can define an environment variable, for example, '%MQMNAME%'. If this parameter is omitted, EHCMQ## connects to the MQSeries default queue manager.

2. Maximum message length.

   This parameter specifies the maximum message length permitted by the MQSeries Link server. This is used to restrict the maximum message length to something less than or equal to the default of 57000 bytes. It can have a value between 1000 and 57000.

3. Message detail level.

   This parameter specifies the message detail level to be written to the log file. If this parameter is omitted, the default is for logging to be off. The values E|W|I can be specified where E=error level, W=warning level and I=information level. I includes levels W and E, W includes level E.

4. Log file path.

   This data string specifies the log file path, which can include a drive identifier if required. The log file is where the session log files are created.

   The data string must be less than 30 bytes in length.

   If the data string is omitted, default values are used. The drive defaults to the drive of the current working directory for the EHCMQ## process. If a drive is specified, a path should also be defined. When the path is omitted it defaults to `'EHCMQLOG'`. If both drive and path are omitted `'EHCMQLOG'` is created as a subdirectory of the current working directory for the EHCMQ## process.

   A log file is created for each session. The log file is named as XXhhmmss.ddd where:

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

5. Total number of permitted sessions.

   This parameter specifies the maximum number of different sessions the server can process at the same time. The parameter, which is optional, can be given a value in the range from 1 to 64. The default is **32**.

6. Number of MQ connections.

   This parameter specifies the number of MQ connections that should be made at startup. Every session requires a MQ connection, which can be made at startup or as session requests are received. The parameter value can be in the range 0 to the total number of permitted sessions. The parameter is optional with a default value of 0. When this option is non-zero, the LOADER server's timeout probably needs to be increased. Refer to "Loading statements for LANDP for OS/2 servers" on page 144 or "Loading statements for LANDP for Windows NT servers" on page 182.

MQ server example

```
MQSeries Link server example

           LWSCONF NAME=AA,
                   TYPE=OS/2,
/* MQSeries Link server definition. Example. */
                   SERVER=(EHCMQ##),
                   PAR&MQ=(EHC_QM,32000,W,D:\EHC_QM\,64,4)
           LWSCONF NAME=BB,
                   TYPE=DOS,
/* LANDP link server client definition. Example, two sessions */
                   CLIENT=(EHCMQ01,AA),
                   CLIENT=(EHCMQ02,AA)
```

MSR/E server definitions

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT

When the magnetic stripe reader/encoder (MSR/E) server is loaded in a DOS, OS/2, or Windows NT workstation, it requires a PAR&MSRE keyword for the workstation that provides services.

On LANDP for AIX, the PAR&MSRE is defined in the post-customization phase.

Keyword **Description**

PAR&MSRE

Is a required keyword to define the MSR/E server parameters, when the server is specified in a SERVER keyword.

You can specify up to 2 parameters:

1. MSR/E device.

   This parameter specifies the product attached to the workstation that provides the MSR/E device to be supported by the server.

   The parameter is required. The parameter value can be:

   4717 IBM 4717 Magnetic Stripe Reader/Encoder (not supported on LANDP for Windows NT)
   4777 IBM 4777 Magnetic Stripe Reader/Encoder
   4778 IBM 4778 PIN Pad Magnetic Stripe Reader

2. COM port.

   This parameter specifies the COM port where the product that provides the MSR/E device will be attached.

   The parameter applies only if value 4777 or 4778 was specified in parameter 1.

On LANDP for DOS and OS/2, the parameter value can be in the range 1 to 4, or M. If the mouse port is to be used, specify M.

On LANDP for Windows NT, the parameter value can be in the range 1 to 8. The mouse port attachment is not supported.

The default is **1**.

If 4778 PIN pad capabilities are to be used, the value must match that specified in the PAR&PINP keyword.

MSR/E server example

```
         LWSCONF  NAME=AA,
                  TYPE=NT,
                  PRODLVL=L50,
/* Magnetic Stripe Reader/Encoder server definition. Example */
                  SERVER=(MSRE4701),
                  PAR&MSRE=(4777,1)
```

Native X.25 server definitions

- LANDP for DOS
- LANDP for OS/2

The native X.25 server requires one PAR&XNAT keyword for the workstation that provides services, and one SES&NSVC keyword for each switched virtual circuit used by the client workstation.

When the native X.25 services are provided by a DOS workstation, only the SES&NSVC keywords are required.

Keyword **Description**

PAR&XNAT

LANDP for OS/2: Is a required keyword to define the native X.25 server parameters, when this server is specified in a SERVER keyword.

You can specify up to 4 parameters:

1. Telephone number selection.

   The parameter is optional. The parameter value can be Y, to include your telephone number in the packet call, or N, not to include it. The default is **N**.

2. Your country identification.

   The parameter is optional. The parameter value is a number of three digits. Valid values for your country can be obtained from the telecommunications company. The default is **000**.

3. Your country subcode.

   The parameter is optional. The parameter value ranges from 0 to 9. Valid values for your country can be obtained from the telecommunications company. The default is **0**.

4. Your subscriber number.

   The parameter is required, if you have assigned the Y value to the first parameter (telephone number selection). If not, the parameter is optional.

   The parameter value is a number of up to 11 digits.

SES&NSVC

Is a required keyword to define a native X.25 server session, when this server is specified in a CLIENT keyword. One keyword is required for each native X.25 session defined, that will be used from this client workstation.

You can specify up to 9 parameters:

1. Name of the workstation providing X.25 Native services.

   The parameter is required. The parameter value is a string of up to 2 alphanumeric characters.

2. Session identification.

   The parameter is required. The parameter value ranges from 1 to 15, and must be unique for each client workstation. This includes all the circuits: native X.25, SNA, or user communication server.

3. Connection type.

   The parameter is optional. The parameter value can be:

   INCO Incoming
   OUTG Outgoing

   It must match the value assigned in the PAR&X25D keyword, or on the X.25 profile. The default is **OUTG**.

4. Destination identification.

   The parameter is required for incoming virtual circuits, and optional for outgoing circuits. The parameter value is a string of up to 8 alphanumeric characters.

5. Partner subscriber address.

   This parameter specifies the subscriber address of the partner you want to communicate with. For international calls, the subscriber number must be preceded by the country identification and the country subcode.

   The parameter is optional for incoming virtual circuits, and required for outgoing circuits. The parameter value is a string of up to 15 digits.

6. User data field. This parameter specifies information of your own that is sent across the communication line every time a call is made.

   The parameter is optional. The parameter value is a string of up to 4 bytes (eight hexadecimal characters).

Facility code

Parameters 7, 8, and 9 define the transmission facilities. The information is optional and made up of up to 63 bytes (126 hexadecimal characters). Refer to the X.25 Codification Facilities Rules from your X.25 Network provider.

7. Facility code (part 1).

   The parameter is optional. The parameter value is a string of up to 21 bytes (42 hexadecimal characters).

8. Facility code (part 2).

   To specify this code, facility code (part 1) must be completely filled.

   The parameter is optional. The parameter value is a string of up to 21 bytes (42 hexadecimal characters).

9. Facility code (part 3).

   To specify this code, facility code (part 1) and facility code (part 2) must be completely filled.

   The parameter is optional. The parameter value is a string of up to 21 bytes (42 hexadecimal characters).

Native X.25 server example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=(X25DLC2),
                  PAR&X252=(20,Y,7,3,PROFILE,214,1,231020107),
/* Native X.25 server definition (using the IBM X.25
   Interface Coprocessor/2). Example */
                  SERVER=(X25NAT##),
                  CLIENT=(X25NAT01,AA),
                  SES&NSVC=(AA,01,OUTG,INBB,203456988)

         LWSCONF  NAME=BB,
                  TYPE=DOS,
/* Native X.25 client definition. Example */
                  CLIENT=(X25NAT01,AA),
                  SES&NSVC=(AA,01,OUTG,INBB,203456988)
```

Object post box server definitions

- LANDP for OS/2

The object post box server requires one PAR&OPBS keyword.

Keyword Description

PAR&OPBS

Is a required keyword to define the object post box server parameters, when the server is specified in a SERVER keyword.

You can specify one parameter:

1. Database path.

   This parameter specifies the path where the object post box server databases are located.

   The parameter is required. The parameter value is a string of up to 30 alphanumeric characters. The format must be:

   D:\\[*directory1*\\[*directory2*\\[*directory3*\\]]]

   A maximum of three levels is permitted for the path.

Object post box server example

```
   LWSCONF  NAME=AA,
            TYPE=OS/2,
/* Object post box server definition.  Example */
            SERVER=(OPBS),
            PAR&OPBS=(C:\OPBSDB\)

   LWSCONF  NAME=BB,
            TYPE=DOS,
/* Object post box client definition.  Example */
            CLIENT=(OPBS,AA)
```

ODBC Query server definitions

- LANDP for Windows NT

The ODBC query server requires one PAR&ODB keyword.

Keyword Description

PAR&ODB

Is a required keyword to define the ODBC query server parameters, when the server is specified in a SERVER keyword.

You can specify up to 5 parameters:

1. Configuration name.

   This parameter specifies the default Data Source name as defined in the ODBC Driver Manager.

   The parameter is optional. The parameter value is a string of up to 20 alphanumeric characters. The default is **CONFIGUR**.

2. Number of initial worker threads.

The number of worker threads grows depending on the amount of throughput. This parameter represents the number of worker threads to be started at initialization.

The parameter is optional. The parameter value is in the range from 1 through 128, but must be less than or equal to parameter 3. The default is **5**.

3. Maximum number of worker threads.

   If system resources are scarce, this parameter puts a lower threshold on the number of worker threads EHCODB## can start after initialization.

   This parameter is optional. The parameter value is in the range from 1 through 128, but must be greater than or equal to parameter 2. The default is **128**.

4. Maximum sessions per workstation.

   This parameter specifies the maximum number of sessions that can be concurrently open on a workstation.

   This parameter is optional. The parameter value is in the range from 1 through 64. The default is **10**.

5. Request time-out.

   This parameter specifies the time, in seconds, that the ODBC query server waits for a reply from the data source before giving a time-out response.

   The parameter is optional. The parameter value ranges from 10 to 32767. The default is **15**.

ODBC query server example

```
   LWSCONF  NAME=AA,
            TYPE=NT,
/* ODBC query server definition.  Example */
            SERVER=(EHCODB01),
            PAR&ODB=(BASEMIX,4,14,10,30)

   LWSCONF  NAME=AB,
            TYPE=NT,
/* ODBC query server client definition.  Example */
            CLIENT=(EHCODB01,AA)
```

PIN pad server definitions

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- LANDP for AIX

When the PIN pad server is loaded in a DOS, OS/2, or Windows NT workstation, it requires one PAR&PINP keyword for the workstation that provides services.

Keyword **Description**

PAR&PINP

Is a required keyword to define the PIN pad server parameters, when the server is specified in a SERVER keyword.

You can specify up to 3 parameters:

1. PIN pad device.

   This parameter specifies the product attached to the workstation that provides the PIN pad device to be supported by the server.

   The parameter is required. The parameter value can be:

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

2. COM port.

   This parameter specifies the COM port where the product that provides the PIN pad device will be attached.

   The parameter applies only if value 4778 was specified in parameter 1. On LANDP for DOS, OS/2, and AIX, the parameter value can be in the range 1 to 4, or M. If the mouse port is to be used, specify M.

   On LANDP for Windows NT, the parameter value can be in the range 1 to 8. The mouse port attachment is not supported.

   The default is **1**.

   If 4778 magnetic stripe reader capabilities are to be used, the value must match that specified in the PAR&MSRE keyword.

3. Magnetic stripe reader selection.

   This parameter specifies whether 4778 magnetic stripe reader capabilities are to be used, or not.

   The parameter applies only if value 4778 was specified in parameter 1. It is optional. The parameter value can be Y, to use 4778 magnetic stripe reader capabilities, or N, not to use them. The default is **Y**.

PIN pad server example

```
         LWSCONF  NAME=BB,
                  TYPE=OS/2,
                  PRODLVL=L50,
/* PIN pad server definition. Example */
                  SERVER=(PINP4702),
                  PAR&PINP=(4778,2,Y)
```

Query server definitions

- LANDP for OS/2
- LANDP for AIX

The query server requires one PAR&SQL keyword.

Keyword **Description**

PAR&SQL Is a required keyword to define the query server parameters, when the server is specified in a SERVER keyword.

You can specify up to 11 parameters:

1. Configuration name.

   This parameter specifies the Database Manager database name.

   The parameter is optional. The parameter value is a string of up to 8 alphanumeric characters. The default is **CONFIGUR**.

2. Maximum concurrent requests.

   This parameter specifies the maximum number of requests the server can process at the same time.

   The parameter is optional. The parameter value ranges from 1 to 128. The default is **5**.

3. Maximum simultaneous commit units.

   This parameter specifies the maximum number of requests from different sessions the server processes at the same time.

   The parameter is optional. The parameter value ranges from 1 to 128. The default is **4**.

4. Maximum sessions per terminal.

   This parameter specifies the maximum number of applications running EHCSQL## sessions plus the number of extra sessions per workstation using the Open Session (OS) function.

   The parameter is optional. The parameter value ranges from 1 to 64. The default is **10**.

5. Shared file response time emulation selection.

   The parameter is optional. The parameter value can be Y, to get uniform response time, or N, not to specify it. The default is **Y**.

6. Request time-out.

   This parameter specifies the time, in seconds, the query server waits for a reply from SQL before giving a time-out response.

   The parameter is optional. The parameter value ranges from 10 to 32767. The default is **15**.

7. Reserved.

8. Reserved.
9. Reserved.
10. Reserved.
11. ODBC flag.

    This parameter, if set to Y, indicates that the ODBC query server EHCODB## is to be invoked rather than the query server EHCSQL##. Valid values are Y and N. The default is N.

    On a Windows NT workstation, only Y is valid. On OS/2 and AIX workstations, only N is valid.

Query server example

```
         LWSCONF  NAME=AA,
                  TYPE=OS/2,
/* Query server definition. Example. */
                  SERVER=(EHCSQL01),
                  PAR&SQL=(CONFIGUR,5,4,10,Y,15)

         LWSCONF  NAME=BB,
                  TYPE=DOS,
/* Query client definition. Example. */
                  CLIENT=(EHCSQL01,AA)
```

RCMS definitions

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT

The remote change management services (RCMS) require one PAR&RCMS keyword.

Keyword **Description**

PAR&RCMS
Is a required keyword to define RCMS, when RCMS is specified in a SERVER keyword.

You can specify up to 9 parameters:

1. Name of the workstation providing SNA services.

   The parameter is required. The parameter value is a string of up to 2 alphanumeric characters.

2. SNA LU pool ID or LU number.

   To use the SNA session with LU pooling support, specify the LU pool ID. The parameter value is a string of two characters: the first character must be alphabetical, and the second one alphanumeric.

   To use the session without LU pooling support, when the SNA services are provided from a DOS workstation, specify the LU number.

The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit.

Omit the parameter to use the session without LU pooling support when the SNA services are provided from an OS/2, Windows NT, or RS/6000-type workstation.

3. DLC type.

   The parameter applies only if the SNA services are provided by a DOS workstation. In this case, it is required except for workstations that have one DLC only, which is not X.25.

   For workstations with DLC other than X.25, to specify the type of DLC used, the parameter value can be:

   SDL Synchronous Data Link Control (SDLC)
   TKR Token-ring
   DCA Device Cluster Attachment (DCA)

   For workstations with DLC being X.25, the parameter value is the number you have assigned as the virtual circuit definition identification in a SES&SPVC keyword or SES&SSVC keyword.

4. Session device name

   This parameter specifies the name of the device on the AS/400 that is used for communication purposes.

   The parameter applies only when the SNA services are provided by an OS/400 system.

   The parameter is optional. The parameter value is a string of up to 10 alphanumeric characters. The default is **QCFBZZRCMS**.

5. System file path.

   This parameter specifies the directory where the logical names file (RCMS.LNF), the RCMS.ACK file, and the EBCDIC-to-ASCII and ASCII-to-EBCDIC translation tables are located in the production workstation at run-time.

   The parameter is required. The parameter value is a string of up to 30 alphanumeric characters. The format must be:

   ```
   D:\[directory1\[directory2\[directory3\]]]
   ```

   A maximum of three levels is permitted for the path.

6. Process file path.

   This parameter specifies the directory where RCMS is to store the events occurring during the process, as well as the message file, and log file.

   The parameter is required. The parameter value is a string of up to 30 alphanumeric characters. The format must be:

   ```
   D:\[directory1\[directory2\[directory3\]]]
   ```

A maximum of three levels is permitted for the path.

7. Reserved.

8. ASCII-to-EBCDIC translation table.

   If only one table is defined, the parameter is optional. The parameter value is a string of three alphanumeric characters, which is the identifier of the translation table. It must match the value assigned in the EXTEN keyword in a XLATETBL vector with TYPE=AERCMS.

   In DBCS mode the parameter specifies the translation mode. The parameter value can be:

   S Standard ASCII-EBCDIC and EBCDIC-ASCII translation

   P 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

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

9. EBCDIC-to-ASCII translation table.

   If only one table is defined, the parameter is optional. The parameter value is a string of three alphanumeric characters, which is the identifier of the translation table. It must match the value assigned in the EXTEN keyword in a XLATETBL vector with TYPE=EARCMS.

   In DBCS mode the parameter does not apply.

RCMS example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=(SNA##,,N),
                  PAR&SNA=(APPL,SRV),
                  SERVER=(TRDLC),
                  PAR&TKR=(48,04,04,00999999,00111111,017,00000),
/* Remote Change Management Services server definition. Example. */
                  SERVER=(RCMS),
                  PAR&RCMS=(AA,021,TKR,,C:\RCMSYS\,
                           C:\RCMPRO\,,OUT,INP)
```

Shared DOS directory services definitions

- LANDP for DOS

The shared DOS directory server requires one PAR&SHDR keyword. You can also specify one SES&SHDR keyword for each client workstation.

Keyword **Description**

PAR&SHDR

Is an optional keyword to define the shared DOS directory server parameters, when the server is specified in a SERVER keyword.

You can specify one parameter:

1. Number of entries for the shared DOS directory server.

   This parameter specifies the number of entries in the profile table that describes the relationship between the *short name* and the directories.

   The parameter is optional. The parameter value ranges from 1 to 99. The default is **10**.

SES&SHDR

Is an optional keyword to define the data area used by the shared DOS directory server, when the server is specified in a CLIENT keyword.

You can specify two parameters:

1. Name of the workstation providing shared DOS directory services.

   The parameter is required, if you specify the keyword. The parameter value is a string of up to 2 alphanumeric characters.

2. Request/reply data area size.

   This parameter specifies the size of the data area, in K.

   The parameter is required, if you specify the keyword. The parameter value ranges from 1 to 56. The default is **4**.

Shared DOS directory services example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
/* Shared DOS directory server definition. Example. */
                  SERVER=(SHRDIR),
                  PAR&SHDR=(10)

         LWSCONF  NAME=BB,
                  TYPE=DOS,
/* Shared DOS directory client definition. Example. */
                  CLIENT=(SHRDIR,AA),
                  SES&SHDR=(AA,10)
```

Shared-file distributor definitions

- LANDP for OS/2

The shared-file distributor requires one PAR&SFD keyword.

Keyword **Description**

PAR&SFD Is a required keyword to define the shared-file distributor parameters, when the server is specified in a SERVER keyword.

You can specify up to 2 parameters:

1. Creation of statistics file selection.

   The parameter is optional. The parameter value can be Y, to create a file to collect statistics, or N not to create the file. The default is **N**.

2. Number of threads.

   This parameter specifies the number of threads to attend and process requests in parallel.

   The parameter is optional. The parameter value ranges from 1 to 252. The default is **2**.

Shared-file distributor example

```
         LWSCONF  NAME=M1,
                  TYPE=OS/2,
                  PRODLVL=L50,
/* Shared-file distributor definition.  Example. */
                  SERVER=EHCSFD01,
                  PAR&SFD=(Y,2),
/* Shared-file server definition.  Example. */
                  SERVER=SHFILE01,
                  PAR&SHFL=(BASEDB11,5,10,Y,3,,EHCSFD01),
/* Shared-file distributor client definition. */
                  CLIENT=(EHCSFD01,M1),
/* Shared-file server client definitions. */
                  CLIENT=(SHFILE01,M1),
                  CLIENT=(SHFILE02,M2),
                  CLIENT=(SHFILE03,M3)

         LWSCONF  NAME=M2,
                  TYPE=OS/2,
                  PRODLVL=L50,
/* Shared-file server definition.  Example. */
                  SERVER=SHFILE02,
                  PAR&SHFL=(BASEDB12,5,10,Y,3,,EHCSFD01),
/* Shared-file distributor client definition. */
                  CLIENT=(EHCSFD01,M1),
/* Shared-file server client definitions. */
                  CLIENT=(SHFILE01,M1),
                  CLIENT=(SHFILE02,M2),
                  CLIENT=(SHFILE03,M3)

         LWSCONF  NAME=M3,
                  TYPE=OS/2,
                  PRODLVL=L50,
/* Shared-file server definition.  Example. */
                  SERVER=SHFILE03,
                  PAR&SHFL=(BASEDB13,5,10,Y,3,,EHCSFD01),
/* Shared-file distributor client definition. */
                  CLIENT=(EHCSFD01,M1),
/* Shared-file server client definitions. */
                  CLIENT=(SHFILE01,M1),
                  CLIENT=(SHFILE02,M2),
                  CLIENT=(SHFILE03,M3)
```

Shared-file replicator definitions

- LANDP for OS/2

The shared-file replicator requires one PAR&SFR keyword.

Keyword **Description**

PAR&SFR Is a required keyword to define the shared-file replicator parameters, when the server is specified in a SERVER keyword.

You can specify up to 4 parameters:

1. Creation of statistics file selection.

   The parameter is optional. The parameter value can be Y, to create a file to collect statistics, or N not to create the file. The default is **N**.

2. Number of threads.

   This parameter specifies the number of threads to attend and process requests in parallel.

   The parameter is optional. The parameter value ranges from 1 to 252. The default is **2**.

3. Shared-file replicator owner.

   This parameter specifies the full name of the owner of the shared-file replicator.

   The parameter value can be:

   EHCSFD## Shared-file distributor with identification ##, where the ## suffix has been substituted to completely identify the distributor.

   EHCSFR## Shared-file replicator with identification ##, where the ## suffix has been substituted to completely identify the replicator.

4. PCB list file.

   This parameter specifies the filename of a file with extension PCN, which will contain the list of PCBs to be accessed by the shared-file replicator.

   The parameter is optional. The parameter value is a string with up to 8 alphanumeric characters. The default is **EHCSFRPC**.

Shared-file replicator example

```
         LWSCONF  NAME=M1,
                  TYPE=OS/2,
                  PRODLVL=L50,
/* Shared-file replicator definition.  Example. */
                  SERVER=EHCSFR01,
                  PAR&SFR=(Y,2,,BASEDB11),
/* Shared-file server definition.  Example. */
                  SERVER=SHFILE01,
                  PAR&SHFL=(BASEDB11,5,10,Y,3,,EHCSFR01),
/* Shared-file replicator client definition. */
                  CLIENT=(EHCSFR01,M1),
/* Shared-file server client definitions. */
                  CLIENT=(SHFILE01,M1),
                  CLIENT=(SHFILE02,M2),
                  CLIENT=(SHFILE03,M3)

         LWSCONF  NAME=M2,
                  TYPE=OS/2,
                  PRODLVL=L50,
/* Shared-file server definition.  Example. */
                  SERVER=SHFILE02,
                  PAR&SHFL=(BASEDB11,5,10,Y,3,,EHCSFR01),
/* Shared-file replicator client definition. */
                  CLIENT=(EHCSFR01,M1),
/* Shared-file server client definitions. */
                  CLIENT=(SHFILE01,M1),
                  CLIENT=(SHFILE02,M2),
                  CLIENT=(SHFILE03,M3)

         LWSCONF  NAME=M3,
                  TYPE=OS/2,
                  PRODLVL=L50,
/* Shared-file server definition.  Example. */
                  SERVER=SHFILE03,
                  PAR&SHFL=(BASEDB11,5,10,Y,3,,EHCSFR01),
/* Shared-file replicator client definition. */
                  CLIENT=(EHCSFR01,M1),
/* Shared-file server client definitions. */
                  CLIENT=(SHFILE01,M1),
                  CLIENT=(SHFILE02,M2),
                  CLIENT=(SHFILE03,M3)
```

Shared-file server definitions

- LANDP for DOS
- LANDP for OS/2
- LANDP for AIX
- LANDP for Windows NT

The shared-file server requires one PAR&SHFL keyword.

Keyword **Description**

PAR&SHFL

Is a required keyword to define the shared-file server parameters, when the server is specified in a SERVER keyword.

You can specify up to 9 parameters:

1. Profile name.

   This parameter specifies the name of the shared-file server profile to be used.

   If only one shared-file server profile is specified in all the SHFLDBD vectors you define, the parameter is optional. The parameter value is a string of up to 8 alphanumeric characters, and must match the value assigned in the SHFLPRF keyword in a SHFLDBD vector.

   If shared-file replicator is to be used, all shared-file servers owned by the same shared-file replicator must use the same profile.

2. Number of additional buffers.

   Each additional buffer requires 1KB of memory.

   The parameter is optional. The parameter value ranges from 0 to 484. The default is **0**.

3. Number of additional sessions.

   The electronic journal, store-for-forwarding, and forwarding servers require an extra session. If those servers are loaded, the parameter is required.

   If the application obtains additional sessions with the Open Session (OS) function, or if a separate session is defined for the electronic journal or the store-for-forwarding server, you must add those sessions.

   The object post box server also requires an extra session.

   The parameter value ranges from 0 to 84. The value plus the number of client workstations must not be higher than 116. The default is **0**.

4. Creation of statistics file selection.

   The parameter is optional. The parameter value can be Y, to create a file to collect statistics, or N not to create the file. The default is **N**.

5. Log management type

   The parameter is optional. The parameter value can be:

   0 Dynamic and static log with a unique log file
   1 Dynamic log with a unique log file
   2 Dynamic and static log with two log files
   3 Dynamic log with two log files

The default is **0**.

6. Number of files open at a time.

   The parameter is optional. The parameter value ranges from 10 to 245.

7. Shared-file server owner.

   This parameter specifies the full name of the owner of the shared-file server.

   The parameter does not apply to DOS, Windows NT, or RS/6000 workstations, or if the shared-file server is used by:

   - Electronic journal server
   - Object post box server
   - Store-for-forwarding server

   The parameter is optional. The parameter value can be:

   EHCSFD## Shared-file distributor with identification ##, where the ## suffix has been substituted to completely identify the distributor.

   EHCSFR## Shared-file replicator with identification ##, where the ## suffix has been substituted to completely identify the replicator.

8. Workstation name on which a backup XLR server is to be run for this active Shared File server.

   If there are active and backup workstations, each must run either OS/2 or Windows NT. If one XLR server runs OS/2 and the other runs Windows NT, either can be the active server. Each XLR workstation must run the Service Availability Manager.

   The parameter is optional.

9. Delay in seconds before the backup takes over from a failed active XLR server. This parameter is meaningful only if parameter 8 is specified.

   The parameter is optional.

   Range 0-999. Default **0** (no automatic takeover).

Shared-file server example

```
         LWSCONF  NAME=AA,
                  TYPE=OS/2,
/* Shared file server definition. Example. */
                  SERVER=(SHFILEBA),
                  PAR&SHFL=(PROFSFS,10,3,Y,0,50)

         LWSCONF  NAME=BB,
                  TYPE=OS/2,
/* Shared file client definition. Example. */
                  CLIENT=(SHFILEBA,AA)
```

Shared-file server example with XLR: In the following example, the active server (SHFILE01) is configured on the workstation O1, with a backup on N1. A takeover by the backup occurs 10 seconds after detection of a failure by the active. EHCSAM must run on both these workstations. EHCSAM is also configured to run on workstation O2. The purpose of EHCSAM on O2 is to maintain state information, which is useful if either XLR workstation is not available during the startup of the workgroup EHCCUS.

```
LANCONF  GROUP=EHCCUS,      /* Workgroup Definition  */
         NAME=XLR,
         SUFFIX=Y,
         WSNAMES=(O1,O2,O3,N1)

LWSCONF  NAME=O1,           /* Workstation definition O1 */
         PRODLVL=L50,
         TYPE=OS/2,

         SERVER=EHCSAM,     /* Service Availability Manager */

         SERVER=(EHCVDMGR), /* Multiple virtual DOS machine relay definition */

         SERVER=(SHFILE01),  /* Shared-file server definition */
         PAR&SHFL=(SHFLPRF1,10,3,N,0,100,,N1,10),

         CLIENT=(SHFILE01,O1) /* Client definitions */

LWSCONF  NAME=O2,            /* Workstation definition O2 */
         PRODLVL=L50,
         TYPE=OS/2,

         SERVER=EHCSAM,      /* Service Availability Manager */

         SERVER=(EHCVDMGR),  /* Multiple virtual DOS machine relay definition */
         CLIENT=(SHFILE01,01)  /* Client definitions */
LWSCONF  NAME=03,            /* Workstation definition 03 */
         PRODLVL=L50,
         TYPE=OS/2,

         SERVER=(EHCVDMGR),  /* Multiple virtual DOS machine relay
                                                definition */

         CLIENT=(SHFILE01,01) /* Client definition */
LWSCONF  NAME=N1,             /* Workstation definition N1 */
         PRODLVL=L50,
         TYPE=NT,

         SERVER=EHCSAM,       /* Service Availability Manager */

         SERVER=(EHCVDMGR),   /* Multiple virtual DOS machine relay
                                                definition */

         CLIENT=(SHFILE01,01)
```

SNA server definitions

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- LANDP for AIX

The SNA server requires one PAR&SNA keyword for the workstation that provides services, and one SES&SNA keyword for each session used by the client workstation.

When the SNA services are provided from an RS/6000 workstation, neither the PAR&SNA nor SES&SNA keyword is required.

When the SNA services are provided from an OS/2 or Windows NT workstation that does not use LU pool support, only the PAR&SNA keyword is required.

When the SNA servers are provided from a DOS workstation and RLE compression is to be used, the EHCCOMP server must be defined in the same workstation.

When the SNA server is used with X.25 DLC, one SES&SPVC keyword for each permanent circuit, and one SES&SSVC keyword for each switched circuit, managed by the SNA server, are also required.

When the SNA server is used with X.25 DLC, and you plan to use the define connection (DC) or query connection (QC) functions, one SBSX25 keyword is required.

Keyword **Description**

PAR&SNA Is a required keyword to define the SNA server parameters, when the server is specified in a SERVER keyword.

You can specify up to 7 parameters (on LANDP for Windows NT, only parameters 1 and 2 can be used).

1. Session initialization.

   This parameter specifies who initializes the session.

   The parameter is optional. The parameter value can be:

   APPL The application initializes the session.
   ANY Either the application or the host computer initializes the session.

   The default is **APPL**.

2. BID command management.

   This parameter specifies who manages the BID command.

   The parameter is optional. The parameter value can be:

   SRV The SNA server manages the BID command.
   APPL The application manages the BID command.

   The default is **SRV**.

3. Correlation table size.

   The parameter applies only to OS/2 workstations. The parameter value ranges from 1 to 255.

4. Wrap selection.

   The parameter applies only to OS/2 workstations. The parameter value can be:

   W To use wrap
   N Not to use wrap

   The default is **N**.

5. LU pool table.

   The parameter applies only to DOS workstations. The parameter value must match the value specified in the NAME keyword of a LUPOOL vector.

6. Server-managed cryptography.

   The parameter applies only to OS/2 workstations. The parameter value can be:

   Y To have cryptography managed by the server.
   N If cryptography will not be managed by the server.

   The default is **N**.

Server-managed cryptography uses the communications provider exits, ACSRENCR.DLL and ACSRDECR.DLL, which are supplied with LANDP, and the TSS SECY server.

7. Master key for server-managed cryptography.

   The parameter applies only to OS/2 workstations, and is only valid if server-managed cryptography has been specified (if parameter 6 on page 463 = **Y**).

   If specified, this parameter must be a string of 1 to 16 alphanumeric characters. If it is not specified, the key defaults to `TMK`*ssww* where *ss* is the session ID and *ww* is the workstation ID.

SES&SNA Is a required keyword to define a SNA server session when the server is specified in a CLIENT keyword. One keyword is required for each SNA session defined.

The keyword is required when the SNA services are provided from an OS/2 or Windows NT workstation that uses LU pool support, or from an RS/6000 system.

You can specify up to 5 parameters:

1. Name of the workstation providing SNA services.

   The parameter is required. The parameter value is a string of up to 2 alphanumeric characters.

2. SNA session identifier.

   This number is used by a server or application to identify the SNA session.

   The parameter is required. The parameter value ranges from 01 to 30, and must be unique for the client workstation. It must contain two digits.

   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. If LU pooling support is required, provide an SES&SNA keyword for each session to be pooled. If CLIENT=(SNA,xx) has been specified in LWSCONF, the SNA session identifier in the SES&SNA keyword may be any two ASCII characters.

3. SNA LU pool ID or LU number.

   To use the SNA session with LU pooling support, specify the LU pool ID. The parameter value is a string of two characters: the first character must be alphabetical, and the second one alphanumeric.

   To use the session without LU pooling support, when the SNA services are provided from a DOS workstation, specify the LU number. The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit. This corresponds to the LOCADDR on the host VTAM definition.

Omit the parameter to use the session without LU pooling support when the SNA services are provided from an OS/2, Windows NT, or RS/6000-type workstation.

4. DLC type used by the workstation providing SNA services.

   The parameter applies only when the SNA services are provided from a DOS workstation, and the session does not use pooling support. In this case, it is required except for workstations that have one DLC only, which is not X.25.

   For workstations with DLC other than X.25, to specify the type of DLC used, the parameter value can be:

   SDL Synchronous Data Link Control (SDLC)
   TKR Token-ring
   DCA Device Cluster Attachment (DCA)

   For workstations with DLC being X.25, the parameter value must match the value assigned as the virtual circuit definition identification (parameter 1) in a SES&SPVC keyword or SES&SSVC keyword.

5. Session device name.

   This parameter specifies the name of the device on the AS/400 that is used for communication purposes.

   The parameter applies only when the SNA services are provided by an OS/400 system. The parameter value is a string of up to 10 alphanumeric characters.

SES&SPVC

LANDP for DOS: Is a required keyword to define the X25DLC and X25DLC2 server sessions, when the SNA server is specified in a SERVER keyword, and there are X.25 *permanent circuits* using SNA support.

You can specify up to 4 parameters:

1. Virtual circuit identification.

   This parameter identifies the virtual circuit, and is used for reference purposes when defining the SNA session that will use it.

   The parameter is required. The parameter value ranges from 1 to 999. The value must be unique for all the circuits, permanent and virtual, that are used through the SNA server.

2. Permanent virtual circuit number.

   The parameter is required. The parameter value ranges from 1 to the number of permanent virtual circuits defined either in the PAR&X25D keyword or in the X.25 profile. It must be unique for the SES&SPVC keywords.

3. Identification block.

The parameter is optional. The parameter value is a hexadecimal number of three digits. The value must match the information in the host computer.

4. Identification number.

   The parameter is optional. The parameter value is a hexadecimal number of five digits. The value must match the information in the host computer.

   The combination of the constant X'0200', the identification block, and the identification number results in the exchange identifier (XID), which is only used for switched connections. For permanent SNA circuits, the XID is optional.

SES&SSVC

LANDP for DOS: Is a required keyword to define the X25DLC and X25DLC2 servers sessions, when the SNA server is specified in a SERVER keyword, and there are X.25 *switched* circuits using SNA support.

The number of SES&SSVC keywords included can be greater than the number of switched virtual calls defined either in the PAR&X25D keyword or in the X.25 profile. However, at run-time the number of switched virtual calls working simultaneously is limited by the number of switched virtual circuits defined.

You can specify up to 10 parameters:

1. Virtual circuit identification.

   This parameter identifies the virtual circuit, and is used for reference purposes when defining the SNA session that will use it.

   The parameter is required. The parameter value ranges from 1 to 999. The value must be unique for all the circuits, permanent and virtual, that are used through the SNA server.

2. Partner subscriber address.

   This parameter specifies the subscriber address of the partner you want to communicate with. For international calls, the subscriber number must be preceded by the country identification and the country subcode.

   The parameter is required. The parameter value is a string of up to 15 digits.

3. Identification block.

   The parameter is required. The parameter value is a hexadecimal number of three digits. The value must match the information in the host computer.

4. Identification number.

The parameter is required. The parameter value is a hexadecimal number of five digits. The value must match the information in the host computer.

The combination of the constant X'0200', the identification block, and the identification number results in the exchange identifier (XID).

5. Type of connection.

   The type of connection selected must match those defined during the customization of the X.25 communications adapter card, and the values assigned either in the PAR&X25D keyword, or in the X.25 profile.

   To assign a circuit as incoming, you must have selected at least one incoming or one both-way call during configuration of the adapter.

   To assign a circuit as outgoing, you must have selected at least one outgoing or one both-way call during configuration of the adapter.

   The parameter is optional. The parameter value can be:

INCO	Incoming calls
BOTH	Both-way calls
OUTG	Outgoing calls

   The default is **OUTG**.

6. Physical unit (PU) identification.

   The parameter is optional. The parameter value is a string of 8 bytes (16 hexadecimal characters) and must match the host computer definitions.

7. User data field.

   This is information of your own that is sent across the communication line every time a call is made.

   The parameter is optional. The parameter value is a string of up to 4 bytes (eight hexadecimal characters).

Facility code

Parameters 8, 9, and 10 define the transmission facilities. The information is optional and made up by up to 63 bytes (126 hexadecimal characters). Refer to the X.25 Codification Facilities Rules from your X.25 Network provider.

8. Facility code (part 1).

   The parameter value is a string of up to 21 bytes (42 hexadecimal characters).

9. Facility code (part 2).

To specify this code, facility code (part 1) must be completely filled. The parameter value is a string of up to 21 bytes (42 hexadecimal characters).

10. Facility code (part 3).

    To specify this code, facility code (part 1) and facility code (part 2) must be completely filled. The parameter value is a string of up to 21 bytes (42 hexadecimal characters).

SBSX25 LANDP for OS/2: Is a required keyword to specify the name of the X.25 directory table, when the SNA server is used with X.25 DLC and you plan to use the define connection (DC) or query connection (QC) functions.

The parameter value must match the value assigned to the TBLNAME keyword of a X25DIR vector.

SNA server examples

```
Example 1
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=(TRDLC),
                  PAR&TKR=(48,04,04,00999999,00111111,017,00000),
/* SNA server definition. Example. */
                  SERVER=(SNA##,,N),
                  PAR&SNA=(APPL,SRV)
         LWSCONF  NAME=BB,
                  TYPE=OS/2,
/*            or TYPE=NT,        */
/* SNA client definition. Example. */
                  CLIENT=(SNA01,AA),
                  SES&SNA=(AA,01,018,TKR)
```

```
Example 2
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=(SNA##,,N),
                  PAR&SNA=(APPL,SRV),
                  SERVER=(X25DLC2),
                  PAR&X252=(20,Y,7,3,PROFILE,214,1,231020107),
/* SNA server definition (using the IBM X.25 Interface
   Coprocessor/2) for permanent circuits. Example */
                  SES&SPVC=(001,1)

         LWSCONF  NAME=BB,
                  TYPE=DOS,
/* SNA client definition (using the IBM X.25 Interface
   Coprocessor/2) for permanent circuits. Example */
                  CLIENT=(SNA02,AA),
                  SES&SNA=(AA,02,019,001)
Example 3
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=(SNA##,,N),
                  PAR&SNA=(APPL,SRV),
                  SERVER=(X25DLC2),
                  PAR&X252=(20,Y,7,3,PROFILE,214,1,231020107),
/* SNA server definition (using the IBM X.25 Interface
   Coprocessor/2) for virtual circuits. Example */
                  SES&SSVC=(002,231020107,017,80011,BOTH)
         LWSCONF  NAME=BB,
                  TYPE=DOS,
/* SNA client definition (using the IBM X.25 Interface
   Coprocessor/2) for virtual circuits. Example */
                  CLIENT=(SNA02,AA),
                  SES&SNA=(AA,02,019,002)
```

Store-for-forwarding server definitions

- LANDP for DOS
- LANDP for OS/2
- LANDP for AIX
- LANDP for Windows NT

The store-for-forwarding server requires one PAR&SFOR keyword.

Keyword **Description**

PAR&SFOR

Is a required keyword to define the store-for-forwarding server parameters, when the server is specified in a SERVER keyword.

You can specify up to 2 parameters:

1. Profile name.

   If only one store-for-forwarding server profile is defined, the parameter is optional. The parameter value is a string of up to 8 alphanumeric characters, and must match the value assigned in the NAME keyword in the SFORWPRF vector.

2. Maximum record size.

   The parameter is optional. The parameter value ranges from 1 to 4, meaning the size in KBs. If that keyword is not specified, the default is **1**.

Store-for-forwarding server example

```
         LWSCONF  NAME=AA,
                  TYPE=OS/2,
                  SERVER=SMGR,
                  PAR&SMGR=(,,,SMGRPRF,C:\SMGR\,C:\SMGR\,C:\SMGR\,C:\SMGR\),
                  SERVER=(SHFILEBA),
                  PAR&SHFL=(PROFSFS,10,3,Y),
                  CLIENT=(SHFILEBA,AA),
                  SERVER=(SFQUERY),
/* Store for forwarding server definition. Example. */
                  SERVER=(SFORFORW),
                  PAR&SFOR=(PROFSTOR,1)

         LWSCONF  NAME=BB,
                  TYPE=DOS,
/* Store for forwarding client definition. Example. */
                  CLIENT=(SFORFORW,AA)
```

Synchronous data link control definitions

- LANDP for DOS

The SDLC server requires one PAR&SDLC keyword.

Keyword **Description**

PAR&SDLC

Is a required keyword to define the SDLC server parameters, when the server is specified in a SERVER keyword.

You can specify up to 25 parameters:

1. Number of buffers.

   The parameter is optional. The parameter value ranges from 8 to 999 (1 buffer = 290 bytes in memory, 256 bytes of data). The default is **8**.

2. Physical address in the host computer of the physical unit (PU).

   The parameter is required. The parameter value is a hexadecimal number of two digits.

3. NRZI (non return to zero inverted) selection.

   The parameter is optional. The parameter value can be Y, to use NRZI, or N, not to use NRZI. The default is **N**.

4. Switched line selection.

   The parameter is optional. The parameter value can be Y, when using a switched line, or N, when using a point-to-point line. The default is **Y**.

5. Connection under application selection.

   The parameter is optional. The parameter value can be:

   Y The application starts communication, by sending a Connect (CN) function call to the SNA server.

   N The SDLC server starts communication automatically.

   The default is **N**.

6. Identification block.

   When working with point-to-point lines, the parameter is optional. The parameter value is a hexadecimal number of three digits.

7. Identification number.

   When working with point-to-point lines, the parameter is optional. The parameter value is a hexadecimal number of five digits.

   The value must match the information in the host computer. The combination of the constant X'0200', the identification block, and the identification number results in the exchange identifier (XID), which is only used for switched connections.

8. Line mode.

   The parameter is optional. The parameter value can be TURN, for turnaround required, or CRTS, for constant request to send. When using a two-wire line, you must select TURN; when using a four-wire line, you can select any of the values. The default is **TURN**.

9. Line time-out.

   This parameter specifies the time-out in seconds.

   The parameter value ranges from 10 to 65. The default is **65**.

10. System Service Control Point (SSCP) name.

    The parameter is optional. The parameter value is a hexadecimal number of ten digits.

    If specified, the SNA server validates the SSCP name that receives in the ACTPU (activate physical unit) command.

11-25. System Service Control Point (SSCP) name.
See information for parameter 10.

Synchronous data link control example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=(SNA##,,N),
                  PAR&SNA=(APPL,SRV),
                  CLIENT=(SNA01,AA),
                  SES&SNA=(AA,01,006,SDL),
/* SDLC server definition. Example. */
                  SERVER=(SDLC),
                  PAR&SDLC=(58,AA,N,Y,Y,017,00067,TURN)
```

System manager operator definitions

- LANDP for DOS
- LANDP for OS/2

The system manager operator requires one PAR&SMOP keyword.

Keyword **Description**

PAR&SMOP

Is a required keyword to define the system manager operator parameters, when the system manager operator is specified in a SERVER keyword.

You can specify up to 3 parameters:

1. Audio signal selection.

   The audio signal notifies the operator of a pending message.

   The parameter is optional. The parameter value can be Y, to use the audio signal, or N, not to use it. The default is **N**.

2. Video signal selection.

   The video signal notifies the operator of a pending message.

The parameter is optional. The parameter value can be Y, to use the video signal, or N, not to use it. The default is **N**.

3. Path.

   This parameter specifies the directory where the system manager operator files are located in the production workstation at run-time.

   The parameter is required and applies only to DOS workstations. The parameter value is a string of up to 30 alphanumeric characters. The format must be:

   ```
   D:\[directory1\[directory2\[directory3\]]]
   ```

   A maximum of three levels is permitted for the path.

System manager operator example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=SMGR,
                  PAR&SMGR=(,,,SMGRPRF,C:\SMGR\,C:\SMGR\,C:\SMGR\,C:\SMGR\),
/* System manager operator server definition. Example. */
                  SERVER=(SMOP),
                  PAR&SMOP=(Y,Y,C:\SMOP\FILES\)
```

System manager server definitions

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT

The system manager server requires one PAR&SMGR keyword.

Keyword Description

PAR&SMGR

Is a required keyword to define the system manager server parameters, when the server is specified in a SERVER keyword.

You can specify up to 10 parameters:

1. Name of the workstation providing SNA services for alerts.

   The parameter applies only when using alerts. In this case, it is required. The parameter value is a string of up to 2 alphanumeric characters.

2. DLC type used by the workstation providing SNA services for alerts.

   The parameter applies only to DOS workstations. In this case, it is required except for workstations that have one DLC only, which is not X.25.

   For workstations with DLC other than X.25, to specify the type of DLC used, the parameter value can be:

SDL Synchronous Data Link Control (SDLC)
TKR Token-ring
DCA Device Cluster Attachment (DCA)

For workstations with DLC being X.25, the parameter value is the number you have assigned as the virtual circuit definition identification in a SES&SPVC keyword or SES&SSVC keyword.

3. Path for alerts.

   This parameter specifies the directory where the file to temporary store alerts is located in the production workstation at run-time. The file is automatically created by the system manager server at run-time, in the path you specify.

   The parameter value is a string of up to 30 alphanumeric characters. The format must be:

   ```
   D:\[directory1\[directory2\[directory3\]]]
   ```

   A maximum of three levels is permitted for the path.

4. Profile name.

   If only one system manager server profile is defined, the parameter is optional. The parameter value is a string of up to 8 alphanumeric characters, and must match the value assigned in the NAME keyword in the SMGRPRF vector.

5. Path for user profiles.

   This parameter specifies the directory where the user profiles are located in the production workstation at run-time.

   If you have defined user profiles, the parameter is required. The parameter value is a string of up to 30 alphanumeric characters. The format must be:

   ```
   D:\[directory1\[directory2\[directory3\]]]
   ```

   A maximum of three levels is permitted for the path.

6. Path for common data.

   This parameter specifies the directory where the common data is located in the production workstation at run-time.

   The parameter is required. The parameter value is a string of up to 30 alphanumeric characters. The format must be:

   ```
   D:\[directory1\[directory2\[directory3\]]]
   ```

   A maximum of three levels is permitted for the path.

7. Path for the log file.

   This parameter specifies the directory where the log file is located in the production workstation at run-time.

If log support is used, the parameter is required. The parameter value is a string of up to 30 alphanumeric characters. The format must be:

```
D:\[directory1\[directory2\[directory3\]]]
```

A maximum of three levels is permitted for the path.

8. Path for the record definition file.

   This parameter specifies the directory where the file containing the record definitions is located in the production workstation at run-time.

   The parameter is required if you have specified COMDTVAL = Y or APPDTVAL = Y in the SMGRPRF vector, or if the system manager server will perform record validation.

   The parameter value is a string of up to 30 alphanumeric characters. The format must be:

   ```
   D:\[directory1\[directory2\[directory3\]]]
   ```

   A maximum of three levels is permitted for the path.

9. Drive for the FBSS#GDT backup.

   The parameter specifies the drive where the FBSS#GDT backup is located in the production workstation at run-time.

   The parameter is optional. The parameter value is an alphabetical character. The customization program provides no default for this parameter.

10. Netview operator ID.

    This parameter specifies the Netview operator ID that receives messages at the host computer.

    The parameter is optional. The parameter value is a string of up to 8 ASCII characters. The customization program provides no default for this parameter.

System manager server example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=(SNA##,,N),
                  PAR&SNA=(APPL,SRV),
                  SERVER=(SDLC),
                  PAR&SDLC=(58,AA,N,Y,Y,017,00067,TURN),
/* System manager server definition. Example. */
                  SERVER=(SMGR),
                  PAR&SMGR=(AA,SDL,C:\SMGR\ALERTS\,PROFSMGR,
                   C:\SMGR\USERS\,C:\SMGR\COMMON\,
                   C:\SMGR\LOG\,C:\SMGR\RECORDS\)
```

TCP/IP wide area communications server definitions

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT

The TCP/IP wide area communications server requires one PAR&TCP keyword for the workstation that provides services and one SES&TCP keyword for each line of the EHCTCP.INI file. This file is used to map LANDP sessions with SNA and PPC servers to TCP/IP protocols, ports and internet addresses.

Keyword **Description**

PAR&TCP Is a required keyword to define the TCP/IP wide area communications server parameters when the server is specified in a SERVER keyword.

You can specify up to 3 parameters.

1. Maximum number of sessions.

   The parameter applies only to DOS workstations. The parameter value can range from 5 through 2048. The default is **5**. The number of sessions required is one for each user session plus two for each Dependent LU Server (DLUS).

2. SNA server emulation.

   The parameter value can be:

   Y To emulate the SNA server with SNA over TCP/IP or tn3270.

   N To not emulate the SNA server.

   The default is **N**.

   Note: When emulating the SNA server a SERVER statement for the SNA server must be included for this workstation. In the case of DOS workstations, although a SERVER statement for the SNA server is required, further SERVER statements for DLC servers are not required because the DLC function is provided by TCP/IP. Also, the LU number and DLC type parameters (on the various SES& and PAR& keywords relating to SNA sessions) are not required.

3. PPC server emulation.

   The parameter value can be:

   Y To emulate the PPC server and provide PPC over TCP/IP.

   N To not emulate the PPC server.

   The default is **N**.

   Note: When emulating the PPC server a SERVER statement for the PPC server must be included for this workstation.

Note: Parameters 2 and 3 can not both be Y on DOS workstations.

Keyword **Description**

SES&TCP Is an optional keyword that may appear more than once when the EHCTCP server is specified in a server keyword. Each occurrence defines a line of the EHCTCP.INI file.

You can specify 3 or 4 parameters:

1. The name of the session or other resource to be defined.

   This parameter is required. It is a string of up to 8 characters. One or more characters can be replaced with # to indicate that this definition applies to names with any character in that position. The last character can be * to indicate that this definition applies to names with any suffix in place of the *. See examples below for use.

2. Type of definition.

   This parameter is required. See examples below for use.

3. Value for definition.

   This parameter is required. See examples below for use.

4. Comment.

   An optional comment of up to 30 characters enclosed in apostrophes.

The purpose of the EHCTCP.INI file is to provide the information required to map LANDP sessions with SNA and PPC servers to TCP/IP protocols, ports and internet addresses. The TELNET tn3270 and MPTN(AnyNet) protocols are supported and these correspond to TCP/IP ports 23 and 397. IP addresses can be in dotted notation or names to be resolved by the domain name server or HOSTS file.

SNA server sessions are identified by SNA followed by the workstation ID followed by the session ID. For example: SNAGW12 is session 12 on workstation GW.

PPC server sessions are identified by the local and partner LU aliases.

Starting from the beginning of the file, the first match to the session identifier is used. # represents any character and * any suffix. For example, SNAGW12 will match SNA##12 and SNAGW*, whichever is first.

Example TCP/IP wide area communications server definitions

Some of the following examples have been formatted for ease of readability. In SES&TCP statements, all blanks separating parameters must be removed. The last three examples show valid formatting.

Example: EMU3270 server sessions:

This example shows how to map EMU3270 server sessions (SNA sessions 51 through 55) from all workstations in the workgroup to the TELNET tn3270 protocol on a host called winmvs20.

```
SES&TCP(SNA##5#, Port23, winmvs20, 'All EMU3270 -> winmvs20 TELNET'),
```

Example: PPC server session to MPTN protocol (AnyNet):

This example shows the definitions required to map a PPC server session to MPTN protocol (AnyNet). The local LU alias is IYAET000, the partner LU alias is IYCKSPP1 and the mode is CICSISC0. The first line maps the partner LU alias to the port and IP address to be used. The IP address is in dotted notation just to demonstrate the range of possibilities. The second line maps the local LU alias to the fully qualified SNA network name of the LU. The third line maps the partner LU alias to the fully qualified SNA network name of the LU. The last line defines the LU6.2 conversation mode. The value parameter must be a pair of numbers separated by a period. The first is the total number of sessions (range 1 through 32767) and the second is the number of contention winner sessions (range 0 through number of sessions).

```
SES&TCP(IYCKSPP1, Port397,  9.20.101.41,       'TCP/IP port and IP address'),
SES&TCP(IYAET000, LocalLU,  GBIBMIYA.IYAET000, 'Local LU fully qualified name'),
SES&TCP(IYCKSPP1, RemoteLU, GBIBMIYA.IYCKSPP1, 'Partner LU fully qual. name'),
SES&TCP(CICSISC0, MODE,     10.5,              'Mode: 10 sessions, 5 winners'),
```

Example: SNA server session to MPTN protocol (AnyNet):

The third example shows the definitions required to map SNA server sessions to MPTN protocol (AnyNet). This requires a dependent LU server (DLUS) on the host computer for which we have used a local alias of DLUS1 (DLUS is a required prefix for this name). The first line maps all remaining SNA sessions (that is, not the EMU3270 sessions mapped earlier) to a pool of 8 LU numbers (2 through 9) on a dependent LU server with local alias DLUS1. DLUS is a required prefix for this name. The value parameter can be a single LU number in the range 2 through 255, or a pair of numbers that define an LU pool. In the latter case the first number defines the lowest LU number in the pool and the second the size of the pool. This line could be replaced by many lines each mapping only one session to an LU number, or some combination of individual and pooled LU numbers. The second and third lines define the local and host fully qualified SNA network names to be used for communication with the DLUS. The fourth line gives the port/protocol and IP address to be used. The fifth and sixth lines define the XID and PU name required for communication with the DLUS.

```
SES&TCP(SNA*,  DLUS1,    2.8,              'Map to pool of LU numbers 2-9'),
SES&TCP(DLUS1, LocalLU,  GBIBMIYA.IYCUT310, 'Local node fully qual. name'),
SES&TCP(DLUS1, RemoteLU, GBIBMIYA.IYCUCDRM, 'Host node fully qualified name'),
SES&TCP(DLUS1, Port397,  winmvs2a,         'TCP/IP port and IP address'),
SES&TCP(DLUS1, XID,      05DE3001,         'XID'),
SES&TCP(DLUS1, PUname,   IYCUR301,         'PU name'),
```

Example: Connection of incoming TCP/IP sessions:

This example shows the definition required to allow connection of incoming TCP/IP sessions. This is required for PPC server 'contention-looser' sessions and for SNA dependent LU sessions (example above). The value is the number of connections that can be in the process of connecting at any one time (not the number of connections). A value of at least 2 is required for MPTN sessions. Increase it if you find that incoming (contention-looser) PPC sessions fail to connect.

```
SES&TCP(Listen, Port397, 2, 'Listen for 2 MPTN sessions'),
```

Note: To aid clarity mixed case is allowed in the EHCTCP.INI file. However the file is treated as upper case in use as all network names must be upper case.

Example: PPC server over TCP/IP (MPTN)

```
LWSCONF NAME=AB,
        TYPE=OS/2,
        PRODLVL=L50,
/* PPC server over TCP/IP */
        SERVER=(PPC,,N),
/* TCP/IP wide area communications server */
        SERVER=(EHCTCP,,N),
        PAR&TCP=(,,Y),
        SES&TCP=(Listen,Port397,2,'Listen for MPTN sessions'),
        SES&TCP=(IYAET000,LocalLU,GBIBMIYA.IYAET000,'Local LU alias'),
        SES&TCP=(IYCKSPP1,RemoteLU,GBIBMIYA.IYCKSPP1,'Partner LU alias'),
        SES&TCP=(IYCKSPP1,PORT397,winmvs20.ibm.com,'Port and IP address'),
        SES&TCP=(CICSISC0,MODE,2.1,'2 sessions, 1 winner'),
/* PPC client */
        CLIENT=(PPC,AB)
```

Example: Emulator sessions to SNA over TCP/IP (TELNET)

```
LWSCONF NAME=AA,
        TYPE=DOS,
        PRODLVL=L50,
/* SNA server over TCP/IP */
        SERVER=(SNA##,,N),
        PAR&SNA=(APPL,SRV),
/* TCP/IP wide area communications server */
        SERVER=(EHCTCP,,N),
        PAR&TCP=(5,Y,),
        SES&TCP=(SNA##5#,PORT23,winmvs20,'Emulator session TELNET'),
/* Emulator */
        SERVER=(EMU32701),
        PAR&3270=(N,N),
        SES&3270=(AA,1,,,ATR,DIS,KBD)
```

Example: SNA server over TCP/IP (MPTN)

```
LWSCONF NAME=AC,
        TYPE=OS/2,
        PRODLVL=L50,
/* SNA server over TCP/IP */
        SERVER=(SNA##,,N),
        PAR&SNA=(APPL,SRV),
/* TCP/IP wide area communications server */
        SERVER=(EHCTCP,,N),
        PAR&TCP=(,Y,),
        SES&TCP=(Listen,Port397,1,'Listen for MPTN sessions'),
        SES&TCP=(SNA####,DLUS1,2.20,'Map all sessions to pool of 20'),
        SES&TCP=(DLUS1,LocalLU,GBIBMIYA.IYCUT310,'Local LU alias'),
        SES&TCP=(DLUS1,RemoteLU,GBIBMIYA.IYCUCDRM,'Partner LU alias'),
        SES&TCP=(DLUS1,PORT397,winmvs20.ibm.com,'Port and IP address'),
        SES&TCP=(DLUS1,XID,05DE3001,'XID'),
        SES&TCP=(DLUS1,PUname,IYCUR301'PU name'),
/* SNA client */
        CLIENT=(SNA01,AC),
        SES&SNA=(AC,01,,,)
```

Token-ring data link control definitions

- LANDP for DOS

The TRDLC server requires one PAR&TKR keyword.

Keyword **Description**

PAR&TKR Is a required keyword to define the TRDLC server parameters, when the server is specified in a SERVER keyword.

You can specify up to 23 parameters:

1. Number of buffers.

   The parameter is optional. The parameter value ranges from 16 to 216 (1 buffer = 272 bytes in memory, 256 bytes of data). The default is **48**.

2. Service access point (SAP) for the workstation.

   The parameter is optional. The parameter value is a hexadecimal number of two digits:

   The first one ranges from 0 to E.
   The second one can be 0, 4, 8, or C.

   The value 00 is not valid. The default is **04**, which is the default SAP used by SNA nodes and identifies the path control as the data link user.

3. Service access point (SAP) for the host.

   The parameter is optional. The parameter value is a hexadecimal number of two digits:

   The first one ranges from 0 to E.
   The second one can be 0, 4, 8, or C.

   The value 00 is not valid. The default is **04**, which is the default SAP used by SNA nodes and identifies the path control as the data link user.

4. Workstation address.

   This parameter specifies the locally administered address for the workstation.

   The parameter is required. The parameter value is a hexadecimal number of eight digits. It ranges from 00000000 to 7FFFFFFF.

   The address becomes a hexadecimal number of twelve digits:

   The first four are 4000.
   The other eight are those provided as the parameter value.

   Note that the workstation address must *not* be the same as the host address.

5. Host address.

This parameter specifies the locally administered address for the host.

The parameter is required. The parameter value is a hexadecimal number of eight digits. It ranges from 00000000 to 7FFFFFFF. The address becomes a hexadecimal number with twelve digits:

The first four are 4000.
The other eight are those provided as the parameter value.

Note that the specified address must *not* be the same as the address of the workstation.

6. SNA XID block number.

   The parameter is optional. The parameter value is a hexadecimal number of three digits, and must match the information in the host computer. The default is **017**.

7. SNA XID identification number.

   The parameter is optional. The parameter value is a hexadecimal number of five digits, and must match the information in the host computer. The default is **00000**.

   The combination of the constant `X'0200'`, the block number, and the identification number results in the SNA exchange identifier (XID).

8. System Service Control Point (SSCP) name.

   The parameter is optional. The parameter value is a hexadecimal number of ten digits.

   If specified, the SNA server validates the SSCP name that receives in the ACTPU (activate physical unit) command.

9 to 23. System Service Control Point (SSCP) name.
See information for parameter 8.

Token-ring data link control example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=(SNA##,,N),
                  PAR&SNA=(APPL,SRV),
/* Token-Ring server definition. Example. */
                  SERVER=(TRDLC),
                  PAR&TKR=(48,04,04,00999999,00111111,017,00000),
                  CLIENT=(SNA01,AA),
                  SES&SNA=(AA,01,001,TKR)
```

Virtual file support definitions

- PC/Integrator

The virtual file support requires one PAR&VFIL keyword.

Keyword **Description**

PAR&VFIL Is a required keyword to define the virtual file support parameters, when this support is specified in a SERVER keyword.

You can specify up to 4 parameters:

1. Name of the workstation providing SNA services.

   The parameter is required. The parameter value is a string of up to 2 alphanumeric characters.

2. SNA LU pool ID or LU number.

   To use the SNA session with LU pooling support, specify the LU pool ID. The parameter value is a string of two characters: the first character must be alphabetical, and the second one alphanumeric.

   To use the session without LU pooling support, when the SNA services are provided from a DOS workstation, specify the LU number. The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit.

   Omit the parameter to use the session without LU pooling support when the SNA services are provided from an OS/2, Windows NT, or RS/6000-type workstation.

3. DLC type used by the workstation providing SNA services.

   The parameter applies only to DOS workstations. In this case, it is required except for workstations that have one DLC only.

   The parameter value can be:

SDL	Synchronous Data Link Control (SDLC)
TKR	Token-ring
DCA	Device Cluster Attachment (DCA)

4. Virtual drive.

   The parameter is required. This parameter specifies the virtual drive assigned to the virtual file support in the workstation.

   The parameter value ranges from C to Z.

Virtual file support example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=(SNA##,,N),
                  PAR&SNA=(APPL,SRV),
                  SERVER=(TRDLC),
                  PAR&TKR=(48,04,04,00999999,00111111,017,00000),
/* Virtual File Support server definition. Example. */
                  SERVER=(VFILE),
                  PAR&VFIL=(AA,016,TKR,V)

         LWSCONF  NAME=BB,
                  TYPE=DOS,
/* Virtual File Support server definition. Example. */
                  SERVER=(VFILE),
                  PAR&VFIL=(AA,017,TKR,V)
```

Virtual volume support definitions

- PC/Integrator
- PC Integrator/2

The virtual volume support requires one PAR&VVOL keyword.

Keyword Description

PAR&VVOL

Is a required keyword to define the virtual volume support parameters, when this support is specified in a SERVER keyword.

You can specify up to 27 parameters:

1. Name of the workstation providing SNA services.

   The parameter is required, unless the workstation is directly attached to the 4700 system via DCADLC, and is the same workstation where the virtual volume support installed. The parameter value is a string of up to 2 alphanumeric characters.

2. SNA LU pool ID or LU number.

   To use the SNA session with LU pooling support, specify the LU pool ID. The parameter value is a string of two characters: the first character must be alphabetical, and the second one alphanumeric.

   To use the session without LU pooling support, when the SNA services are provided from a DOS workstation, specify the LU number. The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit.

   Omit the parameter to use the session without LU pooling support when the SNA services are provided from an OS/2, Windows NT, or RS/6000-type workstation.

3. DLC type used by the workstation providing SNA services.

The parameter applies only to DOS workstations. In this case, it is required except for workstations that have one DLC only.

The parameter value can be:

SDL	Synchronous Data Link Control (SDLC)
TKR	Token-ring
DCA	Device Cluster Attachment (DCA)

4. Public virtual volume 1 selection.

   The parameter is optional. The parameter value can be Y, to select PUBLIC1 virtual volume, or N, not to select it. The default is **N**.

5 to 27. Public virtual volume *n* selection (n = 2, 3, ... , 24).

The parameter is optional. The parameter value can be Y, to select PUBLIC*n* virtual volume, or N, not to select it. The default is **N**.

At least one public virtual volume (n = 1, 2, ... ,24) must be selected.

Virtual volume support example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=(SNA##,,N),
                  PAR&SNA=(APPL,SRV),
                  SERVER=(TRDLC),
                  PAR&TKR=(48,04,04,00999999,00111111,017,00000),
/* Virtual Volume Support server definition. Example. */
                  SERVER=(RDVVOLS),
                  PAR&VVOL=(AA,013,TKR,Y,Y,Y,N)

         LWSCONF  NAME=BB,
                  TYPE=DOS,
/* Virtual Volume Support server definition. Example. */
                  SERVER=(RDVVOLS),
                  PAR&VVOL=(AA,014,TKR,Y,Y,Y,N)
```

X.25 data link control definitions

- LANDP for DOS

The X25DLC server, which supports the IBM PC X.25 Communications Adapter, requires one PAR&X25D keyword.

The X25DLC2 server, which supports the IBM X.25 Interface Coprocessor/2, requires one PAR&X252 keyword. To use the IBM X.25 Interface Coprocessor/2, it is required to run the customization program provided with the IBM X.25 Interface Coprocessor/2 support program. For information, refer to the *User's Guide* and the *Programmer's Reference* manuals of that product.

To use the X.25 support through the native X.25 server, it is necessary to specify one SES&NSVC keyword for each switched virtual circuit used by the client workstation. See "Native X.25 server definitions" on page 444.

To use the X25DLC support through the SNA server, it is necessary to specify one SES&SPVC keyword for each permanent circuit, and one SES&SSVC keyword for each switched circuit, managed by the SNA server. See “SNA server definitions” on page 462.

Keyword **Description**

PAR&X25D

Is a required keyword to define the X25DLC server parameters, when the server is specified in a SERVER keyword.

You can specify up to 16 parameters:

1. Number of buffers.

   The parameter is optional. The parameter value ranges from 20 to 2000 (1 buffer = 144 bytes in memory, 128 bytes of data). The default is **20**.

2. Telephone number selection.

   The parameter is optional. The parameter value can be Y, to include your telephone number in the packet call, or N, not to include it. The default is **Y**.

Number of virtual circuits and calls

Parameters 3, 4, 5, and 6 specify, respectively, the number of:

- Permanent virtual circuits
- Incoming virtual calls
- Both-way virtual calls
- Outgoing virtual calls

The sum of the permanent virtual circuits, incoming virtual calls, both-way virtual calls, and outgoing virtual calls, must be between 1 and 20. Note that you cannot assign 0 to the four parameters, but at least one of them must be assigned a non-zero value.

3. Number of permanent virtual circuits.

   The parameter is optional. The parameter value ranges from 0 to 20.

4. Number of incoming virtual calls.

   The parameter is optional. The parameter value ranges from 0 to 20.

5. Number of both-way virtual calls.

   The parameter is optional. The parameter value ranges from 0 to 20.

6. Number of outgoing virtual calls.

   The parameter is optional. The parameter value ranges from 0 to 20.

Logical channel identification

Parameters 7, 8, and 9 specify, respectively, the logical channel identification for:

- First incoming virtual call
- First both-way virtual call
- First outgoing virtual call

They are assigned in ascending order with the lowest being permanent virtual circuits:

- Permanent virtual circuits
- First incoming virtual call
- First both-way virtual call
- First outgoing virtual call

The circuit IDs are assigned consecutively, starting with the logical channel ID specified for the first circuit of each type. Therefore, if you had three permanent virtual circuits, the logical channel ID of the first incoming virtual call must be at least 4. Logical channel IDs cannot overlap. The maximum number of circuits is 20.

Note: Several circuit validations are performed on the data you provide:

- The number of permanent virtual circuits must be less than the logical channel ID of any incoming, both-way, or outgoing virtual calls.
- The maximum logical channel ID of incoming, both-way, and outgoing virtual calls cannot exceed 4095.
- The logical channel ID of the first incoming, both-way, or outgoing virtual calls must be left blank when the respective number of incoming, both-way, and outgoing virtual circuits is zero.

7. Logical channel identification for the first incoming virtual call.

   If the number of incoming virtual calls is different from zero, the parameter is required. The parameter value ranges from 1 to 4095.

8. Logical channel identification for the first both-way virtual call.

   If the number of both-way virtual calls is different from zero, the parameter is required. The parameter value ranges from 1 to 4095.

9. Logical channel identification for the first outgoing virtual call.

   If the number of outgoing virtual calls is different from zero, the parameter is required. The parameter value ranges from 1 to 4095.

10. Level 2 window size.

This parameter specifies the maximum number of messages that the workstation can send to the network before receiving an answer at frame level.

The parameter is optional. The parameter value ranges from 1 to 7. The default is **7**.

11. Level 2 retransmission limit.

    This parameter specifies the number of frame retransmission attempts after an error occurs.

    The parameter is optional. The parameter value ranges from 1 to 20. The default is **3**.

12. Level 2 retransmission time-out.

    This parameter specifies the amount of time, in hundredths of seconds, the system waits for an answer before retrying the transmission.

    The parameter is optional. The parameter value ranges from 100 to 500. The default is **150**.

13. Interrupt number.

    This parameter specifies the interrupt number selected on the X.25 communications adapter card when it was installed in the workstation.

    The parameter is optional. The parameter value ranges from 2 to 4. The default is **3**.

14. Your country identification.

    The parameter is required. The parameter value is a number of three digits. Valid values for your country can be obtained from the telecommunications company.

15. Your country subcode.

    The parameter is required. The parameter value ranges from 0 to 9. Valid values for your country can be obtained from the telecommunications company.

16. Your subscriber number.

    The parameter is required. The parameter value is a number of up to 11 digits.

PAR&X252

Is a required keyword to define the X25DLC2 server parameters, when the server is specified in a SERVER keyword.

You can specify up to 8 parameters:

1. Number of buffers.

The parameter is optional. The parameter value ranges from 20 to 2000 (1 buffer = 144 bytes in memory, 128 bytes of data). The default is **20**.

2. Telephone number selection.

   The parameter is optional. The parameter value can be Y, to include your telephone number in the packet call, or N, not to include it. The default is **Y**.

3. Memory pool size.

   This parameter specifies the amount of memory, in KB, reserved for the IBM X.25 Co-Processor Support program.

   The parameter is optional. The parameter value ranges from 7 to 99.

   The recommended value is: 6KB + (*n* ÷ 2)KB, where *n* is the number of virtual circuits. If you specify a value that is too low, unpredictable errors can occur. The default is **7**.

4. Adapter number.

   The parameter is optional. The parameter value ranges from 0 to 7. The default is **0**.

   IBM X.25 and other IBM Realtime Interface Co-Processor adapters are numbered sequentially, starting with 0, according to the physical slots into which they are plugged in the workstation. Empty slots and slots used by other adapters are not counted.

5. Profile name.

   This parameter specifies a profile created with the customization program provided with the X.25 Co-Processor Support program.

   If only one X25DLC2 server profile is defined, the parameter is optional. The parameter value is a string of up to 8 alphanumeric characters.

6. Your country identification.

   The parameter is optional. The parameter value is a number of three digits. Valid values for your country can be obtained from the telecommunications company. The default is **000**.

7. Your country subcode.

   The parameter is optional. The parameter value ranges from 0 to 9. Valid values for your country can be obtained from the telecommunications company. The default is **0**.

8. Your subscriber address.

   The parameter is required. The parameter value is a number of up to 11 digits.

X.25 data link control examples

```
/* Following is Example 1, which corresponds to keyword  */
/* specifications for the X25DLC server.                 */

         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=(X25NAT##),
                  CLIENT=(X25NAT01,AA),
                  SES&NSVC=(AA,01,OUTG,INXB,111111111),
/* Definition using the IBM PC X.25 Communications Adapter. */
                  SERVER=(X25DLC),
                  PAR&X25D=(20,Y,1,2,4,4,2,4,8,7,3,150,3,214,1,
                  231020107)
```

```
/* Following is Example 2, which corresponds to keyword  */
/* specifications for the X25DLC server.                 */

         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=(X25NAT##),
                  CLIENT=(X25NAT01,AA),
                  SES&NSVC=(AA,01,OUTG,INXB,111111111),
/* Definition using the IBM X.25 Interface Coprocessor/2. */
                  SERVER=(X25DLC2),
                  PAR&X252=(20,Y,7,3,PROFILE,214,1,231020107)
```

3270 emulator definitions

- LANDP for DOS
- LANDP for OS/2 (3270 emulator running in OS/2 MVDM)
- LANDP for Windows NT (3270 emulator running in Windows NT MVDM)

The 3270 emulator requires one PAR&3270 keyword, and as many SES&3270 keywords as 3270 emulated sessions are to be defined in the workstation.

To define several 3270 emulators in the same workstation with the same parameter values, only one PAR&3270 keyword is required.

Keyword **Description**

PAR&3270

Is a required keyword to define the 3270 emulator parameters, when the emulator is specified in a SERVER keyword.

You can specify up to 2 parameters:

1. 3270 emulator high level language application programming interface (HLLAPI) selection.

   The parameter value can be Y, to use HLLAPI, or N, not to use HLLAPI. The default is **N**.

2. Cryptographic support selection.

The parameter does not apply to DBCS mode. It is optional. The parameter value can be Y, to use the cryptographic support, or N, not to use it. The default is **N**.

Note: When working with LANDP for OS/2 or Windows NT workstations, the 3270 emulator can be used only in the appropriate MVDM. When working in DBCS mode, the 3270 emulator cannot be used in an MVDM, and thus it cannot run on a LANDP for OS/2 or Windows NT workstation.

SES&3270 Is a required keyword to define a 3270 emulator session, when the emulator is specified in a SERVER keyword. One keyword is required for each 3270 emulated session defined.

You can specify up to 13 parameters:

1. Name of the workstation providing SNA services.

   The parameter is required. The parameter value is a string of up to 2 alphanumeric characters.

2. 3270 emulated session number.

   The parameter is required. The parameter value ranges from 1 to 5, and must be unique for the workstation where the 3270 emulator will be installed. The value must match the 3270 emulator suffix, which identifies the session to be used (*x* value in EMU3270*x* servername).

3. SNA LU pool ID or LU number.

   To use the SNA session with LU pooling support, specify the LU pool ID. The parameter value is a string of two characters: the first character must be alphabetical, and the second one alphanumeric.

   To use the session without LU pooling support, when the SNA services are provided from a DOS workstation, specify the LU number. The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit.

   Omit the parameter to use the session without LU pooling support, when the SNA services are provided from an OS/2, Windows NT, or RS/6000-type workstation.

4. DLC type used by the workstation providing SNA services.

   The parameter applies only to DOS workstations. In this case, it is required except for workstations that have one DLC only, which is not X.25.

   For workstations with DLC other than X.25, to specify the type of DLC used, the parameter value can be:

   SDL Synchronous Data Link Control (SDLC)
   TKR Token-ring
   DCA Device Cluster Attachment (DCA)

For workstations with DLC being X.25, the parameter value is the number you have assigned as the virtual circuit definition identification in a SES&SPVC keyword or SES&SSVC keyword.

5. Display color attributes.

   This parameter identifies the color attributes table used for the workstation display.

   The parameter value is a string of three alphanumeric characters, which is the identifier of the color attributes table. It must match the value assigned in the EXTEN keyword in a DISPLATT vector with TYPE = 3270. Specify **ATR** to use the default table.

6. Display translation table.

   This parameter identifies the table that is used to translate from host computer EBCDIC to personal computer system ASCII, when receiving from the host computer.

   The parameter value is a string of three alphanumeric characters, which is the identifier of the translation table. It must match the value assigned in the EXTEN keyword in a XLATETBL vector with TYPE = EA3270. Specify **DIS** to use the default table.

   In DBCS mode, this parameter applies only to single-byte characters (SBCS). DBCS characters are translated by the DBTR server.

7. Keyboard translation table.

   This parameter identifies the translation table used for the workstation keyboard.
   The parameter value is a string of three alphanumeric characters, which is the identifier of the translation table. It must match the value assigned in the EXTEN keyword in a KBD3270 vector. Specify **KBD** to use the default table.

   There must also be an XLATETBL vector of type AE3270 and a KBD3270X vector with keyword EXTEN.

8. Host application session identification.

   This parameter identifies the session that is used for the application program in the host computer.
   The parameter is optional. If specified, it overrides the value assigned in the E3270HKx keyword in the DEFAULTS vector. The parameter value is a string of up to 8 alphanumeric characters.

9. Alternate screen height.

   This parameter specifies the height (number of rows) of the 3270 alternate screen to be emulated. The value can be in the range 24 through 49. This should not include the operator information area line at the bottom of the screen. The default is 24.

10. Alternate screen width.

This parameter specifies the width (number of columns) of the 3270 alternate screen to be emulated. The value can be 80 or 132. The default is 80. For LANDP for Windows NT workstations, only 80 is supported.

11. Blinking.

    This parameter indicates whether blinking text is supported. The parameter value can be Y (use blinking) or N (no blinking). The default is N.

12. Print screen.

    This parameter indicates whether the 3270 emulator should handle the Print Screen key. The parameter value can be Y or N. The default is Y.

13. Host buffer size.

    This parameter specifies the buffer size to be used for communication with the host computer. The parameter value can be in the range 2048 through 4096. The default is 2048. The parameter value specified must match the RU size detailed in the bind session.

3270 emulator example

```
         LWSCONF NAME=AA,
                 TYPE=DOS,
                 SERVER=(SNA##,,N),
                 PAR&SNA=(APPL,SRV),
                 SERVER=(TRDLC),
                 PAR&TKR=(48,04,04,00999999,00111111,017,00000),
/* 3270 emulator server definition. Example. */
                 SERVER=(EMU32701),
                 PAR&3270=(N,N),
                 SES&3270=(AA,1,011,TKR,ATR,DIS,KBD,,27,132,Y,N,4096)

         LWSCONF NAME=BB,
                 TYPE=DOS,
/* 3270 emulator server definition. Example. */
                 SERVER=(EMU32701),
                 PAR&3270=(N,N),
                 SES&3270=(AA,1,012,TKR,ATR,DIS,KBD,,,,,,)
```

3287 printer emulator definitions

- LANDP for DOS
- LANDP for OS/2 (3287 printer emulator running in OS/2 MVDM)
- LANDP for Windows NT (3287 printer emulator running in Windows NT MVDM)

The 3287 printer emulator requires one PAR&3287 keyword, and as many SES&3287 keywords as 3287 printer emulated sessions are to be defined in the workstation.

When working in DBCS mode, the information related to IBM 4201 Proprinter applies to IBM 5575 Printer or IBM 5577 Printer; the information related to IBM 4712 Transaction Printer, IBM 4722 Document Printer, IBM 4009 Universal Banking Printer, IBM 4772 Universal Financial Printer, or IBM 9068-D01 Multi-Purpose Passbook Printer applies to IBM 4748 Document Printer.

Note: When working with LANDP for OS/2 or Windows NT workstations, the 3270 emulator can be used only in an OS/2 or Windows NT MVDM respectively. When working in DBCS mode, the 3270 emulator cannot be used in an MVDM, and thus it cannot run on LANDP for OS/2 or Windows NT workstations.

Keyword Description

PAR&3287

Is a required keyword to define the 3287 printer emulator parameters, when the emulator is specified in a SERVER keyword.

You can specify up to 5 parameters:

1. Printer selection.

   This parameter specifies the printer to be used for output.

   The parameter is optional. The parameter value can be:

   1 IBM 4201 Proprinter, or equivalent device
   2 IBM 4712 Transaction Printer, IBM 4722 Document Printer, IBM 4009 Universal Banking Printer, or IBM 4772 Universal Financial Printer
   3 IBM 4019, 4029, or 4039 Laser Printer, in HP/PCL mode.

   The default is **1**.

2. Printer character string for the 3287 printer emulator.

   The parameter value is a string of three alphanumeric characters, which is the identifier of the EBCDIC to ASCII translation table. It must match the value assigned in the EXTEN keyword in a XLATETBL vector with TYPE = EA3287.

   In DBCS mode the parameter does not apply. It is optional. The default is **PRT**.

3. Printer 3287 attributes table.

   The parameter value is the extension of the file containing the printer 3287 attributes table. It must match the value assigned in the EXTEN keyword in the P3287ATT vector.

   It is optional. The default is **TAB**.

4. Time interval of polling.

   This parameter specifies the amount of time, in seconds, the 3287 printer emulator waits before checking for a file to be printed.

   The time interval you specify is used only when the 3287 printer emulator is idle. However, when there is no file to be printed, a short interval deteriorates performance.

The parameter is optional. The parameter value ranges from 1 to 60. The default is **15**.

5. Number of printers supported.

   The parameter is optional. The parameter value ranges from 1 to 3. Specify:

   - 1 to use LPT1 only
   - 2 to use LPT1 and LPT2
   - 3 to use LPT1, LPT2, and LPT3

   The default is the number of parallel ports available on the workstation.

SES&3287

Is a required keyword to define a 3287 printer emulator session, when the emulator is specified in a SERVER keyword. One keyword is required for each 3287 printer emulated session defined.

You can specify up to 9 parameters:

1. Name of the workstation providing SNA services.

   The parameter is required. The parameter value is a string of up to 2 alphanumeric characters.

2. 3287 printer emulated session number.

   The parameter is required. The parameter value ranges from 1 to 5, and must be unique for the workstation where the 3287 printer emulator will be installed.

3. SNA LU pool ID or LU number.

   To use the SNA session with LU pooling support, specify the LU pool ID. The parameter value is a string of two characters: the first character must be alphabetical, and the second one alphanumeric.

   To use the session without LU pooling support, when the SNA services are provided from a DOS workstation, specify the LU number. The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit.

   Omit the parameter to use the session without LU pooling support when the SNA services are provided from an OS/2, Windows NT, or RS/6000-type workstation.

4. DLC type used by the workstation providing SNA services.

   The parameter applies only when the SNA services are provided from a DOS workstation. In this case, it is required except for workstations that have one DLC only, which is not X.25.

   For workstations with DLC other than X.25, to specify the type of DLC used, the parameter value can be:

   SDL Synchronous Data Link Control (SDLC)
   TKR Token-ring

DCA Device Cluster Attachment (DCA)

For workstations with DLC being X.25, the parameter value is the number you have assigned as the virtual circuit definition identification in a SES&SPVC keyword or SES&SSVC keyword.

5. Printer identification.

   The parameter is optional. The parameter value can be:

1	LPT1
2	LPT2
3	LPT3

   The default is **1**.

6. Characters density.

   This parameter specifies the default characters density, in characters per inch (CPI), for the 3287 printer emulator. The density can be modified at run-time using the CM command of the operator interface, or the local resource manager server.

   The parameter is optional. The parameter value can be:

10	Normal
12	Medium
17	Condensed

   The default is **17**.

   When working with IBM 5575 Printer or IBM 5577 Printer, condensed density means 15 CPI.

7. Page length.

   This parameter must match the length, in inches, of the paper page to be used, and the switch settings in the printer attached to the workstation.

   The parameter is optional. The parameter value can be 11 or 12.

   The default is **11**.

8. Form feed before every listing selection.

   The parameter is optional. The parameter value can be Y, to perform a form feed before every listing, or N, not to perform it. The default is **N**.

9. Host application session identification.

   This parameter identifies the session that is used for the application program in the host computer.

   The parameter is optional. If specified, it overrides the value assigned in the E3287SIx keyword in the DEFAULTS vector. The parameter value is a string of up to 8 alphanumeric characters.

3287 printer emulator example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=(SNA##,,N),
                  PAR&SNA=(APPL,SRV),
                  SERVER=(TRDLC),
                  PAR&TKR=(48,04,04,00999999,00111111,017,00000),
/* 3287 printer emulator server definition. Example. */
                  SERVER=(EMU3287),
                  PAR&3287=(1,PRT,TAB),
                  SES&3287=(AA,1,017,TKR,1,17,11,N)

         LWSCONF  NAME=BB,
                  TYPE=DOS,
/* 3287 printer emulator server definition. Example. */
                  SERVER=(EMU3287),
                  PAR&3287=(1,PRT,TAB),
                  SES&3287=(AA,1,018,TKR,1,17,11,N)
```

Financial printer server definitions

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT
- LANDP for AIX

When the financial printer server is loaded in a DOS, OS/2, or Windows NT workstation, it requires one PAR&47X2 keyword for the workstation that provides services.

No matter the operating system in the server workstation, the server requires as many SES&47X2 keywords as printer server sessions are to be defined in the client workstation.

Keyword **Description**

PAR&47X2

Is a required keyword to define the financial printer server parameters, when the server is specified in a SERVER keyword.

You can specify up to 17 parameters:

1. Parallel port usage.

   The parameter is optional. The parameter value can be:

   C Common usage
   N Not used

   The default is **N**.

2. Port 1 usage.

   The parameter is optional. The parameter value can be:

   C Common usage

S	Shared (A/B) usage
N	Not used

The default is **N**.

3. Baud rate for port 1.

   This parameter specifies the number of bits per second the printer device driver sends data to the port when it receives data to be printed.

   Select a lower baud rate, if necessary, based on the cable length and the working environment.

   The parameter applies only when using port 1. It is optional. The parameter value can be:

   150
   300
   600
   1200
   2400
   4800
   9600

   The default is **9600**.

4. Printer model for port 1.

   The parameter is optional. The parameter value can be:

4009	4009 printer
4712	4712 or 9069 printer
4722	4722 printer
4772	4772 or 9068 printer
9069	9069 printer

   The default is **4722**.

5. REMS selection for port 1.

   The parameter applies only when using port 1. It is optional. The parameter value can be Y, to specify that REMS is installed, or N, to specify that REMS is not installed. The default is **N**.

6. Port 2 usage.

   The parameter is optional. The parameter value can be:

C	Common usage
S	Shared (A/B) usage
N	Not used

   The default is **N**.

7. Baud rate for port 2.

This parameter specifies the number of bits per second the printer device driver sends data to the port when it receives data to be printed.

Select a lower baud rate, if necessary, based on the cable length and the working environment.

The parameter applies only when using port 2. It is optional. The parameter value can be:

150
300
600
1200
2400
4800
9600

The default is **9600**.

8. Printer model for port 2.

   The parameter is optional. The parameter value can be:

4009	4009 printer
4712	4712 or 9069 printer
4722	4722 printer
4772	4772 or 9068 printer
9069	9069 printer

   The default is **4722**.

9. REMS selection for port 2.

   The parameter applies only when using port 2. It is optional. The parameter value can be Y, to specify that REMS is installed, or N, to specify that REMS is not installed. The default is **N**.

10. Port 3 usage.

    The parameter is optional. The parameter value can be:

C	Common usage
S	Shared (A/B) usage
N	Not used

    The default is **N**.

11. Baud rate for port 3.

    This parameter specifies the number of bits per second the printer device driver sends data to the port when it receives data to be printed.

    Select a lower baud rate, if necessary, based on the cable length and the working environment.

The parameter applies only when using port 3. It is optional. The parameter value can be:

150
300
600
1200
2400
4800
9600

The default is **9600**.

12. Printer model for port 3.

    The parameter is optional. The parameter value can be:

4009	4009 printer
4712	4712 or 9069 printer
4722	4722 printer
4772	4772 or 9068 printer
9069	9069 printer

    The default is **4722**.

13. REMS selection for port 3.

    The parameter applies only when using port 3. It is optional. The parameter value can be Y, to specify that REMS is installed, or N, to specify that REMS is not installed. The default is **N**.

14. Port 4 usage.

    The parameter is optional. The parameter value can be:

C	Common usage
S	Shared (A/B) usage
N	Not used

    The default is **N**.

15. Baud rate for port 4.

    This parameter specifies the number of bits per second the 4712 or 4722 printer device driver sends data to the port when it receives data to be printed.

    Select a lower baud rate, if necessary, based on the cable length and the working environment.

    The parameter applies only when using port 4. It is optional. The parameter value can be:

150
300
600
1200
2400

4800
9600

The default is **9600**.

16. Printer model for port 4.

    The parameter is optional. The parameter value can be:

4009	4009 printer
4712	4712 or 9069 printer
4722	4722 printer
4772	4772 or 9068 printer
9069	9069 printer

    The default is **4722**.

17. REMS selection for port 4.

    The parameter applies only when using port 4. It is optional. The parameter value can be Y, to specify that REMS is installed, or N, to specify that REMS is not installed. The default is **N**.

SES&47X2

Is a required keyword to define a financial printer server session, when the server is specified in a CLIENT keyword. One keyword is required for each printer server session defined.

You can specify up to 7 parameters:

1. Name of the workstation providing printer services.

   The parameter is required. The parameter value is a string of up to 2 alphanumeric characters.

2. Session identification.

   This value is used by a server or application to identify the printer session.

   The parameter is required. The parameter value is a string of two alphanumeric characters. For each workstation, it must be unique for this service, and must match the ## value of the financial printer server in the CLIENT keyword.

3. Port identification.

   If financial printer services are provided from an RS/6000 workstation, the parameter does not apply. The parameter is required for OS/2, DOS, or Windows NT. The parameter value can be:

0	Parallel port
1	Port 1
2	Port 2
3	Port 3
4	Port 4

The value must match the port usage in the PAR&47X2 keyword for the server workstation. Value 4 applies only to DOS, OS/2, and Windows NT workstations.

4. Port usage.

   The parameter is required. The parameter value can be:

   A Shared A
   B Shared B
   C Common

   The value must match the port usage in the PAR&47X2 keyword for the server workstation, provided it runs DOS, OS/2, or Windows NT. If the parallel port is used, only value C can be specified.

5. Length of data to be printed.

   This parameter specifies the maximum number of KB to be printed at a time.

   If financial printer services are provided from an RS/6000 workstation, the parameter does not apply. The parameter is optional. The parameter value ranges from 1 to 4. The default is **1**.

   The value to be used for each printer is the maximum value specified for all the sessions defined for that printer.

6. Device name.

   This parameter specifies where the device is attached.

   If the financial printer services are provided from a DOS, OS/2, or Windows NT workstation, the parameter does not apply. The parameter is required. The parameter value can be:

   - If an Xstation is specified, a string of up to 25 characters, without blanks and enclosed within quotes.
   - If a TTY device is specified, the following, within quotes:

     ```
     \dev\tty\string
     ```

     where *string* is a string of up to 18 characters, without blanks.

7. Port number.

   This parameter specifies the port number of the serial port in the Xstation.

   If the financial printer services are provided from a DOS, OS/2, or Windows NT workstation, the parameter does not apply. If the device name specified through parameter 6 specifies a TTY device, the parameter does not apply, either.

   The parameter is required. The parameter value ranges from 1 to 4.

Financial printer server example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
/* 4712/22 printer server definition. Example. */
                  SERVER=(PR47X2##),
                  PAR&47X2=(N,C,9600,4722,N,C,9600,4722,Y)

         LWSCONF  NAME=BB,
                  TYPE=OS/2,
/* 4712/22 printer client definition. Example. */
                  CLIENT=(PR47X2JO,AA),
                  SES&47X2=(AA,JO,1,C)
```

IBM 4721 self-service document printer

- LANDP for DOS

The 4721 printer requires one PAR&SP21 keyword to define the printer server parameters, and one PAR&PT21 keyword to define the printer integrator parameters.

Keyword **Description**

PAR&SP21

Is a required keyword to define the IBM 4721 Self-Service Document Printer server parameters, when the server is specified in a SERVER keyword.

You can specify two parameters:

1. Number of 4721 printers.

   This parameter specifies how many 4721 printers are managed by the server. The parameter value ranges from 1 to 3. The default is **1**.

2. 4721 printer ID.

   This parameter specifies the first two characters of the 4721 terminal address that is used for the connection between the 4721 printer server and the served 4721 printers.

   The parameter is optional. The parameter value must be unique in the network, for each 4721 printer server and the served 4721 printers. This enables you to configure several LANDP workgroups in the same network where every workstation with the 4721 printer server has the same ID.

   The parameter value is a string of two alphanumeric characters. The default is the name of the workstation (workstation ID) where the 4721 printer server is installed.

   The customization program creates an ASCII run-time file, named DEVSP47.TAB, which contains the information.

   Note: The parameter is supported by 4721 Server V. 2.00 or higher.

PAR&PT21

Is a required keyword to define the 4721 printer integrator parameters, when the server is specified in a SERVER keyword.

You can specify up to 8 parameters:

1. Name of the workstation providing SNA services.

   The parameter is required. The parameter value is a string of up to 2 alphanumeric characters.

2. 4721 printer number.

   This parameter identifies the 4721 printer to be used.

   The parameter is required. The parameter value ranges from 1 to 3, and must take into account the value assigned in the PAR&SP21 keyword.

3. SNA LU pool ID or LU number for session 1.

   To use the SNA session with LU pooling support, specify the LU pool ID. The parameter value is a string of two characters: the first character must be alphabetical, and the second one alphanumeric.

   To use the session without LU pooling support, when the SNA services are provided from a DOS workstation, specify the LU number. The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit.

   To use the session without LU pooling support, when the SNA services are provided from an OS/2, Windows NT, or RS/6000-type workstation, specify Y.

   If you do not wish to use the session, omit the parameter.

4. DLC type used by the workstation providing SNA services for session 1.

   The parameter applies only to DOS workstations. In this case, it is required except for workstations that have one DLC only, which is not X.25.

   For workstations with DLC other than X.25, to specify the type of DLC used, the parameter value can be:

   SDL Synchronous Data Link Control (SDLC)
   TKR Token-ring
   DCA Device Cluster Attachment (DCA)

   For workstations with more than one X.25DLC, to specify the particular X.25DLC used, the parameter value ranges from 1 to 999 and must match the virtual circuit definition defined for the SNA server in the SES&SPVC or SES&SSVC keyword.

5. SNA LU pool ID or LU number for session 2.

   To use the SNA session with LU pooling support, specify the LU pool ID. The parameter value is a string of two characters: the first character must be alphabetical, and the second one alphanumeric.

To use the session without LU pooling support, when the SNA services are provided from a DOS workstation, specify the LU number. The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit.

To use the session without LU pooling support, when the SNA services are provided from an OS/2, Windows NT, or RS/6000-type workstation, specify Y.

If you do not wish to use the session, omit the parameter.

6. DLC type used by the workstation providing SNA services for session 2.

   The parameter applies only to DOS workstations. In this case, it is required except for workstations that have one DLC only, which is not X.25.

   For workstations with DLC other than X.25, to specify the type of DLC used, the parameter value can be:

SDL	Synchronous Data Link Control (SDLC)
TKR	Token-ring
DCA	Device Cluster Attachment (DCA)

   For workstations with more than one X.25DLC, to specify the particular X.25DLC used, the parameter value ranges from 1 to 999 and must match the virtual circuit definition defined for the SNA server in the SES&SPVC or SES&SSVC keyword.

7. SNA LU pool ID or LU number for session 3.

   To use the SNA session with LU pooling support, specify the LU pool ID. The parameter value is a string of two characters: the first character must be alphabetical, and the second one alphanumeric.

   To use the session without LU pooling support, when the SNA services are provided from a DOS workstation, specify the LU number. The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit.

   To use the session without LU pooling support, when the SNA services are provided from an OS/2, Windows NT, or RS/6000-type workstation, specify Y.

   If you do not wish to use the session, omit the parameter.

8. DLC type used by the workstation providing SNA services for session 3.

   The parameter applies only to DOS workstations. In this case, it is required except for workstations that have one DLC only, which is not X.25.

   For workstations with DLC other than X.25, to specify the type of DLC used, the parameter value can be:

SDL Synchronous Data Link Control (SDLC)
TKR Token-ring
DCA Device Cluster Attachment (DCA)

For workstations with more than one X.25DLC, to specify the particular X.25DLC used, the parameter value ranges from 1 to 999 and must match the virtual circuit definition defined for the SNA server in the SES&SPVC or SES&SSVC keyword.

IBM 4721 printer example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=(SNA##,,N),
                  PAR&SNA=(APPL,SRV),
                  SERVER=(TRDLC),
                  PAR&TKR=(48,04,04,00999999,00111111,017,00000),
/* 4721 self-Service Document printer server definition. Example. */
                  SERVER=(SP4721##),
                  PAR&SP21=(1),
                  CLIENT=(SP4721##,AA),
                  SERVER=(PT4721),
                  PAR&PT21=(AA,1,005,TKR)
```

IBM 4731, 4738, 4739 personal banking machines

- LANDP for DOS
- LANDP for OS/2

The IBM 4731, 4738, 4739 Personal Banking Machine (PBM) requires one PAR&4731 keyword to define the PBM server.

Keyword **Description**

PAR&4731

Is a required keyword to define the IBM 4731, 4738, 4739 PBM server parameters, when the server is specified in a SERVER keyword.

You can specify up to 6 parameters:

1. Name of the workstation providing SNA services.

   The parameter is optional. It is required, if SNA services will be used.

   The parameter value is a string of up to 2 alphanumeric characters.

2. Configuration path.

   This parameter specifies the directory where the IBM 4731, 4738, 4739 PBM files are located in the production workstation at run-time.

   The parameter is required. The parameter value is a string of up to 30 alphanumeric characters. The format must be:

   ```
   D:\[directory1\[directory2\[directory3\]]]
   ```

   A maximum of three levels is permitted for the path.

3. SNA LU pool ID or LU number for operational session.

   To use the SNA session with LU pooling support, specify the LU pool ID. The parameter value is a string of two characters: the first character must be alphabetical, and the second one alphanumeric.

   To use the session without LU pooling support, when the SNA services are provided from a DOS workstation, specify the LU number. The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit.

   To use the session without LU pooling support, when the SNA services are provided from an OS/2, Windows NT, or RS/6000-type workstation, specify Y.

   If you do not wish to use the session, omit the parameter.

4. DLC type used by the workstation providing SNA services for operational session.

   The parameter applies only to DOS workstations. In this case, it is required except for workstations that have one DLC only, which is not X.25.

   For workstations with DLC other than X.25, to specify the type of DLC used, the parameter value can be:

   SDL Synchronous Data Link Control (SDLC)
   TKR Token-ring
   DCA Device Cluster Attachment (DCA)

   For workstations with DLC being X.25, the parameter value is the number you have assigned as the virtual circuit definition identification in a SES&SPVC keyword or SES&SSVC keyword.

5. SNA LU pool ID or LU number for utility session.

   To use the SNA session with LU pooling support, specify the LU pool ID. The parameter value is a string of two characters: the first character must be alphabetical, and the second one alphanumeric.

   To use the session without LU pooling support, when the SNA services are provided from a DOS workstation, specify the LU number. The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit.

   To use the session without LU pooling support, when the SNA services are provided from an OS/2, Windows NT, or RS/6000-type workstation, specify Y.

   If you not wish to use the session, omit this parameter.

6. DLC type used by the workstation providing SNA services for utility session.

The parameter applies only when the SNA services are provided from a DOS workstation. In this case, it is required except for workstations that have one DLC only, which is not X.25.

For workstations with DLC other than X.25, to specify the type of DLC used, the parameter value can be:

SDL Synchronous Data Link Control (SDLC)
TKR Token-ring
DCA Device Cluster Attachment (DCA)

For workstations with DLC being X.25, the parameter value is the number you have assigned as the virtual circuit definition identification in a SES&SPVC keyword or SES&SSVC keyword.

IBM 4731, 4738, 4739 PBM example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=(SNA##,,Y),
                  PAR&SNA=(ANY,SRV),
                  SERVER=(SDLC,,N),
                  PAR&SDLC=(8,C1,Y,N,N,03D,12345,CRTS),
/* 4731, 4738, 4739 Personal Banking Machine server definition. Example. */
                  SERVER=(SS#####,,N),
                  PAR&4731=(AA,C:\PROD\FBSS\WORK\,002,SDL,001,SDL)

         LWSCONF  NAME=BB,
                  TYPE=DOS,
/* 4731, 4738, 4739 Personal Banking Machine client definition. Example. */
                  CLIENT=(SS#####,AA)
```

IBM 4733 teller assist unit

- LANDP for DOS
- LANDP for OS/2

The IBM 4733 Teller Assist Unit requires one PAR&4733 keyword.

Keyword **Description**

PAR&4733
Is a required keyword to define the IBM 4733 Teller Assist Unit server parameters, when the server is specified in a SERVER keyword.

You can specify up to 2 parameters:

1. Line speed.

   The parameter is required. The parameter value can be:

   1200
   2400
   4800
   9600

The default is **9600**.

2. Port.

   The parameter is optional. The parameter value ranges from 1 to 3. The default is **1**.

IBM 4733 teller assist unit example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
/* 4733 Teller Assist Unit server definition. Example. */
                  SERVER=(DTAU4733),
                  PAR&4733=(9600,2)

         LWSCONF  NAME=BB,
                  TYPE=DOS,
/* 4733 Teller Assist Unit client definition. Example. */
                  CLIENT=(DTAU4733,AA)
```

IBM 4737 self-service transaction station

- LANDP for DOS
- LANDP for OS/2

The IBM 4737 Self-Service Transaction Station requires one PAR&4737 keyword, and one PAR&PBMS keyword to define the PBM support.

Keyword **Description**

PAR&4737
Is a required keyword to define the IBM 4737 Self-Service Transaction Station server parameters, when the server is specified in a SERVER keyword.

You can specify one parameter:

1. Configuration path.

   This parameter specifies the directory where the IBM 4737 files are located in the production workstation at run-time.

   The parameter is required. The parameter value is a string of up to 30 alphanumeric characters. The format must be:

   D:\[*directory1*\[*directory2*\[*directory3*\]]]

   A maximum of three levels is permitted for the path.

PAR&PBMS
Is a required keyword to define the PBM server parameters, when the server is specified in a SERVER keyword.

You can specify up to 3 parameters:

1. Name of the workstation providing SNA services.

The parameter is required. The parameter value is a string of up to 2 alphanumeric characters.

2. SNA LU pool ID or LU number.

   To use the SNA session with LU pooling support, specify the LU pool ID. The parameter value is a string of two characters: the first character must be alphabetical, and the second one alphanumeric.

   To use the session without LU pooling support, when the SNA services are provided from a DOS workstation, specify the LU number. The parameter value ranges from 1 to 254, and must be unique for the data link control (DLC) or virtual circuit.

   Omit the parameter to use the session without LU pooling support when the SNA services are provided from an OS/2, Windows NT, or RS/6000-type workstation.

3. DLC type.

   The parameter applies only if the SNA services are provided by a DOS workstation. In this case, it is required except for workstations that have one DLC only, which is not X.25.

   For workstations with DLC other than X.25, to specify the type of DLC used, the parameter value can be:

SDL	Synchronous Data Link Control (SDLC)
TKR	Token-ring
DCA	Device Cluster Attachment (DCA)

   For workstations with DLC being X.25, the parameter value is the number you have assigned as the virtual circuit definition identification in a SES&SPVC keyword or SES&SSVC keyword.

IBM 4737 station example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
                  SERVER=(SNA##,,Y),
                  PAR&SNA=(ANY,SRV),
                  SERVER=(SDLC,,N),
                  PAR&SDLC=(8,C1,Y,N,N,,,CRTS),
/* 4737 Self-Service Transaction Station server definition.
   Example. */
                  SERVER=(SS######,,N),
                  PAR&4737=C:\4737PBM\,
                  SERVER=(PBMS,,N),
                  PAR&PBMS=(AA,002,SDL),
                  CLIENT=(SS######,AA)

         LWSCONF  NAME=BB,
                  TYPE=DOS,
/* 4737 Self-Service Transaction Station client definition.
   Example. */
                  CLIENT=(SS######,AA)
```

IBM 4748 printer server definitions

- LANDP for DOS
- LANDP for OS/2
- LANDP for Windows NT

The 4748 printer server requires one PAR&4748 keyword, and as many SES&4748 keywords as printer server sessions are to be defined in the client workstation.

Under Windows NT, the 4748 printer server does not support the 4748 printer itself, but supports a 9055-01 or 9068-D01 printer when its ID byte is set to 1E(hexadecimal).

Keyword **Description**

PAR&4748

Is a required keyword to define the printer server parameters, when the server is specified in a SERVER keyword.

You can specify up to 12 parameters:

1. Port 1 usage.

   The parameter is optional. The parameter value can be:

   C Common usage
   S Shared (A/B) usage
   N Not used

   The default is **N**.

2. Baud rate for port 1.

This parameter specifies the number of bits per second the printer device driver sends to the port when it receives data to be printed.

Select a lower baud rate, if necessary, based on the cable length and the working environment.

The parameter applies only when using port 1. It is optional. The parameter value can be:

150
300
600
1200
2400
4800
9600

The default is **9600**.

3. Printer model for port 1.

   The parameter is optional. The parameter value can be:

   4748 4748 printer (not supported by LANDP for Windows NT)
   9055 9055 printer in native mode (ID=1E hex) or 4748 emulation mode (ID=1D (hex)). 4748 emulation is not supported under Windows NT.
   9068 printer configured as a 9055-D01 (ID=1E hex))

   The default is **4748**.

4. REMS selection for port 1.

   The parameter is optional, and applies only when the printer for port 1 is 9055. The parameter value can be Y, to specify that REMS is installed, or N, to specify that REMS is not installed. The default is **N**.

5. Port 2 usage.

   The parameter is optional. The parameter value can be:

   C Common usage
   S Shared (A/B) usage
   N Not used

   The default is **N**.

6. Baud rate for port 2.

   This parameter specifies the number of bits per second the printer device driver sends to the port when it receives data to be printed.

   Select a lower baud rate, if necessary, based on the cable length and the working environment.

   The parameter applies only when using port 2. It is optional. The parameter value can be:

   150

300
600
1200
2400
4800
9600

The default is **9600**.

7. Printer model for port 2.

   The parameter is optional. The parameter value can be:

4748	4748 printer (not supported by LANDP for Windows NT)
9055	9055 printer in native mode (ID=1E (hex)) or 4748 emulation mode (ID=1D (hex)). 4748 emulation is not supported under Windows NT. 9068 printer configured as a 9055-D01 (ID=1E hex))

   The default is **4748**.

8. REMS selection for port 2.

   The parameter is optional, and applies only when the printer for port 2 is 9055. The parameter value can be Y, to specify that REMS is installed, or N, to specify that REMS is not installed. The default is **N**.

9. Driver file name (COM.SYS or COMDMA.SYS).

   This parameter specifies the printer device driver.

   The parameter applies only to OS/2 workstations. It is optional. The parameter format is *xxxxxxxx.yyy*. The default is **COMDMA.SYS**.

10. Alert setting.

    This parameter specifies whether alerts are required. It is optional. The parameter value can be Y, to specify that alerts are required, or N, to specify that alerts are not required. The default is **N**.

11. Code conversion.

    This parameter specifies whether code conversion is supported. It is optional. The parameter value can be:

Y	Code conversion required
N	Code conversion not required

    The default is **Y**.

12. UDC file name.

    This parameter specifies the UDC file name. It is optional. The parameter format is *xxxxxxxx.yyy*. The default is **PR4748.UDC**.

SES&4748 Is a required keyword to define a printer server session, when the server is specified in a CLIENT keyword. One keyword is required for each printer server session defined.

You can specify up to 5 parameters:

1. Name of the workstation providing printer services.

   The parameter is required. The parameter value is a string of up to 2 alphanumeric characters.

2. Session identification.

   This value is used by a server or application to identify the printer session.

   The parameter is required. The parameter value is a string of two alphanumeric characters. For each workstation, it must be unique for this service.

3. Port identification.

   The parameter is required. The parameter value can be:

   1 Port 1
   2 Port 2

   The value must match the port usage in the PAR&4748 keyword for the server workstation.

4. Port usage.

   The parameter is required. The parameter value can be:

   A Shared A
   B Shared B
   C Common

   The value must match the port usage in the PAR&4748 keyword for the server workstation.

5. Length of data to be printed.

   This parameter specifies the maximum number of KB to be printed at a time.

   The parameter is optional. The parameter value ranges from 1 to 4. The default is **1**.

   The value to be used for each printer is the maximum value specified for all the sessions defined for that printer.

4748 printer server example

```
         LWSCONF  NAME=AA,
                  TYPE=DOS,
/* 4748 printer server definition. Example. */
                  SERVER=(PR4748##,,Y),
                  PAR&4748=(C,9600)

         LWSCONF  NAME=BB,
                  TYPE=OS/2,
/* 4748 printer client definition. Example. */
                  CLIENT=(PR474801,AA),
                  SES&4748=(AA,01,1,C)
```

IBM 4770 printer server definitions

- LANDP for OS/2

The 4770 printer server requires one PAR&4770 keyword, and as many SES&4770 keywords as printer server sessions are to be defined in the client workstation.

Keyword **Description**

PAR&4770

Is a required keyword to define the 4770 printer server parameters, when the server is specified in a SERVER keyword.

You can specify up to 7 parameters:

1. Parallel port usage.

   The parameter is optional. The parameter value can be:

   C Common usage
   N Not used

   The default is **N**.

2. Port 1 usage.

   The parameter is optional. The parameter value can be:

   C Common usage
   N Not used

   The default is **N**.

3. Baud rate for port 1.

   This parameter specifies the number of bits per second the 4770 printer device driver sends data to the port when it receives data to be printed.

   The parameter applies only when using port 1. It is optional. The parameter value can be:

   1200
   9600

   The default is **9600**.

4. Port 2 usage.

   The parameter is optional. The parameter value can be:

   C Common usage
   N Not used

   The default is **N**.

5. Baud rate for port 2.

   This parameter specifies the number of bits per second the 4770 printer device driver sends data to the port when it receives data to be printed.

   The parameter applies only when using port 2. It is optional. The parameter value can be:

   1200
   9600

   The default is **9600**.

6. Port 3 usage.

   The parameter is optional. The parameter value can be:

   C Common usage
   N Not used

   The default is **N**.

7. Baud rate for port 3.

   This parameter specifies the number of bits per second the 4770 printer device driver sends data to the port when it receives data to be printed.

   The parameter applies only when using port 3. It is optional. The parameter value can be:

   1200
   9600

   The default is **9600**.

SES&4770 Is a required keyword to define a 4770 printer server session, when the server is specified in a CLIENT keyword. One keyword is required for each printer server session defined.

You can specify up to 4 parameters:

1. Name of the workstation providing printer services.

   The parameter is required. The parameter value is a string of up to 2 alphanumeric characters.

2. Session identification.

   This value is used by a server or application to identify the printer session.

The parameter is required. The parameter value is a string of two alphanumeric characters. For each workstation, it has to be unique for this service and must match the ## value of the 4770 printer server in the CLIENT keyword.

3. Port identification.

   The parameter is required. The parameter value can be:

0	Parallel port
1	Port 1
2	Port 2
3	Port 3

   The value must match the port usage in the PAR&4770 keyword for the server workstation.

4. Length of data to be printed.

   This parameter specifies the maximum number of KB to be printed at a time.

   The parameter is optional. The parameter value ranges from 1 to 4. The default is **1**.

   The value to be used for each printer is the maximum value specified for all the sessions defined for that printer.

4770 printer server example

```
         LWSCONF  NAME=AA,
                  TYPE=OS/2,
/* 4770 printer server definition. Example. */
                  SERVER=(PR4770##),
                  PAR&4770=(N,C,9600)

         LWSCONF  NAME=BB,
                  TYPE=OS/2,
/* 4770 printer client definition. Example. */
                  CLIENT=(PR477001,AA),
                  SES&4770=(AA,01,1)
```

Editing model configuration data

Before editing model configuration data, identify your requirements:

- If you have workstation configurations that are not similar but use similar servers, you can define server model configuration vectors (SVRMODEL) for those servers.
- If you have workstation configurations in a workgroup that are similar, you can define workstation model configuration vectors (WSMODEL) for those workstations. If the workstations use similar servers, you can also define SVRMODEL vectors for those servers.
- If you have workgroup configurations that are similar, you can define workgroup model configuration vectors (LANMODEL) for those workgroups. Note that you

also have to define the model configuration vectors for all the workstations in the workgroup (LWSCONF).

If the workstations are similar, you can define WSMODEL vectors for those workstations. Likewise, if the workstations use similar servers, you can define SVRMODEL vectors for those servers.

To edit model configuration data:

- Start the EDITSPC procedure to edit the model configuration file (see “Creating and editing configuration vectors” on page 408)

  or

- Open a file with filename MODELS and extension SPC (MODELS.SPC), using any text editor. The MODELS.SPC file *must* be located in the EHCCUS directory (see “Customization directory structure” on page 37).

The EDITSPC procedure is provided to edit .SPC files.

Write the model configuration vectors on that file:

- Edit a LANMODEL vector to include information about a workgroup model configuration. You will have to write one LANMODEL vector for each workgroup.
- Edit one LWSCONF vector for each workstation in the workgroup model configuration, to include information about the workstation and the servers in that workstation. Those vectors *must* be placed right after the corresponding LANMODEL vector.

  The LWSCONF vector can contain a reference to a WSMODEL vector, some references to SVRMODEL vectors, and actual definitions.
- Edit one WSMODEL vector for each workstation model configuration to be defined, to include information about the workstation and the servers in that workstation.

  The WSMODEL vector can contain some references to SVRMODEL vectors, and actual definitions.
- Edit one SVRMODEL vector for each server model configuration to be defined, to include information about the server.

The following sections explain the vectors, their keywords, and the parameters corresponding to the keywords.

When you generate the MODELS.SPC file through the generation procedure, using the customization data stored in the internal repository, the order in which the vectors appear in the file may be different from the order in which you wrote them.

Model configuration vectors - one by one

Information about each particular vector, including an example, follows.

LANMODEL vector

Contains workgroup information, and must be placed in the MODELS.SPC file.

A quick reference

Vector position	None. It can be followed by up to 250 LWSCONF vectors.
List of keywords	MODNAME DEFAULTS, LANADAPT, LANGUAGE, SUFFIX, XPORT, TCPPORT, ADAPTNUM, NETBUFF, WSNAMES, NWSIDDUP, PARLIP
Vector relates to	LWSCONF (NAME keyword), LANCONF (MODEL keyword), DEFAULTS
Vector format	LANMODEL MODNAME=*xxxxxxxx*, [DEFAULTS=*xxxxxxxx*,] [LANADAPT=*x*,] [LANGUAGE=*xxx*,] [SUFFIX=*x*,] [XPORT=*x*,] [TCPPORT=*xxxxx*,] [ADAPTNUM=*x*,] [NETBUFF=*xx*,] [NWSIDDUP=*x*,] [PARLIP=*x*,] WSNAMES=( )

The keywords in the first line in the list identify the workgroup model configuration.

Other keywords define the workgroup model configuration. For information on these keywords, refer to "LANCONF vector" on page 411.

Keyword Description

MODNAME

Is a required keyword to specify the name of the workgroup model configuration.

The parameter value is a model name of up to 8 alphanumeric characters. The value is assigned in the MODEL keyword in the LANCONF vector to refer to the workgroup model configuration.

WSNAMES

Is a required keyword to specify the names of the workstations in the workgroup model configuration. This keyword can specify up to 250 parameters, which is the maximum number of workstations in a workgroup supported by the LANDP family.

Each parameter is the name of a workstation in the workgroup model configuration. The parameter value is a string of up to 2 alphanumeric characters. The value is assigned in the NAME keyword in the LWSCONF vector that contains information about the workstation.

Each workstation specified in the WSNAMES keyword requires one LWSCONF vector to define that particular workstation in the workgroup model configuration. For information on this vector, refer to “LWSCONF vector” on page 417.

LANMODEL vector example

```
LANMODEL MODNAME=MODEL01,
         WSNAMES=(AA,BB,CC),
         LANGUAGE=001

LWSCONF  NAME=AA,
         TYPE=OS/2,
         PRODLVL=L50,
         SERVER=(EHCSQL01),
         PAR&SQL=(CONFIGUR,5,4,10,Y,15)
LWSCONF  NAME=BB,
         TYPE=OS/2,
         PRODLVL=L50,
         CLIENT=(EHCSQL01,AA)
LWSCONF  NAME=CC,
         TYPE=OS/2,
         PRODLVL=L50,
         CLIENT=(EHCSQL01,AA)
```

WSMODEL vector

Contains information about a workstation model configuration.

A quick reference

Vector position	None
List of keywords	MODNAME TYPE, SYSLVL, PRODLVL, SOFTPACK, FRAME, RPLOAD, POOLSIZE, PARLIP, PARLIPEX, LANDPDCE, ADAPTNUM, NETBUFF, DBCSXLAT, PARWIN, SERVER, CLIENT PAR*xxxxx*, SES*xxxxx*
Vector relates to	SVRMODEL vector

Vector format	WSMODEL MODNAME=*xxxxxxx*, [TYPE=( ),] [SYSLVL=*x*,] [PRODLVL=*xxx*,] [SOFTPACK=*xxxxxxxx*,] [FRAME=*xxxx*,] [RPLOAD=*x*,] [POOLSIZE=*xxx*,] [PARLIP=( ),] [PARLIPEX=( ),] [LANDPDCE=*x*,] [ADAPTNUM=*x*,] [NETBUFF=*xx*,] [DBCSXLAT=*x*,] [PARWIN=*xx*,] [SERVER=( ),] [CLIENT=( ),] [PARxxxxx=( ),] [SESxxxxx=( )]

The MODNAME keyword identifies the workstation model configuration.

The keywords in the second line in the list define the workstation model configuration. The keywords in the third line define the servers related to that workstation. For information on all these keywords, refer to “LWSCONF vector” on page 417.

Keyword **Description**

MODNAME

Is a required keyword to specify the name of the workstation model configuration.

The parameter value can be any model name, of up to 8 alphanumeric characters.

WSMODEL vector example

```
WSMODEL  MODNAME=MODEL02,
         TYPE=OS/2,
         PRODLVL=L50,
         SERVER=(PR47X2##,MODEL03),
         CLIENT=(EHCSQL01,AA),
         CLIENT=(SNA02,AA),
         SES&SNA=(AA,02,019,001)
```

SVRMODEL vector

Contains information about a server model configuration.

A quick reference

Vector position	None
List of keywords	MODNAME SVRNAME, EXPMEM, SERVPARM PAR*xxxxx*, SES*xxxxx*
Vector relates to	None
Vector format	SVRMODEL MODNAME=*xxxxxxx*, SVRNAME=*xxxxxxxx*, [EXPMEM=*x*,] [SERVPARM=( ),] [PARxxxxx=( ),] [SESxxxxx=( )]

The MODNAME keyword identifies the server model configuration. The SVRNAME and the EXPMEM keywords identifies the server.

The other keywords define the server model configuration. For information on those keywords, refer to “LWSCONF vector” on page 417.

Keyword **Description**

MODNAME

Is a required keyword to specify the name of the server model configuration.

The parameter value can be any model name, of up to 8 alphanumeric characters.

SVRNAME Is a required keyword to specify the server name. The parameter value is a string of up to 8 alphanumeric characters.

Refer to the table in “Vectors - a quick reference” on page 54 for the names corresponding to the functional areas (string in parenthesis).

The following servers require that the server suffix (##) is substituted by the corresponding value to completely identify the server:

EHCMQ##
EHCODB##
EHCSQL##
ELECJO##
MSRE47##
PINP47##
PR47X2##
PR4770##
SHFILE##

The following servers require a suffix to identify the session to be used:

BIWP*x*
BPP*x*
EMU3270*x*

LDA7*x*

Note that BIWP and LDA 7 program need the suffix only when they are installed in a DOS workstation.

If the following servers are to be run in OS/2 MVDM, the server names to be specified are:

BIWP	VBIWP*x*
LDA 7 program	VLDA7*x*

where *x* is the suffix to identify the session.

EXPMEM Is an optional keyword to specify the expanded memory selection. The parameter applies only to DOS workstations.

The parameter value can be:

Y The server is loaded in *expanded memory*
N The server is loaded in *conventional memory*

The default is **N**.

SERVPARM
Is an optional keyword to specify a string that will be added to the loading statement of the server.

The parameter value is a string of up to 40 characters, enclosed between quotes.

SVRMODEL vector example

```
SVRMODEL MODNAME=MODEL03,
         SVRNAME=PR47X2##,
         PAR&47X2=(N,C,9600,4722,,C,9600,4722)
```

Appendix E. Using TCP/IP for internal communication

> **Note for DOS users**
>
> For DOS users, the statements in this appendix about TCP/IP apply equally to PC/TCP.

OS/2 DOS 6000 NT

You can configure a LANDP workgroup to use TCP/IP as its internal communication protocol. This means that LANDP workstations can be connected to a TCP/IP network and use TCP/IP to support the LANDP client/server mechanism.

The low-level communication protocols used by the TCP/IP implementation in each environment (token-ring, Ethernet, X.25, and so on) are transparent to LANDP and exclusively controlled by TCP/IP.

LANDP uses standard TCP/IP functions and facilities. This requires a properly implemented TCP/IP network. For details on installing and configuring a TCP/IP network, refer to the appropriate TCP/IP support product manuals.

Because LANDP internal communication is transparent to clients and servers, no special considerations apply when developing applications or user servers.

LANDP Internet Protocol (LIP)

LANDP uses the standard TCP/IP user datagram protocol (UDP). LANDP adds extensions to this protocol to ensure reliable communication. The resulting protocol (and the LANDP server that implements it) is referred to as the LANDP Internet Protocol (LIP).

The following sections describe the LIP, and provide configuration information.

LIP workstation ID definition

LIP refers to each workstation in the workgroup by its LIP name. The LIP name of a workstation consists of the prefix 'LIP-' followed by the workstation id and optionally followed by the workgroup name if SUFFIX=Y is specified in the LANCONF configuration vector.

For example: the LIP name of workstation 'S1' in workgroup 'BRANCH3' would be 'LIP-S1BRANCH3' if SUFFIX=Y were specified and 'LIP-S1' otherwise.

The LIP names are used as TCP/IP host names or host alias names for the workstations.

Within a LANDP workgroup the workstations that need to communicate with any particular workstation are called its related workstations. These are the workstations to which it provides services, or from which it receives services.

LIP uses TCP/IP to resolve the LIP name of every related workstation into an IP address. This requires the LIP names and corresponding IP addresses to be defined to a TCP/IP names server or in a local HOSTS file, but see "Unresolved workstation IDs" on page 526.

A sample of a HOSTS file follows:

```
214.1.1.51   basement     LIP-AAWHAREHS
214.1.1.55   bossdesk     LIP-BBWHAREHS
214.1.1.67   firstfloor   LIP-CCWHAREHS
```

This sample corresponds to a LANDP workgroup with the following characteristics:

- The workgroup name is: WHAREHS
- SUFFIX=Y has been specified on the LANCONF vector
- There are three LANDP related workstations: AA, BB, and CC
- TCP/IP host names of the workstations are: basement, bossdesk, and firstfloor
- IP addresses are: 214.1.1.51, 214.1.1.55, and 214.1.1.67

Note: If the HOSTS file or the NAMES server is changed, you have to load the LIP again.

TCP/IP verification

Before attempting to use LANDP for the first time, you should verify TCP/IP communications and definitions with the PING TCP/IP program. On each workstation, after TCP/IP has been started, issue the PING command with each related workstation ID.

For example, on a workstation AA belonging to a LANDP workgroup with the name WHAREHS, and related to LANDP workstations with the IDs BB and CC, you must issue:

```
PING LIP-AAWHAREHS
PING LIP-BBWHAREHS
PING LIP-CCWHAREHS
```

PING should report successful contact with every workstation. If it does not, either the network or the TCP/IP definitions are not ready.

For detailed information on the PING program, refer to the TCP/IP manuals. See "Bibliography" on page 571.

You can also perform this verification at production time when two workstations are unable to contact each other.

Unresolved workstation IDs

Unresolved workstation IDs can occur when not all workstations in the workgroup are in existence, or when DHCP (Dynamic Host Configuration Protocol) is being used to dynamically assign IP addresses.

In normal operation if any LIP name cannot be resolved then LANDP loading terminates. However when LIP address checking is turned off (with the PARLIP keyword or the /J parameter of EHCLIP) then unresolved LIP names are logged but LANDP loading continues.

Turning off LIP address checking is useful when not all workstations in the workgroup are in existence, or when DHCP (Dynamic Host Configuration Protocol) is being used to dynamically assign IP addresses. When LIP names cannot be resolved there will not, initially, be any attempt at communication with the workstations with unresolved LIP names. However, if any of these workstations make contact then communications will resume.

DHCP (Dynamic Host Configuration Protocol)

In an environment where DHCP is being used there are some alternative configurations to choose from:

- One option is to use DDNS or WINS to dynamically associate DHCP assigned IP addresses with LIP names. In this situation LIP should be loaded without LIP address checking in all workstations.
- Another option, not using DDNS or WINS, is to configure workstations providing services (servers) to have fixed IP addresses resolvable from the LIP name via a names server and all the other workstations (clients) to obtain their IP addresses from DHCP. In this situation LIP should be loaded without LIP address checking in the servers and with LIP address checking in the clients.

Port number

The LIP uses one UDP port for communication in each workstation. The port number must be the same for all the workstations in a LANDP workgroup. If the default port number conflicts with other application requirements it can be changed during customization or at loading time.

Workstation identification

The LIP establishes and maintains a session with each related LANDP workstation in the workgroup as defined at customization.

When the session is established, workstations identify themselves to each other to verify the LANDP configuration. If a mismatch is detected in the workstation identification or the corresponding INET addresses, the session is not established and an error message is logged.

Data interchange

Data is exchanged by sending and receiving datagrams. Datagrams are acknowledged by the destination workstation. Datagrams that are not acknowledged are retransmitted until they are acknowledged or until the LIP detects that contact is lost. If this happens, the session is reset, an attention message is logged, and the procedure to re-establish contact is started.

Both ends of a session can send data simultaneously at any time.

Depending on the message size, messages exchanged during a session are sent in one or more datagrams. The LIP stores information about each datagram in a retransmission table that is used if the datagram has to be retransmitted.

In LANDP for DOS, the size of this table is defined at customization or with a LIP loading parameter. If there is no space left in the retransmission table, the datagram cannot be sent and the process has to be retried later.

Message segmentation

The default datagram length used by the LIP is 1472 bytes on DOS and 4272 bytes on other operating systems. The maximum datagram lengths are 32711 and 32767 respectively. To support messages longer than the datagram length, a segmentation mechanism divides them into segments and sends each segment as a datagram. The segments are reassembled at the receiving end. This process consumes time and resources. Therefore, the overall performance can be affected.

Session partner availability

The LIP can detect whether a session partner is available during the periods of time when there is no traffic over the network. The LIP sends short availability probe datagrams at regular intervals to the related workstations. If these datagrams are not acknowledged, the LIP starts the session reset and re-establishment procedures.

You can disable this feature with a LIP parameter. However, this is not recommended in environments where clients can lock server resources. If a client locks a resource and then becomes unavailable, the resources would remain locked since the server would not be able to detect the lost session.

Communication errors

LANDP provides trace tools to diagnose errors and track problems. These tools use trace servers to log information about internal and external errors, and provide statistics.

For more information about these tools, see *LANDP Problem Determination*.

Appendix F. Host communication definitions

This appendix contains information for a possible host set-up that you will need when working with LANDP. You must modify these examples for your own host system.

ACF/NCP (Read by VTAM)

Three examples are provided that correspond to LANDP SNA/SDLC, SNA/X.25, and SNA/TRDLC configurations.

SNA/SDLC

The access method configuration must match the LANDP SNA/SDLC configuration set up during customization. Following is an example for ACF/VTAM®.

Note: This is only an example; you *must* tailor it to your environment and requirements.

```
WPCGR20  GROUP LNCTL=SDLC,
               DIAL=NO,
               TYPE=NCP,
               ANS=CONTINUE,
               DATRATE=LOW,
               DISCNT=NO,
               DUPLEX=FULL,
               IRETRY=NO,
               ISTATUS=INACTIVE,
               MAXDATA=265,
               MAXOUT=2,
               NEWSYNC=NO,
               NRZI=NO,
               PASSLIM=254,
               PUTYPE=2,
               PUDR=NO,
               REPLYTO=2.0,
               RETRIES=(3,15,40),
               SERVLIM=10,
               SPDSEL=NO,
               TEXTTO=1.0

WL1PC    LINE ADDRESS=001,
              SPEED=9600,
              CLOCKNG=EXT

WSRV1    SERVICE ORDER=(WP1AA,WP1AC)

WP1AA    PU ADDR=AA
```

```
FU1AA01  LU LOCADDR=01,
            MODETAB=TABPCLU,
            SSCPFM=FSS,
            DLOGMOD=ENTRYF,
            PACING=0,
            VPACING=0

FU1AA41  LU LOCADDR=41,
            MODETAB=TABPCLU,
            SSCPFM=FSS,
            LOGAPPL=CICS2,
            DLOGMOD=ENTRYP,
            PACING=1,
            VPACING=0

FU1AA37  LU LOCADDR=37,
            MODETAB=TABPCLU,
            SSCPFM=FSS,
            DLOGMOD=ENTRYF,
            PACING=0,
            VPACING=0

FU1AA81  LU LOCADDR=81,
            MODETAB=TABPCLU,
            SSCPFM=USSSCS,
            USSTAB=ROBTAB,
            DLOGMOD=ENTRYZ,
            PACING=0,
            VPACING=0

WP1AC    PU ADDR=AC,
            MAXOUT=7,
            PACING=7,
            PASSLIM=7,
            VPACING=7

FU1AC01  LU LOCADDR=01,
            MODETAB=MODCRYP,
            DLOGMOD=TABDSX11,
            SSCPFM=USSSCS
```

SNA/X.25

The access method configuration must match the LANDP SNA/X.25 configuration set up during customization. Following is an example for ACF/VTAM.

Note: This is only an example; you *must* tailor it to your environment and requirements.

```
SWPCX25  VBUILD TYPE=SWNET,
               MAXGRP=10,
               MAXNO=50

SPF01    PU    ADDR=AA,
               IDBLK=017,
               IDNUM=80001,
               ISTATUS=ACTIVE,
               DISCNT=(YES,F),
               MAXDATA=265,
               MAXPATH=8,
               PUTYPE=2,
               SSCPFM=USSSCS,
               VPACING=2

         PATH  DIALNO=C3231020108,PID=1,
               GRPNM=XF1SNET

SUF0101  LU    LOCADDR=01,
               MODETAB=TABPCLU,
               SSCPFM=FSS,
               DLOGMOD=ENTRYF,
               PACING=0,
               VPACING=0

SUF0141  LU    LOCADDR=41,
               MODETAB=TABPCLU,
               SSCPFM=FSS,
               LOGAPPL=CICS2,
               DLOGMOD=ENTRYP,
               PACING=1,
               VPACING=0

SUF0137  LU    LOCADDR=37,
               MODETAB=TABPCLU,
               SSCPFM=FSS,
               DLOGMOD=ENTRYK,
               ENCR=REQD,
               PACING=0,
               VPACING=0

SUF0181  LU    LOCADDR=81,
               MODETAB=TABPCLU,
               SSCPFM=USSSCS,
               USSTAB=ROBTAB,
               DLOGMOD=ENTRYZ,
               PACING=0,
               VPACING=0
```

SNA/TRDLC

The access method configuration must match the LANDP SNA/TRDLC configuration set up during customization. Following is an example for ACF/VTAM.

Note: This is only an example; you *must* tailor it to your environment and requirements.

```
SWPCTOKE VBUILD TYPE=SWNET,
               MAXGRP=2,
               MAXNO=25

SPT11    PU    ADDR=1,
               IDBLK=05D,
               IDNUM=11111,
               PUTYPE=2,
               MODETAB=TABPCLU,
               MAXPATH=2

SDT1120  PATH  DIALNO=0104400011111111,
               GRPNM=GTL120

SDT1145  PATH  DIALNO=0104400011111111,
               GRPNM=GTL145

SUT111   LU    LOCADDR=1,
               USSTAB=ROBTAB,
               DLOGMOD=ENTRYF

SUT1137  LU    LOCADDR=37,
               USSTAB=ROBTAB,
               DLOGMOD=ENTRYF

SUT1235  LU    LOCADDR=35,
               USSTAB=ROBTAB,
               MODETAB=MODCRYP,
               DLOGMOD=TABDSX11

SUT1500  LU    LOCADDR=0,
               DLOGMOD=FBSSMODE,
               SPAN=(450),
               RESSCB=2

SUT1590  LU    LOCADDR=90,
               DLOGMOD=FBSSMODE,
               SPAN=(450)

SUT1141  LU    LOCADDR=41,
               USSTAB=ROBTAB,
               DLOGMOD=ENTRYP

SUT1181  LU    LOCADDR=81,
               USSTAB=ROBTAB,
               DLOGMOD=ENTRYZ
```

Note: If you are using the LANDP for AIX SNA server, the IDBLK keyword must be set to the value 071.

SNA over TCP/IP dependent LUs

When using the TCP/IP wide area communications server the dependent LU names must have a common alphanumeric prefix and a numeric suffix with a constant difference between the suffix and the LOCADDR. For example:

```
IYCKT300 LU LOCADDR=2
IYCKT301 LU LOCADDR=3
IYCKT327 LU LOCADDR=29
IYCKT481 LU LOCADDR=183
```

VTAM MODETAB

Following is an example of VTAM MODETAB for the previous definitions.

Note: This is only an example; you *must* tailor it to your environment and requirements.

```
MODFBSS MODETAB
```

MODEENT for LU type 0

```
ENTRYF   MODEENT LOGMODE=ENTRYF,
                 FMPROF=X'04',
                 TSPROF=X'04',
                 PRIPROT=X'B1',
                 SECPROT=X'B1',
                 COMPROT=X'7080',
                 RUSIZES=X'8787'
```

Note: As an alternative, the logmode EMU3790 in the default MVS VTAM mode table can be used.

MODEENT for LU type 1 (IBM LANDP 3287 printer emulator)

```
ENTRYP   MODEENT LOGMODE=ENTRYP,
                 TSPROF=X'03',
                 FMPROF=X'03',
                 PRIPROT=X'B1',
                 SECPROT=X'B0',
                 COMPROT=X'3080',
                 PSERVIC=X'010000F90000000000000000',
                 RUSIZES=X'8585',
                 PSNDPAC=X'01',
                 SRCVPAC=X'01'
```

MODEENT for LU type 2 (DBCS)

```
LDPDBCS  MODEENT LOGMODE=LDPDBCS,
                 TSPROF=X'03',
                 FMPROF=X'03',
                 PRIPROT=X'B1',
                 SECPROT=X'B0',
                 COMPROT=X'3080',
                 PSERVIC=X'028000000000000000000300',
                 RUSIZES=X'8585'
```

Note: This LOGMODE corresponds to the CICS-supplied typeterm DFHLU2 in DFHTYPE.

MODEENT for LU type 2 (IBM LANDP 3270 emulator)

```
ENTRYZ   MODEENT LOGMODE=ENTRYZ,
                 TSPROF=X'03',
                 FMPROF=X'03',
                 PRIPROT=X'B1',
                 SECPROT=X'90',
                 COMPROT=X'3080',
                 PSERVIC=X'028000000000185000007E00',
                 RUSIZES=X'87C7'
```

Note: Nonqueryable LU_2s are also now supported.

The LANDP V5 emulator rejects the bind image if any of the following is true:

- The session is not Logical Unit Type 2 (LU2).
- Cryptography is requested for the non-crypto version of the emulator.
- The primary logical unit maximum send RU size is greater than the emulator host buffer size.
- The queryable flag is not on, and the screen dimensions (rows and columns) in the bind image are incompatible with those supplied on the emulator LOADER statement.

It is recommended that you set on the queryable flag in the VTAM MODEENT LOGMODE statement. This provides greater flexibility and allows the use of the extended color and highlighting support.

MODEENT for RCMS

```
ENTDSX11 MODEENT LOGMODE=ENTDSX11,
                 FMPROF=X'03',
                 TSPROF=X'04',
                 PRIPROT=X'B0',
                 SECPROT=X'A0',
                 COMPROT=X'4000',
                 RUSIZES=X'8585'
```

MODEENT for LU type 6.2

This is an example for dependent LU:

```
SNASVCMG MODEENT LOGMODE=SNASVCMG,
                 TYPE=0,
                 FMPROF=X'13',
                 TSPROF=X'07',
                 PRIPROT=X'B0',
                 SECPROT=X'B0',
                 COMPROT=X'D0B1',
                 SSNDPAC=X'00',
                 SRCVPAC=X'00',
                 RUSIZES=X'8585',
                 PSNDPAC=X'00',
                 PSERVIC=X'060200000000000000000300'
```

This is an example for peer-to-peer LU:

```
FBSSMODE MODEENT LOGMODE=FBSSMODE,
                 TYPE=0,
                 FMPROF=X'13',
                 TSPROF=X'07',
                 PRIPROT=X'B0',
                 SECPROT=X'B0',
                 COMPROT=X'D0B1',
                 SSNDPAC=X'00',
                 SRCVPAC=X'00',
                 RUSIZES=X'F7F7',
                 PSNDPAC=X'00',
                 PSERVIC=X'060200000000000000002C00'
```

CICS definition examples

Note: These are only examples; you *must* tailor them to your environment and requirements.

Definition for LU_0

```
ENTRYF     MODEENT LOGMODE=ENTRTF,
           FMPROF=X'04',
           TSPROF=X'04',
           PRIPROT=X'B1',
           SECPROT=X'B0',
           COMPROT=X'7080',
           RUSIZES='8787'

TYPETERM = ALU0M2
TERM     = L711
```

Note: If the logmode EMU3790 in the MVS VTAM default mode table is used, the IBM-supplied TYPETERM DFHLU2C2 in DFHTYPE can be used.

Definition for LU_1

```
TYPETERM(LDPSCSP)      GROUP(LANDP)    DESCRIPTION()
CICS/ESA®  RDO OFF-LINE UTILITY PROGRAM
                       RESOURCE-TYPE
                                       DEVICE(SCSPRINT)        TERMMODEL()               SESSIONTYPE()
                                       LDCLIST()               SHIPPABLE(NO)
                       MAPPING-PROPERTIES
                                       PAGESIZE(24,80)         ALTPAGE(0,0)              ALTSUFFIX()
                                       FMHPARM(NO)             OBOPERID(NO)
                       PAGING-PROPERTIES
                                       AUTOPAGE(YES)
                       DEVICE-PROPERTIES
                                       DEFSCREEN(0,0)          ALTSCREEN(   ,   )        APLKYBD(NO)
                                       APLTEXT(NO)             AUDIBLEALARM(NO)          COLOR(NO)
                                       COPY(NO)                DUALCASEKYBD(NO)          EXTENDEDDS(NO)
                                       HILIGHT(NO)             KATAKANA(NO)              LIGHTPEN(NO)
                                       MSRCONTROL(NO)          OBFORMAT(NO)              PARTITIONS(NO)
                                       PRINTADAPTER(NO)        PROGSYMBOLS(NO)           VALIDATION(NO)
                                       FORMFEED(NO)            HORIZFORM(NO)             VERTICALFORM(NO)
                                       TEXTKYBD(NO)            TEXTPRINT(NO)             QUERY(NO)
                                       OUTLINE(NO)             SOSI(NO)                  BACKTRANS(NO)
                                       CGCSGID(0,0)
                       SESSION-PROPERTIES
                                       ASCII(NO)               SENDSIZE(256)             RECEIVESIZE(256)
                                       BRACKET(YES)            LOGMODE()
                       DIAGNOSTIC-DISPLAY
                                       ERRLASTLINE(NO)         ERRINTENSIFY(NO)          ERRCOLOR(NO)
                                       ERRHILIGHT(NO)
                       OPERATIONAL-PROPERTIES
                                       AUTOCONNECT(NO)         ATI(YES)                  TTI(YES)
                                       CREATESESS(NO)          RELREQ(YES)               DISCREQ(YES)
                                       NEPCLASS(0)             SIGNOFF(YES)              XRFSIGNOFF(NOFORCE)
                       MESSAGE-RECEIVING-PROPERTIES
                                       ROUTEDMSGS(ALL)         LOGONMSG(NO)
                       APPLICATION-FEATURES
                                       BUILDCHAIN(NO)          USERAREALEN(40)           IOAREALEN(512,0)
                                       UCTRAN(NO)
                       RECOVERY

                                       RECOVOPTION(SYSDEFAULT)
                                       RECOVNOTIFY(NONE)
```

Definition for forwarding

```
TYPETERM(A360001F)     GROUP(TCTHURS)  DESCRIPTION()
CICS/ESA   RDO OFF-LINE UTILITY PROGRAM
                       RESOURCE-TYPE
                                       DEVICE(3600)            TERMMODEL()               SESSIONTYPE()
                                       LDCLIST()               SHIPPABLE(NO)
                       MAPPING-PROPERTIES
                                       PAGESIZE(1,40)          ALTPAGE(0,0)              ALTSUFFIX()
                                       FMHPARM(NO)             OBOPERID(NO)
                       PAGING-PROPERTIES
                                       AUTOPAGE(YES)
                       DEVICE-PROPERTIES
                                       DEFSCREEN(0,0)          ALTSCREEN(   ,   )        APLKYBD(NO)
                                       APLTEXT(NO)             AUDIBLEALARM(NO)          COLOR(NO)
                                       COPY(NO)                DUALCASEKYBD(NO)          EXTENDEDDS(NO)
                                       HILIGHT(NO)             KATAKANA(NO)              LIGHTPEN(NO)
                                       MSRCONTROL(NO)          OBFORMAT(NO)              PARTITIONS(NO)
                                       PRINTADAPTER(NO)        PROGSYMBOLS(NO)           VALIDATION(NO)
                                       FORMFEED(YES)           HORIZFORM(NO)             VERTICALFORM(NO)
                                       TEXTKYBD(NO)            TEXTPRINT(NO)             QUERY(NO)
                                       OUTLINE(NO)             SOSI(NO)                  BACKTRANS(NO)
                                       CGCSGID(0,0)
                       SESSION-PROPERTIES
                                       ASCII(NO)               SENDSIZE(30270)           RECEIVESIZE(30720)
                                       BRACKET(YES)            LOGMODE(0)
                       DIAGNOSTIC-DISPLAY
                                       ERRLASTLINE(NO)         ERRINTENSIFY(NO)          ERRCOLOR(NO)
                                       ERRHILIGHT(NO)
                       OPERATIONAL-PROPERTIES
                                       AUTOCONNECT(NO)         ATI(YES)                  TTI(YES)
                                       CREATESESS(YES)         RELREQ(YES)               DISCREQ(YES)
                                       NEPCLASS(0)             SIGNOFF(YES)              XRFSIGNOFF(NOFORCE)
                       MESSAGE-RECEIVING-PROPERTIES
                                       ROUTEDMSGS(ALL)         LOGONMSG(NO)
                       APPLICATION-FEATURES
                                       BUILDCHAIN(YES)         USERAREALEN(40)           IOAREALEN(1024,32767)
                                       UCTRAN(NO)
                       RECOVERY

                                       RECOVOPTION(SYSDEFAULT)
                                       RECOVNOTIFY(NONE)
```

Definition for LU_2

The CICS-supplied TYPETERM definition DFHLU2C2 in group DFHTYPE corresponds to the VTAM logmode ENTRYZ shown.

For an LU_2 with extended data streams, TYPETERM DFHLU2 in group DFHTYPE is suitable.

IMS/VS terminal statement examples

Note: This is only an example; you *must* tailor it to your environment and requirements.

Definition for LU_0

```
TYPE     UNITYPE=SLUTYPEP

TERMINAL NAME=DCDSZU00,
         OUTBUF=1024,
         OPTIONS=(TRANRESP, PAGDEL, ACK, NOBID)
```

Definition for LU_1

```
TYPE     UNITYPE=SLUTYPE1

TERMINAL NAME=DCDPZU02
         COMPT1=(PRINTER1,MFS=SCS1),
         OPTIONS=(OPNDST, SHARE, RELRQ),
         OUTBUF=256
```

Definition for LU_2

```
TYPE     UNITYPE=SLUTYPE2

TERMINAL NAME=LUFBS2,
         TYPE=3270-A02,
         SIZE=(24,80),
```

IBM AS/400 connectivity using SNA LU_2 protocol

Communication between an AS/400 system and a LANDP workstation can use SNA LU_2 protocols. The following parameters must be defined in the AS/400 computer:

Line

This AS/400 definition must have the same communications parameters as specified in LANDP customization.

Communication Controller or Workstation Control Unit

- The Communication Controller (only in OS/400 Release 1.2) supported by LANDP is equivalent to an IBM 4701 Model 0, Type; Finance.

  The Station Address must be the same as the PU address defined in LANDP customization for SNA Communication server.

- The Workstation Control Unit supported by LANDP is equivalent to an IBM 3174 Model 0.

The Station Address must be the same as the PU address defined in LANDP customization for SNA Communication server.

Display Station

The Display Station supported by LANDP is equivalent to an IBM 3277, Model 0.

The Local Address must be the same as the LU defined for the emulator session during LANDP customization.

Remarks:

- Use the 3270 emulator keyboard definition vector KBD3270 to define the AS/400 function keys for your workstation keyboard during customization. (The AS/400 function keys are described in *AS/400 Communications User's Guide,* SC21-9601.)
- In an AS/400 computer working in S/36 Native mode, and in a S/36 computer, the Command Entry Line is not automatically displayed in upper case, as it normally is in a S/36 Environment.

IBM AS/400 connectivity using SNA LU_0 protocol

The following parameters must be defined in the AS/400 system for communications between AS/400 system and LANDP workstations using SNA LU_0 protocol:

Line

This AS/400 definition must have the same communication parameters as specified in LANDP customization.

Communication Controller

The Controllers supported are:

- IBM 4701 model 0 type Finance.
- IBM 4702 model 0 type Finance.

The Station Address must be the same as the one defined in FBSS Customization for SNA Communication.

Note: This Communication Controller definition is available with OS/400* Release 1.2 and up.

Communication Device

The Communication Devices supported are:

- IBM 4704 type Finance. Used for Non-ICF Finance Communications.
- IBM *FNCICF type Finance. Used for ICF Finance Communications.

  Note: OS/400 Release 2.0 is needed.

The Local Address must be the same as the LU defined in LANDP customization for the LU_0 Session.

There is no need to configure the LANDP controller as a 3174 because AS/400 also supports attachment of 3270 devices to 4700 controllers.

Appendix G. LANDP implementation using WorkSpace On-Demand

WorkSpace On-Demand (WSoD), a component of OS/2 Warp Server for e-Business, provides server-managed client support. The following describes how LANDP can be integrated into the WSoD client environment.

WorkSpace On-Demand Road Map gives a detailed walkthrough of how to install an application onto a WSoD server. Figure 16 (based on page 59 of *WorkSpace On-Demand Handbook (Release 2)*) shows how an application like LANDP is integrated into the WSoD directory structure for remote IPL.

```
\──────IBMLAN─┬─RPL─┬─BB20.US
              │     │
              │     │
              │     ├─LANDP (A)
              │     │
              │     │
              │     ├─MACHINES─┬─BB20.US──
              │     │          │
              │     │          ├─client_name──┬─config.sys
              │     │          │              │
              │     │          │              └─LANDP (B)
              │     └─FITS     └─client_name ..
              │
              │
              ├─DBDC───BB20USDU.FIT
              │
              │
              └─RPLUSER─┬─BB20.US
                        │
                        ├─client_name──┬─
                        │              │
                        │              └─LANDP (C)
                        └─client_name ..
```

Figure 16. Integrating LANDP into WorkSpace On-Demand directory structure

The application (LANDP) directory structure is as follows:

- Files to be shared by multiple WSoD clients are in directory `RPL\`*`yyyy`* ***(read-only)***, where *yyyy* is the directory for the LANDP files (directory **A**).
- Machine-specific files ***(read-only)*** are in `RPL\MACHINES\`*`client_name`*`\`*`yyyy`* (directory **B**).
- Machine-specific files ***(read-write)*** are in `\IBMLAN\RPLUSER\`*`client_name`*`\`*`yyyy`* (directory **C**).

Note: There is a set of files (for **B** and **C**) for each WSoD client that is also a LANDP workstation.

Distributing files

Following customisation, the LANDP files distributed by GETTING are ***machine-specific***. They are for a specific workstation (although they include many files that could be shared across other workstations). There are no user specific files. The following is an analysis of how the LANDP files could be distributed into the WSoD directory structure.

The following LANDP files can be shared across multiple machines, and are read-only (that is, suitable for inclusion in directory **A**):

All EXE files
All DLL files
All HLP files
All MSG files
All COM files
All SYS files
Device drivers
SERVERS.LST
3270 translate tables (AE3270.* DIS3270.* EA3270.* FK3270.*)
RCMS translate tables (AERCMS.* EARCMS.*)
All PAN files

All other files should be placed into a machine-unique directory. The following files may be machine unique (read/write or read/only). Included are files that are targeted to a specific machine and that may be modified by selected servers. The following are suitable for inclusion in directory **C**:

AUTOFBSS.BAT
AUTOFBSS.CMD
Other command files (for example, CREADB.CMD, RCK.CMD, RST.CMD)
*.CFG
Shared file server files (*.DBD, *.PCB, *.SEQ)
SNA.BID

There are many other miscellaneous files that can be distributed to each workstation dependent upon the servers selected, and that convey machine-configured information specific to a workstation.

These should similarly be placed into a machine-unique directory.

CONFIG.SYS updates

The CONFIG.ADD file includes modifications to CONFIG.SYS that are required for a given WSoD client (for example to include support for additional device drivers). This is identical to the process for a non-WSoD workstation. CONFIG.SYS is usually in `RPL\MACHINES\`*`client_name`*`\CONFIG.SYS`. Typical statements that could be added include:

```
LIBPATH=Z:\LANDP
DPATH=Z:\LANDP
DEVICE=Z:\LANDP\EHCVDMVD.SYS
DEVICE=Z:\LANDP\4772PDD.SYS /D1 /C11 /B19600
```

File index table (FIT) updates

This implementation is a machine-specific application. It does not follow the user around and always stays with that machine.

The following entries need to be added to the machine's file index table (FIT file). The FIT file entries are dependent upon which LANDP servers are assigned to the WSoD client. The following is an example of the relevant LANDP section of a FIT file used for a WSoD client on which Shared-File server and System Manager will run.

One possible suggestion is to add these entries to the machine classes that LANDP will be running on. This can be done either by modifying the machine FIT file `\IBMLAN\RPL\FITS\DFBB20`*`xx`*`.FIT` (where *xx* is the country code) before creating those machines, or by modifying the individual machine FIT files.

```
LANDP Support
Z:\LANDP                LANDP
Z:\LANDP\*.CFG              \\rplservername\WRKFILES\client_name\LANDP
Z:\LANDP\*.LOG              \\rplservername\WRKFILES\client_name\LANDP
Z:\LANDP\*.BAT              \\rplservername\WRKFILES\client_name\LANDP
Z:\LANDP\*.DAT              \\rplservername\WRKFILES\client_name\LANDP
Z:\LANDP\*.DBD              \\rplservername\WRKFILES\client_name\LANDP
Z:\LANDP\*.PCB              \\rplservername\WRKFILES\client_name\LANDP
Z:\LANDP\*.SEQ              \\rplservername\WRKFILES\client_name\LANDP
Z:\LANDP\EHCLOG*.DAT        \\rplservername\WRKFILES\client_name\LANDP
Z:\LANDP\EHCTRC*.BAT        \\rplservername\WRKFILES\client_name\LANDP
Z:\LANDP\FBSS#*             \\rplservername\WRKFILES\client_name\LANDP
Z:\LANDP\AUTOFBSS.CMD       \\rplservername\WRKFILES\client_name\LANDP\AUTOFBSS.CMD
```

Multiple workgroup support

If you wish to have more than one LANDP workgroup in a single WSOD server machine, one option is to define an additional instance of LANDP. For example, if two workgroups have different LANCONF LANGUAGE values, two sets of FIT file entries (as above) could be defined:

```
;LANDP Support (English)
Z:\LANDPENG                 LANDPENG
Z:\LANDPENG\*.CFG            \\rplservername\WRKFILES\client_name\LANDPENG
 ...
;LANDPESP Support (Spanish)
Z:\LANDPESP                 LANDPESP
Z:\LANDPESP\*.CFG            \\rplservername\WRKFILES\client_name\LANDPESP
 ...
```

LANDP customization considerations

The assigned drive location on the WSoD server on which the client system and application files reside defaults to Z:. This is the client user's boot drive. The rest of LANDP would have to be on the same drive and configured that way. For example, the target drive for location of the files used by RCMS server (PAR&RCMS parameters 5 and 6) and System Manager server (PAR&SMGR parameters 5, 6, 7, 8) for the production workstation would need to conform to this default, Z:. (These files are subsequently copied to their target directories by EHCVAL).

For example:

```
/* RCMS */
      SERVER=RCMS,
      PAR&RCMS=(AB,,,,Z:\LANDP\,Z:\LANDP\,,,),
/* System Manager */
      SERVER=(SMGR,,N),
      PAR&SMGR=(AB,,Z:\LANDP\,SMGRPRF1,Z:\LANDP\,
                      Z:\LANDP\,Z:\LANDP\,Z:\LANDP\,Z,USER1)
```

Generally, LANDP facilities run on a WSoD client as they do on an OS/2 workstation. Servers that have been tested include Shared_File server, Financial Printer server with the financial printer attached to the WSoD client, System Manager and System Manager Operator, and 3270 emulator running in a VDM.

Careful attention needs to be given to planning the use of Shared File (SF) server, specifically when using XLR or Replicator support. The underlying purpose of XLR or Replicator support is to provide immediate database recovery from a backup server in case of machine or system failure on the workstation on which SF server resides. Clearly it is inappropriate to have both the SF server and its backup residing on the same WSoD server.

Run Time

By default, WSoD does not allow anything to be written to a disk on a WSoD client workstation (it may be a media-less workstation). Any I/O requests are read/written to the disk on the WSoD server, for example, shared-file access or any attempt to write trace or logging information.

References:

1 *WorkSpace On-Demand Road Map*, G325-3849 (10/98)

2 *WorkSpace On-Demand Handbook Release 2.0*, SG24-5117 (10/98)

3 *WorkSpace On-Demand Early Customer Experiences*, SG24-5107 (10/98)

4 *IBM Up and Running! OS/2 Warp Server*, S25H-8004

5 *WorkSpace On-Demand Administrator’s Guide*, (10/98), on the website `www.ibm.com/software/network/workspace/library`

Appendix H. 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
- AIX/6000
- AnyNet
- Application System/400
- AS/400
- AT
- BookManager
- C/2
- CICS
- CICS OS/2
- CICS/ESA
- COBOL/2
- Common User Access
- CUA
- DB2 Universal Database
- Distributed Database Connection Services/2
- Distributed Relational Database Architecture
- DRDA
- Extended Services
- IBM
- IBMLink
- IMS/ESA
- LAN Distance
- LANDP
- Macro Assembler/2
- Micro Channel
- MQSeries
- MVS/XA
- OfficeVision
- OfficeVision/MVS
- OfficeVision/VM
- Operating System/2
- Operating System/400
- OS/2
- OS/390
- Presentation Manager
- RETAIN
- RISC System/6000
- RS/6000
- S/390
- SecureWay
- SP
- ThinkPad
- VisualAge
- VisualGen
- VM/ESA
- VSE/ESA
- VTAM
- WIN-OS/2
- Workplace Shell
- WebSphere
- Xstation Manager
- XT

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 Kjøbenhavns 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. (I) (A) (2) Any value of an independent variable; for example, a search key; a number identifying the location of an item in a table. (I) (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) Contract 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 resourec 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 LANDP 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 LANDP family trace facility that informs about the LANDP-related LAN and displays the status of the local area network.

LAN Distributed Platform. The former name for the LANDP 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

pacing. 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. This clients do not require 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 an 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 deciphers 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.

[2] 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.[3]

IBM Personal Computer LAN Support Program.[3]

IBM Personal Computer Baseband and Broadband.[3]

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.

[3] 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

Sending your comments to IBM

LANDP® Family

Installation and Customization

GC34-5530-00

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

Installation and Customization

GC34-5530-00

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

LANDP® Family
LANDP Installation and Customization GC34-5530-00

IBM

You can send your comments POST FREE on this form from any one of these countries:

Australia
Belgium
Bermuda
Cyprus
Denmark
Finland
France
Germany
Greece
Hong Kong
Iceland
Israel
Italy
Luxembourg
Monaco
Netherlands
New Zealand
Norway
Portugal
Republic of Ireland
Singapore
Spain
Sweden
Switzerland
United Arab Emirates
United States of America

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

NE PAS AFFRANCHIR

NO STAMP REQUIRED

IBM

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

GC34-5530-00